

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДОНЕЦЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ  
ІМ. М. ТУГАН-БАРАНОВСЬКОГО

Н. М. Войтюшенко  
А. І. Остапець

# ІНФОРМАТИКА І КОМП'ЮТЕРНА ТЕХНІКА

*2-ге видання*

*Рекомендовано  
Міністерством освіти і науки України  
як навчальний посібник для студентів  
вищих навчальних закладів*

Київ  
“Центр учбової літератури”  
2009

УДК 004(075.8)  
ББК 32.97я73  
В 65

*Гриф надано  
Міністерством освіти і науки України  
(Лист № 14/18.2-710 від 20.03.2006)*

Рецензенти:

**Гузь М. Г.** – доктор економічних наук, професор;

**Чилікін О. І.** – доктор економічних наук, професор;

**Сукманов В. О.** – доктор технічних наук, професор.

Войтюшенко Н. М., Інформатика і комп'ютерна техніка: *навч. пос. [для студ. вищ. навч. закл.]*/ Н. М. Войтюшенко, А. І. Остапеч. – [2-ге вид.]. – К.: Центр учбової літератури, 2009. – 564 с. – ISBN 978-966-364-825-5

Навчальний посібник містить матеріал за базовим курсом інформатики і комп'ютерної техніки для користувачів персональних комп'ютерів, що відповідає нормативній програмі Міністерства науки і освіти України. У нього входять інформаційні, технічні і програмні основи ПК, основні зведення і матеріал по одержанню навичок роботи з операційною системою Windows, сервісними й офісними програмами, мережними ресурсами, основи алгоритмізації базових обчислювальних процесів, основні зведення і матеріал по одержанню навичок роботи з об'єктно-орієнтованою мовою Visual Basic, приклади використання Visual Basic у додатках Microsoft Office.

Навчальний посібник призначений для студентів економічних і технічних спеціальностей усіх форм навчання, що освоюють базовий курс комп'ютерної підготовки.

**УДК 004(075.8)**  
**ББК 32.97я73**

**ISBN 978-966-364-825-5**

© Войтюшенко Н.М., Остапеч А.І., 2009.  
© Центр учбової літератури, 2009.

## Зміст

Вступ .....	6
Розділ 1. Загальні основи обробки даних на комп'ютерах.....	9
1.1 Інформаційні основи комп'ютерної техніки.....	9
1.1.1 Інформація й форми її уявлення.....	10
1.1.2 Інформаційні процеси й технології.....	14
1.2 Технічні засоби обробки інформації.....	19
1.2.1 Персональний комп'ютер і склад його системного блоку...	19
1.2.2 Зовнішні пристрої ПК.....	28
1.3 Програмне забезпечення ПК .....	47
Розділ 2 Операційні системи і сервісні програми.....	54
2.1 Операційна система Windows.....	54
2.1.1 Історична довідка і загальна характеристика Windows. ....	54
2.1.2 Робочий стіл Windows.....	55
2.1.3 Прийоми роботи з маніпулятором «миша».....	57
2.1.4 Вікна Windows.....	58
2.1.5 Папки й файли. Програми Мой комп'ютер і Проводник ....	64
2.1.6 Файлові операції в Windows.....	68
2.1.7 Вбудовані додатки Windows .....	79
2.2 Сервісні програми.....	85
2.2.1 Віруси й боротьба з ними.....	85
2.2.2 Обслуговування дисків.....	94
2.2.3 Програми архівації .....	99
2.2.4 Запис даних на компакт-диски.....	108
Розділ 3. Системи обробки текстів.....	119
3.1 Текстовий редактор Word.....	119
3.1.1 Уведення, редагування і ручне форматування тексту в Microsoft Word.....	121
3.1.2 Графічні об'єкти в текстах.....	125
3.1.3 Оформлювальні засоби .....	132
3.1.4 Засоби автоматизації вводу й заповнення документів.....	141
3.2 Системи оптичного розпізнавання текстів.....	164
3.3 Системи електронного перекладу.....	170
3.3.1 Загальна характеристика програм машинного перекладу..	171
3.3.2 Основні прийоми робіт з додатком Пролінг Офіс Стандарт (програми РУТА і ПЛАЙ).....	181
3.3.3. Програма Pragma 4.x.....	186
Розділ 4. Табличний процесор Excel. ....	191
4.1 Основи роботи з листом.....	191
4.1.1 Загальна характеристика й область застосування програм обробки електронних таблиць. Програма Excel...	191
4.1.2 Вікно EXCEL і його елементи.....	192
4.1.3 Заповнення таблиці робочого листа EXCEL.....	194

4.1.4	Створення діаграм.....	199
4.1.5	Друк робочого листа.....	201
4.2	Використання формул.....	217
4.2.1	Типи посилань у формулах.....	217
4.2.2	Загальна характеристика умонтованих функцій Excel.....	219
4.2.3	Формули з масивами.....	225
4.2.4	Присвоєння імен чарункам і діапазнам.....	228
4.2.5	Робота з групою листів. Об'ємні формули.....	230
4.3	Консолідація листів.....	233
4.3.1	Консолідація по розташуванню.....	234
4.3.2	Консолідація по категорії.....	236
4.3.3	Створення зв'язків із вхідними листами.....	237
4.4	Використання зведених таблиць .....	238
Розділ 5.	Підготовка прилюдних виступів і презентацій.....	255
5.1	Загальна характеристика пакета PowerPoint .....	255
5.2	Створення примітивної презентації.....	256
5.3	Представлення презентації.....	261
5.4	Створення витонченої презентації .....	266
5.5	Переміщення і демонстрація презентації на інших ПК .....	274
Розділ 6.	Комп'ютерні мережі та мережні технології.....	277
6.1	Локальні комп'ютерні мережі.....	278
6.1.1	Локальні комп'ютерні мережі та їх можливості.....	278
6.1.2	Топологія ЛКМ і методи доступу до каналів зв'язку.....	280
6.1.3	Програмні засоби ЛКМ.....	285
6.1.4	Захист даних у мережі.....	289
6.2	Глобальні мережі. INTERNET.....	290
6.2.1	Загальні принципи побудови глобальних мереж.....	290
6.2.2	Підключення до INTERNET.....	299
6.2.3	Інформаційні послуги Internet.....	302
6.2.4	Організація комплексної роботи користувача в Internet .....	318
6.2.5	Інформаційно-пошукові системи INTERNET, технологія пошуку.....	321
Розділ 7.	Основи алгоритмізації і програмування (Інструментальні засоби програмування) .....	327
7.1	Загальна характеристика етапів рішення задач на ОМ.....	328
7.2	Поняття алгоритму. Основні властивості і методи запису алгоритмів.....	330
7.3	Алгоритми основних видів обчислювальних процесів .....	334
7.3.1	Простий не розгалужений обчислювальний процес.....	334
7.3.2	Розгалужені обчислювальні процеси .....	335
7.3.3	Циклічні обчислювальні процеси .....	336
7.4	Алгоритмічні мови: поняття і класифікація.....	346

Розділ 8. Загальні відомості й основні поняття мови програмування Visual Basic.....	353
8.1 Користувальницька оболонка пакета.....	353
8.2 Основні принципи розробки програм (проектів) у середовищі Visual Basic.....	355
8.3 Реалізація першого етапу проектування.....	361
8.4 Реалізація другого етапу проектування.....	365
8.4.1 Елементи мови.....	365
8.4.2 Уведення – вивід даних.....	368
8.4.3 Надання привабливості формі й створення виконавчого файлу.....	371
8.4.4 Використання лінійок прокручування.....	374
8.5 Реалізація розгалужених алгоритмів.....	377
8.6 Селекторні кнопки (перемикачі), прапорці, рамки.....	388
8.7 Проектування циклічних процесів.....	398
8.7.1 Арифметичні цикли, використання декількох форм у проекті	399
8.7.2 Ітераційні цикли.....	405
8.7.3 Складні цикли, використання меню.....	407
8.8 Робота з файлами.....	421
Розділ 9. Створення баз даних засобами MS Access.....	431
9.1 Призначення і загальні принципи побудови СУБД.....	431
9.2 Створення бази даних.....	435
9.2.1 Таблиці.....	440
9.2.2 Екранні форми.....	449
9.2.3 Запити.....	456
9.2.4 Звіти.....	471
9.2.5 Робота у режимі конструктора.....	478
9.3 Сервісні функції пакета.....	485
Розділ 10. Використання Visual Basic для автоматизації робіт у Microsoft Office .....	490
10.1 Автоматизація робіт з документами у Microsoft Word за допомогою VBA.....	491
10.2 Автоматизація робіт із таблицями у Microsoft Excel за допомогою VBA.....	503
10.2.1 Створення і використання користувальницьких функцій.....	505
10.2.2 Використання шаблонів й керуючих кнопок для створення документів .....	510
10.2.3 Форми як засоб взаємодії користувача і табличного процесора.....	517.
Словник термінів.....	531
Додаток А.....	551
Додаток Б.....	555
Додаток В.....	560

## Вступ

Важко назвати область людської діяльності, у якій не знайшли застосування засоби комп'ютерної техніки. У той же час розвиток самої комп'ютерної техніки й інформаційних технологій у цілому характеризується стрімкістю, швидкою зміною концептуальних представлень, технічних засобів, методів і сфер застосування. Тому сучасний висококваліфікований фахівець повинний мати високий рівень знань в області інформаційних технологій і вільно адаптуватися до складу технічних і програмних засобів, що постійно змінюється.

Навчальний посібник підготовлений за курсом «Інформатика і комп'ютерна техніка» відповідно до типової програми базового рівня підготовки фахівців вищої кваліфікації економічних і технічних спеціальностей з урахуванням сучасних досягнень в області технічного й програмного забезпечення. Матеріали посібника засновані на досвіді викладання ряду дисциплін, таких як «Інформатика і комп'ютеризація», «Комп'ютерна техніка і програмування», «Комп'ютерні технології», «Інформаційні системи», «АРМ фахівця» на денному й заочному відділеннях.

З огляду на сучасні тенденції по активізації самостійної роботи студентів у посібнику особлива увага була приділена питанням освоєння практичних навичок роботи на ПК із сучасними програмними засобами. Матеріал у посібнику даний з врахуванням різної початкової підготовки студентів.

Навчальний посібник складається з десяти розділів.

У розділі 1 «Загальні основи підготовки даних на комп'ютерах» визначені базові поняття інформатики й інформаційних технологій, коротко розглянуті питання архітектури сучасних ПК; склад, структура і характеристика програмного забезпечення в цілому.

У розділі 2 «Операційні системи і сервісні програми» приведені загальна характеристика, основні прийоми і конкретні приклади роботи з операційною системою Windows і основними сервісними програмами, необхідними для роботи користувача.

У розділі 3 «Системи обробки тексту» охарактеризовані основні сучасні програмні засоби для роботи з текстом: текстові редактори, засоби оптичного розпізнавання, перекладачі. Для кожного виду програм

описана і показана на прикладах технологія роботи з формування документів різної складності.

У розділі 4 «Табличний процесор Excel» наведені характеристика табличного процесора і приклади його використання по формуванню таблиць і діаграм. Розділ також містить матеріал, що передує професійному використанню Excel: надані додаткові зведення по адресації даних, по роботі з формулами, аркушами і книгами, по консолідації і зведених таблицях.

У розділі 5 «Підготовка публічних виступів і презентацій» розглянута робота пакета Power Point по створенню презентацій і виступів із демонстрацією на ПК.

У розділі 6 «Комп'ютерні мережі і мережні технології» дана коротка характеристика локальних і глобальних мереж, можливостей Internet, зведення про підключення і використання основних інформаційних сервісів глобальної комп'ютерної мережі.

У розділі 7 «Основи алгоритмізації і програмування (Інструментальні способи програмування)» дана коротка характеристика етапів розв'язання задач на комп'ютерах; визначення алгоритму, основні властивості і методи запису алгоритмів; розглянута алгоритмізація основних видів обчислювальних процесів; поняття і характеристика алгоритмічних мов.

У розділі 8 «Загальні зведення й основні поняття мови програмування Visual Basic» описана користувальницька оболонка пакета; основні принципи розробки програм (проектів) у середовищі Visual Basic: елементи мови, приклади реалізації першого і другого етапів проектування лінійних, розгалужених, циклічних алгоритмів, технологія роботи зі створення і використання файлів прямого доступу.

У розділі 9 «Створення баз даних засобами MS Access» розглянуті призначення і загальні принципи побудови баз даних на ПК, створення і використання основних об'єктів бази даних на прикладі СУБД Access.

У розділі 10 «Використання Visual Basic для автоматизації робіт у Microsoft Office» звернена увага на використання мови Visual Basic для автоматизації робіт у раніше вивчених додатках – Word і Excel; приведені приклади прийомів програмування у зазначених додатках.

У кожному розділі посібника спочатку розглядаються загальні теоретичні питання, а потім опис технології розв'язання задач супроводжується конкретними прикладами. Наприкінці кожного розділу приводяться контрольні питання, завдання для самоперевірки і література.

При оформленні матеріалів використовувалися наступні угоди:

Що позначається	Як позначається (приклад)	Особливості написання
Клавіші, що натискаються	У квадратних дужках <b>[F4]</b>	Напівжирним шрифтом
Спільне натискання клавіш	У квадратних дужках і з'єднуються знаком плюса <b>[Alt+F4]</b>	Напівжирним шрифтом
Пункт меню	меню або м. <b>Формат</b>	Напівжирним шрифтом
Команда меню або названня інструментальної кнопки	команда або к. <b>Столбец</b>	Напівжирним шрифтом.
Пункт меню	п. <b>Автоподбор ширини</b>	Напівжирним шрифтом
Назва елементів діалогового вікна	вкладка <b>Выравнивание</b>	Напівжирним шрифтом
Повний шлях виконання операції	меню <b>Формат</b> → команда <b>Столбец</b> → п. <b>Автоподбор ширини</b> або м. <b>Формат</b> → к. <b>Ячейки</b> → вкладка <b>Вирівнювання</b> → виставити прапорець <b>Переносити за словами</b>	
Ім'я файлу	<i>Мій малюнок</i>	Курсив
Терміни, що уперше зустрічаються	<b>Колонититул</b>	Жирний курсив



# Розділ 1. Загальні основи обробки даних на комп'ютерах

## 1.1 Інформаційні основи комп'ютерної техніки

По сучасних уявленнях інформація є однією з вхідних категорій світобудови поряд із матерією й енергією. Будь-яка діяльність людини являє собою процес збору й переробки інформації, прийняття на її підставі рішень і їхнє виконання.

У рамках науки інформація є первинним і невизначуваним поняттям. Воно припускає наявність:

- матеріального носія інформації,
- джерела інформації,
- передавача інформації,
- приймача і
- каналу зв'язку між джерелом і приймачем.

Вивчає інформацію інформатика.

**Інформатика** – фундаментальна природнична наука, що вивчає загальні властивості інформації, процеси, методи й засоби її обробки (збір, збереження, перетворення, переміщення, видача) [5, с. 13], а також принципи функціонування цих засобів і методи керування ними.

Як наука інформатика з'явилася порівняно недавно – у 50-і роки двадцятого століття, незважаючи на те, що з інформацією і її обробкою людина мала справу завжди. Це обумовлено можливостями людини й суспільства по нагромадженню, збереженню й обробці інформації. Можна виділити наступні три етапи в розвитку інформаційних можливостей людського суспільства: поява писемності, винахід друкарства й створення електронної обчислювальної техніки.

Так, поява писемності в п'ятому тисячоріччі до нашої ери, дозволила перейти від збереження й передачі інформації безпосередньо людиною до фіксації даних на носіях (табличках, папірусах і т.д., паперові) і передачі їх через значні часові й просторові проміжки.

Винахід друкарства, а потім телеграфу, телефону, радіо, телебачення забезпечив тиражування й доступ до інформації величезної кількості людей, що у свою чергу з'явилося передумовою до лавинообразного зростання потоків інформації. Людині потрібна була допомога в обробці постійно зростаючих обсягів інформації.

Створення електронних обчислювальних машин, обґрунтування принципів виміру, кодування й передачі інформації викликали до життя науку інформатику і забезпечили людині інтелектуальну допомогу по обробці інформації.

Особливості інформатики як наукової й навчальної дисципліни полягають у тому, що прикладна її частина затребувана сьогодні у всіх сферах людського суспільства, людьми різних віків і професій.

### 1.1.1 Інформація й форми її уявлення

Термін «інформація» походить від латинського *informatio* – роз'яснення, виклад, поінформованість.

Більш вузьке визначення дається в техніці, де це поняття містить у собі всі відомості, що є об'єктом збереження, передачі й перетворення.

З поняттям інформації зв'язані такі поняття, як повідомлення, дані й сигнал.

*Сигнал* – будь-який процес, що несе інформацію.

*передачі*

*Дані* – це інформація, представлена у формалізованому виді і призначена для обробки її технічними засобами, наприклад, ЕОМ.

Усе різноманіття інформації навколо нас можна класифікувати по видах. Наприклад, у залежності від області виникнення, інформацію, що відображає процеси і явища неживої природи, називають елементарною, процеси тваринного й рослинного світу – біологічною, людського суспільства – соціальною.

По способах передачі і сприйняття розрізняють наступні види інформації: візуальну – передається видимими образами й символами, аудіальну – звуками, тактильну – відчуттями, органолептичну – запахами й смаком, машинну – видається й сприймається засобами ОТ, і т.п.

Розрізняють дві форми уявлення інформації – *безперервну й дискретну*. Безперервна інформація виникає в результаті протікання безперервних фізичних процесів різної природи і може приймати будь-які проміжні значення. Наприклад, процес протікання електричного струму в ланцюгу, процес механічного переміщення тіла, процес поширення світла. У дискретній формі уявлення інформація в заданих межах може

приймати тільки окремі фіксовані значення. Наприклад, інформація в економічних документах завжди носить дискретний характер.

Величину інформації необхідно вимірювати. **Кількістю інформації** називають числову характеристику, що відображає той ступінь невизначеності (неповноту знань), що зникає після одержання інформації. Цю міру невизначеності в теорії інформації називають ентропією. Якщо в результаті одержання повідомлення досягається повна ясність у якомусь питанні, говорять, що була отримана повна або вичерпна інформація. Якщо після одержання повідомлення невизначеність залишилася колишньою, виходить, інформації отримано не було (нульова інформація).

Наведені міркування показують, що між поняттями інформація, невизначеність і можливість вибору існує тісний зв'язок. Так, будь-яка невизначеність припускає можливість вибору, а будь-яка інформація, зменшуючи невизначеність, зменшує і можливість вибору. При повній інформації вибору немає. Часткова інформація зменшує число варіантів вибору, скорочуючи тим самим невизначеність.

Кількість інформації, яку можна одержати при відповіді на питання типу «так – ні», називається бітом.

**Бит** – мінімальна одиниця кількості інформації, тому що одержати інформацію меншу, чим 1 біт, неможливо. При одержанні інформації в 1 біт невизначеність зменшується в 2 рази.

Група з 8 бітів інформації називається **байтом**. Якщо біт – мінімальна одиниця інформації, то байт – основна. Існують похідні одиниці інформації: кілобайт (Кбайт, Кб), мегабайт (Мбайт, Мб) і гігабайт (Гбайт, Гб).  $1 \text{ Кб} = 1024 \text{ байта}$ ,  $1 \text{ Мбайт} = 1024 \text{ Кбайта}$ ,  $1 \text{ Гб} = 1024 \text{ Мбайта} = (1024 * 1024 * 1024) \text{ байтів}$ . Ці одиниці найчастіше використовують для вказівки обсягу пам'яті ЕОМ. У той же час запис або зчитування даних із пам'яті здійснюється з декількох байтів (чарунка, до якої звертаються). Наприклад, у комп'ютерах ІВМ чарунка пам'яті поєднує 2 байти. Комбінація зв'язаних сусідніх чарунок, що обробляються спільно, називається **машинним словом**.

У сучасних комп'ютерах обробляється мультимедійна інформація, що складається не тільки із символів, але і зі звуків, зображень. Як усі ці різні за формою й сприйняттям людиною види інформації представляються в комп'ютері?

Уся інформація усередині комп'ютера зображується в цифровому виді за допомогою двійкової системи числення, незважаючи на те, що

вимагає приблизно в 3.3 раза більше числа розрядів, чим десяткова. Це зв'язано з тим, що для уявлення розряду двійкового числа може бути використаний будь-який запам'ятовуючий елемент, який має два стійких стани («так – ні», є напруга – немає напруги). Двійкове зображення чисел записується в комп'ютерні чарунки, що мають обмежений розмір (кінцева кількість двійкових розрядів). Це приводить до того, що нескінченна безліч дійсних чисел замінюється кінцевою безліччю кодів цих чисел і операціями над ними.

У комп'ютері використовуються наступні форми уявлення даних:

- цілі числа;
- числа з фіксованою крапкою (комою);
- числа з плаваючою крапкою (комою);
- десяткові числа;
- символічні дані.

Цілі числа займають 2 байти й охоплюють діапазон від  $-32768$  до  $+32768$ . Якщо число перевищує зазначений діапазон, то в залежності від програми, користувачеві може виводитися повідомлення про помилку або число автоматично переводиться в дійсну форму (програми Excel, Calc, MathCAD).

При представленні числа у формі з фіксованою крапкою вказується знак числа й модуль числа. Іноді таку форму представлення чисел називають природною формою. Місце крапки (коми) постійно для всіх чисел і в процесі рішення задач не міняється.

Форма представлення чисел із фіксованою крапкою спрощує апаратну реалізацію ЕОМ, зменшує час виконання машинних операцій, однак при рішенні задач на машині необхідно постійно стежити за тим, щоб усі вхідні дані, проміжні й остаточні результати знаходилися в припустимому діапазоні представлення. Якщо цього не дотримуватись, то можливе переповнення розрядної сітки, і результат обчислень буде невірним. Від цих недоліків у значній мірі вільні ЕОМ, що використовують форму представлення чисел із плаваючою крапкою, або нормалізовану форму.

У формі з плаваючою *комою* число представляється у виді добутку:  $= m \cdot q^p$ ,

де  $m$  – мантиса ( $q^{-1} \leq |m| < q^0$ , тобто мантиса завжди менше нуля, але перша значуща цифра повинна стояти після десяткової крапки);

$q$  – основа системи числення;

$p$  – порядок (ступінь).

Наприклад, 0,1;  $0,589 \cdot 10^3$ ;  $0,97 \cdot 10^{-2}$ ; 0,34E6; 0,86E-5 (E – еквівалентно 10).

Час обробки чисел у формі з плаваючою комою набагато більше, ніж із цілими або фіксованою комою.

При представленні інформації у виді десяткових багато розрядних чисел кожна цифра замінюється двоїчно-десятковим кодом.

Для представлення текстової або алфавітно-цифрової інформації використовують спеціальні числові коди – ДКОИ, КОИ-7, ASCII. Для представлення одного символу приділяється один байт. У багатьох ПК використовується код ASCII.

Кожній формі представлення чисел властиві свої правила обробки, що приводить до необхідності точного опису типів перемінних перед їхнім використанням у програмах. Крім того, числа різних типів вимагають для збереження різних ресурсів пам'яті, що особливо важливо при великих обсягах оброблюваної інформації.

Усе вище сказане явно відноситься до алфавітно-цифрової інформації. А мультимедійна інформація – звук, деякі види зображень (телевізійний сигнал та т.п.) мають аналогову природу. Представлення такої інформації у комп'ютері розглянемо на прикладі звуку.

Звук – це хвильові коливання щільності повітря, що уловлює мікрофон і перетворює у коливання напруги електричного струму. Одержуємо аналоговий сигнал, із яким комп'ютер не вміє працювати. Перетворення аналогового сигналу в цифрову форму здійснюється за допомогою аналого-цифрового перетворювача (АЦП), що через визначені проміжки часу вимірює амплітуду хвилі і зберігає в пам'яті комп'ютера. Безперервному аналоговому сигналові ставиться у відповідність числовий ряд даних про зміну амплітуди звукової хвилі в часі. При відтворенні звуку аналоговий сигнал відновлюється по наявних числових значеннях. Чим більше вимірів в одиницю часу (**частота дискретизації**), тим вище якість перетворення звуку.

Значення частотної дискретизації повинне бути більше частоти звуку, інакше звуковий сигнал перетвориться в неінформативний шум. Людина сприймає звукові хвилі в діапазоні 20...20000 Гц, значить частота дискретизації повинна бути більше 20000 Гц, тобто більше 20 тисяч вимірів у секунду (20 кгц). Для якісного звучання звичайно використовується частота дискретизації 44,1 – 48 кГц, найкращі аудіо формати використовують частоту 96 кГц.

Під кожен результат вимірів відводиться визначена кількість розрядів пам'яті – **розрядність дискретизації**, що характеризує точність вимірів. Звичайно використовується розрядність 16 біт, що дозволяють закодувати  $2^{16} = 65536$  значень амплітуди. (аналогічно палітрі з 65536 звуків).

При стандартних частоті ( 44,1 кГц) і розрядності ( 16 біт=2 байт) дискретизації одна хвилина звуку в пам'яті займає:

при монозапису  $44100 * 2 * 60 = 5292000$  байт;

при стереозапису  $44100 * 2 * 60 * 2 = 10584000$  байт.

Одержуємо, що одна хвилина звучання в незжатому стані займає близько 10 Мбайт. В даний час розроблені формати, що дозволяють зберігати звук у стиснутому стані без втрати якості, наприклад, МР3 ( 1 хвилина займає 1 Мбайт).

### 1.1.2 Інформаційні процеси й технології

Інформація витягається із загального потоку повідомлень і використовується за допомогою інформаційних технологічних процесів. Інформаційні процеси (збір, обробка й передача інформації) завжди відігравали важливу роль у науці, техніці й житті суспільства. У ході еволюції людства проглядається стійка тенденція до автоматизації цих процесів, хоча їхній внутрішній зміст, власне кажучи, залишився незмінним.

**Збір інформації** – це діяльність суб'єкта, у ході якої він одержує відомості про цікавлячий його об'єкт. Збір інформації може виконуватися або людиною, або за допомогою технічних засобів і систем – апаратно. (Наприклад, користувач може одержати інформацію про рух потягів або літаків сам, вивчивши розклад, або ж від іншої людини безпосередньо, або через якісь документи, складені цією людиною, або за допомогою телефону, мережі і т.д.). Задача збору інформації не може бути вирішена у відриві від інших задач, зокрема, задачі обміну інформацією (передачі).

**Обмін інформацією** – це процес, у ході якого джерело інформації її передає, а одержувач – приймає. Якщо в переданих повідомленнях виявлені помилки, то організується повторна передача цієї інформації.

Прийняту інформацію одержувач може використовувати неодноразово. З цієї метою він повинен зафіксувати її на матеріальному носії

(магнітному, фото, кіно й ін.). Процес формування вхідного, несистематизованого масиву інформації називається **нагромадженням** інформації.

**Збереження інформації** – це процес підтримки вхідної інформації у виді, що забезпечує видачу даних по запитах кінцевих користувачів у встановлений термін.

**Обробка інформації** – це упорядкований процес її перетворення відповідно до алгоритму рішення задачі.

Після рішення задачі обробки інформації результат повинен бути виданий кінцевому користувачеві в необхідному виді за допомогою зовнішніх пристроїв ЕОМ у вигляді текстів, таблиць, графіків і ін.

Усі інформаційні процеси тісно зв'язані з поняттями об'єкта, характеристик (властивостей) об'єкта, бази даних, бази знань, інформаційних технологій, інформаційної інфраструктури.

Під **об'єктом** буде розумітися будь-який предмет, явище, процес або стан, що сприймається як деяке ціле, має відмітні ознаки й ім'я. Об'єкти в програмному забезпеченні – операційні системи, драйвери, програми, файли; інші об'єкти – таблиці, малюнки, поля, чарунки і т.д.

Під **характеристикою** (властивістю) об'єкта розуміють сукупність конкретних ознак об'єкта в числовому або якісному вираженні. Наприклад, файл має такі характеристики як розмір, час і дату створення, автора, місце запису.

**База даних** – це автоматизоване сховище даних. У ній за допомогою програмних комплексів – СУБД (**система управління базами даних**) накопичуються, зберігаються, обробляються і представляються в потрібному для користувача вигляді величезні обсяги інформації.

**База знань** – автоматизоване сховище всіх необхідних відомостей про специфіку й закони даної проблемної області, про способи розв'язання задач, які у ній виникають.

**Інформаційна (комп'ютерна) технологія** (ІТ) – це сукупність методів, виробничих процесів і програмно-технічних засобів, об'єднаних у технологічний ланцюжок, що забезпечує збір, обробку, збереження, поширення й відображення інформації з метою зниження трудомісткості процесів одержання й використання інформаційного ресурсу, а також підвищення їхньої надійності й оперативності.

**Інформаційний ресурс** – це особливий вид ресурсу, що ґрунтується на ідеях і знаннях, накопичених у результаті науково-виробничої дія-

льності людей у визначеній предметній області, і представлений у вигляді зручному для накопичення й використання (документи й масиви документів у бібліотеках, архівах, фондах, банках даних, депозитаріях, музейних сховищах і ін.). На відміну від інших видів ресурсів (енергетичних, матеріальних, трудових, фінансових) названий ресурс невичерпний, як невичерпні людські знання.

Таким чином, метою ІТ є одержання інформації, а критеріями її якості є своєчасність доставки інформації користувачеві, її надійність, вірогідність, повнота.

Найважливіші процедури інформаційної технології умовно поділяються на функціонально часові стадії (рисунок 1.1). Послідовність виконання зазначених процедур і сам їхній перелік для конкретних об'єктів може бути різний. Їхній склад і особливості виконання багато в чому залежать від об'єкта, що веде обробку інформації і тих процесів, що протікають у середовищі його знаходження.

З усіх видів технологій інформаційна технологія сфери управління найвимогливіша до «людського фактора», що оказує принциповий вплив на кваліфікацію працівника, зміст його праці, фізичне й розумове навантаження, професійні перспективи й рівень соціальних відносин.

Очевидно, що реалізацію частини зазначених процедур можна покласти на комп'ютерну техніку. В останні роки ІТ пройшли шлях від автоматизації окремих процесів до створення систем, що оказують безпосередній вплив на бізнес, науку, у цілому на життя людського суспільства. Такі успіхи стали можливі лише при створенні і постійному розвитку інформаційної інфраструктури.



Рисунок 1.1 – Узагальнена структура інформаційного технологічного процесу



**Інформаційно-комунікаційна інфраструктура [Information and communication infrastructure]** – сукупність територіально розподілених державних і корпоративних інформаційних систем, ліній зв'язку, мереж і каналів передачі даних, засобів комутації й управління інформаційними потоками, а також організаційних структур, правових і нормативних механізмів, що забезпечують їхнє ефективне функціонування.

Відповідно до рівня розвитку комп'ютерної техніки і систем зв'язку можна виділити наступні організаційні форми їх використання, що тісно зв'язані з визначеними типами ІТ: централізована, децентралізована і розподілена.

**Централізована форма обробки інформації** зв'язана з появою великих могутніх ЕОМ у другій половині 60-х років ХХ століття й зосередженням процедур машинної обробки інформації в обчислювальних центрах (ОЦ). При цьому фахівець спілкувався з ЕОМ через посередника – працівника ОЦ, що мало свої негативні сторони (тривалість обробки, неоперативність у прийнятті рішень і т.п.).

Промисловий випуск персональних комп'ютерів обумовив появу **децентралізованої обробки інформації** у другій половині 70-х років ХХ сторіччя. Децентралізована обробка інформації припускає наявність ПК на робочому місці фахівця й активну його взаємодію з машиною в ході виконання своїх службових обов'язків. Процедури ІТ тепер виконуються безпосередньо на підприємствах і в організаціях. Відповідно до потреб окремих фахівців були створені діалогові людино-машинні комплекси – АРМ фахівців – для реалізації конкретних виробничих обов'язків. Були розроблені АРМ бухгалтера, економіста, менеджера, інженера і т.д.

У високо розвинених країнах із середини 80-х років ХХ століття стала розвиватися розподілена форма обробки інформації. Це зв'язано з визначними недоліками автономного використання ПК:

- дублювання частини інформації на різних комп'ютерах підприємства,
- труднощами перенесення значних обсягів інформації з одного ПК на інший,
- використанням надлишкового числа периферійних пристроїв і т.п.

**Розподілена обробка інформації** базується на використанні можливостей комп'ютерних мереж і сполучає переваги як централізованої, так і децентралізованої обробки даних. Мережні ІТ забезпечують вико-

ристання єдиної інформаційної бази, спільне розв'язання задач різними фахівцями, оперативність у пошуку й одержанні інформації і т.д.

В Україні у теперішній час також активно розвивається розподілена форма обробки інформації.

Інформація й, забезпечуючи її одержання й обробку технологічні процеси, є складовою частиною інформаційної системи підприємства, компанії, галузі і т.п.

**Інформаційні системи (ІС)** – це сукупність даних, необхідних для ефективного управління економічними об'єктами, а також використовуваної інформаційної технології в сукупності зі стосовними до них трудовими й фінансовими ресурсами організації (рисунок 1.2).

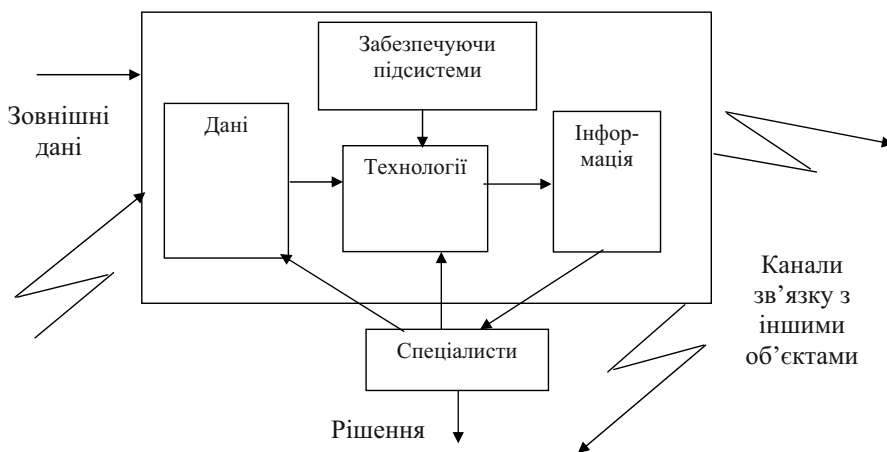


Рисунок 1.2 – Взаємозв'язок ІС, ІТ та інформаційної інфраструктури

Інформаційні системи мають складну ієрархічну структуру і складаються з окремих підсистем (складових частин), що забезпечують функціонування об'єкта в цілому. Будь-яка працююча складна ІС є результатом спільної роботи більш простих підсистем (функціональних – робоче середовище різних фахівців, і забезпечуючих – допоміжних для організації роботи ІС і її частин). Збій у роботі одного з елементів ІС може привести до збою роботи всієї інформаційної системи.

## 1.2 Технічні засоби обробки інформації

Обчислювальна техніка являє собою матеріальну основу інформаційних технологій, із її допомогою здійснюється збір, збереження, передача й обробка інформації. Сукупність пристроїв, призначених для автоматичної або автоматизованої обробки даних, називають **обчислювальною технікою**. Центральним ядром такої сукупності є ЕОМ.

**ЕОМ** – це електронний прилад, призначений для автоматизації створення, введення, збереження, обробки й видачі даних у формі, необхідної для подальшої роботи. У залежності від задач, що розв'язуються, обсягів і потоків інформації, яка обробляється, можуть використовуватися різні типи ЕОМ – від наймогутніших суперЕОМ (десятки мільярдів операцій у секунду й ціна десятки мільйонів доларів) до мікроЕОМ, вмонтованих у годинники, записні книжки, платіжні картки.

Класифікація ЕОМ звичайно здійснюється по таких основних параметрах як:

- кількість операцій, що виконуються у секунду,
- габарити комп'ютерів [1].

Границі класифікаційних угруповань досить умовні, тому що стрімкий науково-технічний прогрес в області електроніки й розвиток високотехнологій постійно поліпшують зазначені параметри і мінімізують розміри комп'ютерів.

### 1.2.1. Персональний комп'ютер і склад його системного блоку

Особливе місце в поширенні й використанні сучасних інформаційних технологій займає сімейство персональних комп'ютерів (ПК).

ПК забезпечують рішення освітніх, професійних, розважальних задач масового користувача – непрофесіонала в області комп'ютерних наук. У підприємствах, установах, фірмах персональними комп'ютерами або їхніми периферійними пристроями оснащуються робочі місця, із їхньою допомогою організується взаємодія персоналу, клієнтів, контрагентів. Тому кожен фахівець повинен орієнтуватися в можливостях ПК, його необхідному мінімальному обслуговуванні й використанні для професійних цілей. Можливості ПК обумовлені його технічними характеристиками й складом, наявним програмним забезпеченням.

В даний час найбільшу популярність і найширше визнання одержали IBM – сумісні комп'ютери; їхній випуск на кілька порядків перевищує випуск комп'ютерів інших сімейств і досягає десятків мільйонів екземплярів у рік. Серед виробників цих комп'ютерів можна зустріти як гігантів світового бізнесу (IBM, Comrag, Hewllet Packard), так і невеликі спеціалізовані фірми, у тому числі й українські.

Однією з основних причин успіху названих комп'ютерів є так називаний принцип відкритої архітектури, застосований у конструкції цих машин. Цей принцип передбачає можливість підключення нових компонентів або заміни старих без переробки самого комп'ютера. Наприклад, можна нарощувати оперативну пам'ять, замінити або додавати жорсткі диски, підключати модем, сканер і т.ін.

У цілому всі типи професійних персональних комп'ютерів універсальні, однак, має місце їхня орієнтація на ті або інші сфери застосування, наприклад, комп'ютери Macintosh займають видне місце у сфері видавництва. Ця орієнтація забезпечується як архітектурою комп'ютера, так і складом його програмного забезпечення.

Персональний комп'ютер типу IBM складається з окремих пристроїв, що поєднуються кабелями (зовнішніми або внутрішніми). Пристрої персонального комп'ютера поділяються на:

- системні й зовнішні,
- обов'язкові і необов'язкові.

Системні пристрої необхідні для функціонування комп'ютера і сховані від користувача: системна (материнська) плата, мікропроцесор, BIOS (апаратна частина базової системи введення-виведення), оперативна пам'ять, контролери, адаптери і т.п.

Зовнішні пристрої – це такі пристрої комп'ютера, із якими безпосередньо працює користувач. Вони поділяються на обов'язкові (клавіатура, монітор (дисплей), дисковод) і додаткові (миша, маніпулятор, модем, сканер, плотер).

Інакше можна сказати, що ПК включає базовий комплект, периферійні пристрої та інші технічні засоби, що орієнтують машину на універсальне й конкретне застосування. Структурна схема типового ПК представлена на рисунку 1.3.

Базовий комплект персонального комп'ютера складається з трьох основних частин:

- системного блоку,
- монітора (дисплея) і

- клавіатури.

При відсутності будь-якої з цих частин ПК непрацездатний. У стандартне постачання, яке формується продавцем, зараз обов'язково включається миша, без якої масовий користувач звичайно не працює.

**Системний блок** є основою комп'ютера. У певній мірі можна сказати, що він і є ПК. У ньому здійснюються всі обчислювальні операції. Він керує введенням-виведенням інформації, здійснює її обробку й збереження, а також керує всіма периферійними пристроями. Якщо розкрити корпус ПК, то ми побачимо, що там мається наступний мінімум пристроїв:

- материнська плата (Motherboard);
- оперативний запам'ятовуючий пристрій (ОЗП);
- адаптери (контролери) зовнішніх пристроїв;
- НМД (накопичувачі на магнітних дисках);
- блок живлення;
- генератор тактової частоти;
- чипсет – комплект мікросхем логіки, що підтримують роботу плати центрального процесора.

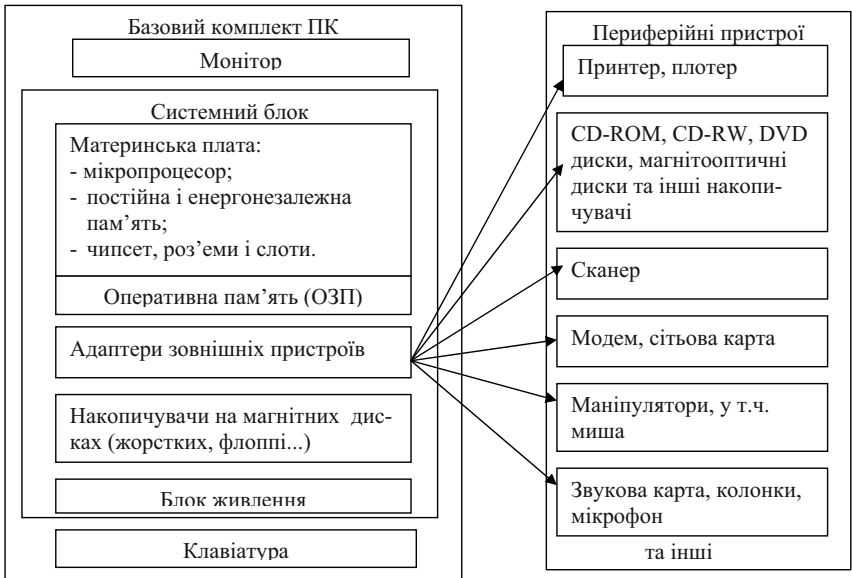


Рисунок 1.3 – Структурна схема ПК

**Материнська плата** (рисунок 1.4) – складний електронний пристрій.

На ній знаходиться мікропроцесор, постійна й енергонезалежна пам'ять, апаратна частина системи введення-виведення (BIOS), системна шина пам'яті, мікропроцесорний комплект (чипсет) – набір мікросхем, які керують роботою внутрішніх пристроїв комп'ютера й визначають основні функціональні можливості материнської плати; рознімання для підключення мікросхем (чипів) оперативної пам'яті, спеціальні рознімання (слоти) для підключення адаптерів зовнішніх пристроїв.

Персональні комп'ютери в цілому характеризуються рівнем, про який дають уявлення визначені параметри їхніх пристроїв: тип мікропроцесора, обсяг оперативної пам'яті, обсяг жорсткого диска (вінчестера), склад і параметри функціонування системних і зовнішніх пристроїв.

**Мікропроцесор** у комплекті з радіатором (металевим кожухом) і кулером (невеликим вентилятором) для охолодження мікросхеми здійснює всі обчислювальні і керуючі операції, тобто включає арифметико-логічний і керуючий пристрої. Швидкодія (тактова частота) і архітектура процесора (розрядність, наявність і ємність кеш – пам'яті, принцип виконання команд) визначають тип комп'ютера.

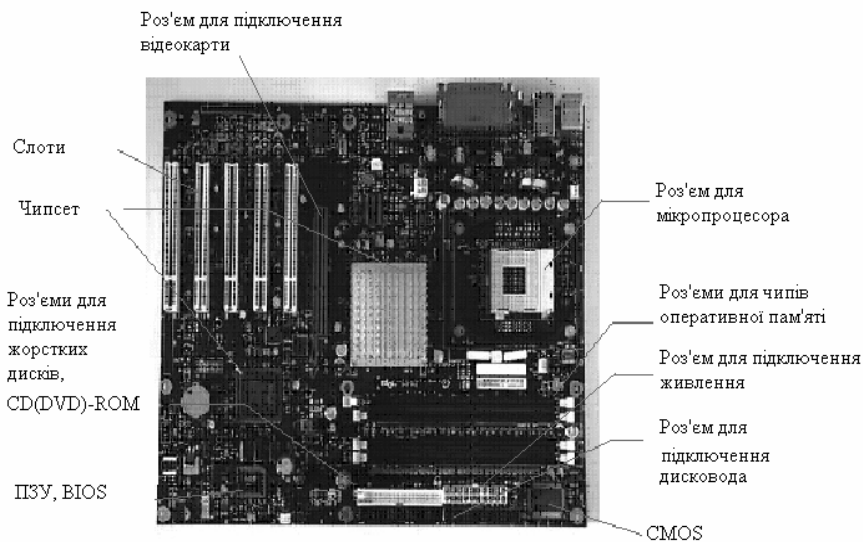


Рисунок 1.4 – Основні компоненти материнської плати

В даний час існують 32-, 64- і 128-бітові мікропроцесори, що обробляють «порції» даних відповідної довжини.

Мікропроцесор характеризується тактовою частотою, що виступає суфіксом у назві мікропроцесора. Типовими тактовими частотами мікропроцесорів є 600, 800, 1000, 1800, 2400 і більш Мгц (мегагерц), наприклад, Pentium IV. Для орієнтації Pentium II (тактова частота від 266 до 450 Мгц) має швидкість 263 млн. операцій у секунду.

Важливою характеристикою сучасних процесорів є ємність і швидкодія вмонтованої кеш-пам'яті. Внутрішня кеш-пам'ять являє собою набір регістрів – найшвидкіших пристроїв для короткострокового збереження й передачі на обробку найбільш часто використовуваних даних. Внутрішня кеш-пам'ять необхідна для підвищення продуктивності роботи процесора й узгодження роботи з більш повільно діючою оперативною пам'яттю. Ємність внутрішньої кеш-пам'яті 32 – 64 Кбайт.

На материнській платі комп'ютера мають пристрої, у яких зберігається інформація про склад і параметри устаткування. Вони ведуть облік часу й календар. Вони ж дають команду на завантаження інформації в оперативну пам'ять. Уміст цих пристроїв не губиться і при виключеному живленні. Апаратно вони виконані у вигляді спеціальних інтегральних схем: *ПЗП (постійний запам'ятовуючий пристрій)* і *CMOS (енергонезалежна пам'ять)*.

У ПЗП міститься спеціальний комплекс програм і функцій, що називається **BIOS (базова система введення-виведення)**. BIOS починає працювати відразу при включенні живлення комп'ютера, робить тестування всього устаткування ПК і дає команду на завантаження операційної системи.

Описуючи пристрої комп'ютера, не можна не згадати про **CMOS**. Справа в тім, що комп'ютер – дуже складний пристрій, у якого є десятки параметрів, що для правильного й оптимального функціонування комп'ютера повинні бути встановлені безпомилково (наприклад, параметри вінчестера й дисководу для гнучких дисків, системна дата, системний час). На більш ранніх комп'ютерах ці параметри виставлялися перемикачами на материнській платі й картах розширення або були фіксованими, що створювало незручності для тих користувачів, які часто (або принаймні часом) змінювали пристрої. Оперативна пам'ять комп'ютера для збереження цих параметрів не придатна: вона енергозалежна і після вимикання живлення вся інформація в ній руйнується. Тому

комп'ютер комплектується спеціальною енергонезалежною пам'яттю (CMOS), у якій і міститься інформація про все устаткування, установлене на комп'ютері, зберігаються всі установки, поточна дата й час. Ця інформація не зникає тому, що CMOS отримує живлення від окремого акумулятора. Якщо живлення CMOS зникне, то вся інформація буде загублена, і CMOS потрібно буде налаштувати вручну, але це зможе зробити тільки кваліфікований користувач.

**Системна шина** – сукупність пристроїв і сигнальних ліній, що служать для зв'язку мікропроцесора з різними компонентами комп'ютера. Її тип значною мірою визначає архітектуру системної плати і ПК і прямо зв'язаний зі швидкістю комп'ютера. У сучасних ПК системна шина поєднує процесор, оперативну пам'ять і кеш-пам'ять другого рівня, що розташовується на материнській платі, але поза процесором і має місткість від 1 до 8 Мбайт. Для збільшення швидкодії ПК використовуються **локальні шини**, що зв'язують мікропроцесор із контролерами периферійних пристроїв. Є наступні шини: ISA (16-бітна, найповільніша), EISA, VESA, PCI (32-бітна). Кожна із шин останніх трьох типів дозволяє підключати пристрої для ISA, але кількість таких пристроїв обмежена – для PCI – до 10 пристроїв.

У сучасному програмному забезпеченні переважає графічний інтерфейс, що вимагає великих ресурсів комп'ютера. Тому в ПК використовують спеціальну шину AGP різної пропускної здатності для підключення відеоадаптера. Сучасні моделі материнських плат підтримують AGP 8X, яка забезпечує швидкість до 2,1 Гб/с.

Широко використовується у сучасних комп'ютерах шина USB для підключення різних зовнішніх повільно діючих пристроїв (клавіатури, миші, принтерів, багатофункціональних пристроїв).

В даний час на ринку з'явилися шини PCI Express, що покликані замінити шину PCI і призначені для підключення найрізноманітнішої периферії (включаючи графічні прискорювачі). Шина PCI Express має дуже високу швидкість передачі даних у розрахунку на один сигнальний контакт – близько 100 Мбайт/с, що в 63 рази більше, ніж швидкість PCI, і в 5 разів більше, ніж швидкість AGP.

Прямий вплив на продуктивність комп'ютера має обсяг оперативного запам'ятовуючого пристрою або інакше оперативна пам'ять із довільною вибіркою RAM (Random Access Memory). **Оперативний запам'ятовуючий пристрій (RAM)** призначено для збереження даних і



програм, що безпосередньо беруть участь у рішенні задач. В ОЗП розміщаються службові й робочі програми, вхідні дані, константи, проміжні результати обчислень. ОЗП має велику швидкодію і забезпечує роботу арифметико-логічного пристрою, що є складовою частиною мікропроцесора. Ця пам'ять називається оперативною тому, що інформація в ній зберігається тільки при включеному живленні. Чим більше обсяг ОЗП, тим більше інформації в одиницю часу може обробити ПК. Апаратно ОЗП виконано у виді невеликих електронних плат – чипів (SIMM або DIMM), поміщених у спеціальних розніманнях на материнській платі. Їх розрізняють по числу контактів, виділяючи 32- і 72-контактні чипи. На сьогодні більш сучасний тип роз'ємних з'єднань – DIMM, що забезпечує час доступу не більше 7-15 нс ( $1 \text{ нс} = 10^{-9} \text{ с}$ ). Обсяг оперативної пам'яті комп'ютера може змінюватися в широких межах: від 1 МБ до 2 і більш Гб і забезпечувати швидкість до 12 Гбайт/с і більше для модулів SD-RAM, DDR, RD-RAM або RIMM, DDR SDRAM, XDR DRAM. В даний час використовуються чипи обсягом 128, 256, 512 Мб, 1 Гб, із яких можна скомпонувати різні об'єми оперативної пам'яті комп'ютера.

На материнській платі знаходяться спеціальні рознімання (слоти). У ці рознімання вставляються адаптери зовнішніх пристроїв. Слоти служать для з'єднання адаптерів із системною шиною комп'ютера.

**Адаптери (контролери) зовнішніх пристроїв (карти розширення)** – це апаратно – програмні пристрої для приєднання периферійних пристроїв.

Типовими картами розширення, що вставляються в рознімання материнської плати, є адаптер (відеокарта) дисплея, сільова карта, внутрішній модем, карта портів, карта контролера дисків.

З переходом від чорно-білих моніторів до кольорового і зі збільшенням вирішення екрана відбулося виділення всіх операцій, зв'язаних із керуванням екраном, в окремий блок, що одержав назву відеоадаптер. Фізично відеоадаптер виконано у вигляді окремої дочірньої плати, що вставляється у один із слотів материнської плати і називається відеокартою. Відеоадаптер узяв на себе функції відеоконтролера, відеопроцесора й відеопам'яті. Адаптер дисплея трансформує цифровий опис зображення екрана в сигнали розгорнення монітора, тому його характеристики тісно зв'язані з типом монітора. Тип відеоадаптера робить прямий вплив на комфортність роботи з комп'ютером. Адаптери розрізняють по наступних характеристиках:

1. Вирішення (роздільність) екрана – кількість крапок, що світяться на екрані, наприклад, 1280 x 768, 1280x1024 крапок і далі. Вирішення екрана є одним із найважливіших параметрів відеопідсистеми. Чим воно вище, тим більше інформації можна відобразити на екрані, але тем менше розмір кожної окремої крапки й тем менше видимий розмір елементів зображення. Використання завищеного вирішення на моніторі малого розміру приводить до того, що елементи зображення стають нерозбірливими і робота з документами й програмами викликає стомлення органів зору. Використання заниженого вирішення приводить до того, що елементи зображення стають великими, але на екрані їх розташовується дуже мало. Якщо програма має складну систему керування і велике число екранних елементів, вони не цілком містяться на екрані. Це приводить до зниження продуктивності праці і неефективній роботі.

Таким чином, для кожного розміру монітора існує своє оптимальне вирішення екрана, що повинен забезпечувати відеоадаптер

2. Кількість відображуваних кольорів (від 16 до 16 мільйонів). Максимально можливе колірне вирішення залежить від властивостей відеоадаптера і, у першу чергу, від кількості встановленої на ньому відеопам'яті. Крім того, воно залежить і від встановленого вирішення екрана. При високому вирішенні екрана на кожен точку зображення доводиться відводити менше місця у відеопам'яті, тому інформація про кольори вимушено виявляється більш обмеженою.

У залежності від заданого екранного вирішення й глибини кольору необхідний об'єм відеопам'яті можна визначити по наступній формулі:

$$P = \frac{(m \cdot n) \cdot b}{8}$$

де:

- P – необхідний об'єм пам'яті відеоадаптера;
- m – горизонтальне вирішення екрана (кількість крапок);
- n – вертикальне вирішення екрана (кількість крапок);
- b – розрядність кодування кольору (біт).

Мінімальна вимога по глибині кольору на сьогоднішній день – 256 кольорів, хоча більшість програм вимагають не менш 65 тис. кольорів (режим High Color). Найбільш комфортна робота досягається при глибині кольору 16,7 млн. кольорів (режим True Color).

Робота в повно кольоровому режимі True Color із високим екранним вирішенням вимагає значних розмірів відеопам'яті.

3. Можливість завантаження різних шрифтів у текстовому режимі.

4. Об'єм відеопам'яті (до 64 Мб) і її тип (DDR SDRAM або DDR RDRAM).

Розрізняють такі основні типи адаптерів дисплея й моніторів: CGA, EGA, Hercules, VGA, SuperVGA. Але це перерахування має лише історичне значення, тому що вже на початок 1992 року монітори й адаптери VGA (а на початок 1995 року – SuperVGA) практично витиснули всі інші типи.

Відеоприскорення – одна з властивостей відеоадаптера, що полягає в тому, що частина операцій по побудові зображень може відбуватися без виконання математичних обчислень в основному процесорі комп'ютера, а чисто апаратним шляхом – перетворенням даних у мікросхемах відеоприскорювача. Відеоприскорювачі можуть входити до складу відеоадаптера (у таких випадках говорять про те, що відеокарта має функції апаратного прискорення), але можуть поставлятися у вигляді окремої плати, що встановлюється на материнській платі і підключається до відеоадаптера.

Розрізняють два типи відеоприскорювачів – прискорювачі плоскої (2D) і тривимірної (3D) графіки. Перші найбільш ефективні для роботи з прикладними програмами (звичайно офісного застосування) і оптимізовані для операційної системи Windows, а другі орієнтовані на роботу мультимедійних розважальних програм, у першу чергу комп'ютерних ігор і професійних програм обробки тривимірної графіки. Звичайно в цих випадках використовують різні математичні принципи автоматизації графічних операцій, але існують прискорювачі, що володіють функціями і двовимірного, і тривимірного прискорення.

**Звукова карта** виявилася одним із найбільш пізніх удосконалень персонального комп'ютера. Вона підключається до одного зі слотів материнської плати у вигляді дочірньої карти і виконує обчислювальні операції, зв'язані з обробкою звуку, мови, музики. Звук відтворюється через зовнішні звукові колонки, що підключаються до виходу звукової карти. Спеціальне рознімання дозволяє відправити звуковий сигнал на зовнішній підсилювач. Є також рознімання для підключення мікрофона, що дозволяє записувати мову або музику і зберігати їх на жорсткому диску для наступної обробки й використання.

Основним параметром звукової карти є розрядність, що визначає кількість бітів, використовуваних при перетворенні сигналів з аналогової у цифрову форму і навпаки. Чим вище розрядність, тим менше погрішність, зв'язана з оцифровкою, тим вище якість звучання. Мінімальною вимогою сьогодення є 16 розрядів, а найбільше розповсюдження мають 32-розрядні і 64 – розрядні пристрої.

В області відтворення звуку найскладніше обстоїть справа зі стандартизацією. Відсутність єдиних централізованих стандартів привело до того, що ряд фірм, що займаються випуском звукового устаткування, де-факто ввели в широке використання свої внутріфірмові стандарти. Так, наприклад, у багатьох випадках стандартними вважають пристрої, сумісні з пристроєм Sound Blaster, права на який належать компанії Creative Labs.

### 1.2.2 Зовнішні пристрої ПК

**Дисплей** (монітор) призначений для виводу інформації. Його основними споживчими параметрами є: розмір і крок маски екрана, максимальна частота регенерації зображення, клас захисту.

Розмір монітора вимірюється між протилежними кутами трубки кінескопа по діагоналі. Одиниця виміру – дюйми. Стандартні розміри: 14"; 15"; 17"; 19"; 20"; 21". В даний час найбільш універсальними є монітори розміром 15 і 17 дюймів, а для операцій із графікою бажані монітори розміром 19-21 дюйм.

<b>Розмір</b>	<b>Оптимальне вирішення</b>
14 дюймів дюй-	640x480
15 дюймів	800x600
17 дюймів	1024x768
19 дюймів	1280x1024

Зображення на екрані монітора виходить у результаті опромінення люмінофорного покриття гостроспрямованим пучком електронів, розігнаних у вакуумній колбі. Для одержання кольорового зображення люмінофорне покриття має крапки або смужки трьох типів, що світяться червоним, зеленим і синім кольором. Щоб на екрані всі три промені сходи-

лися строго в одну крапку й зображення було чітким, перед люмінофором ставлять маску – панель із регулярно розташованими отворами або щілинами. Частина моніторів оснащена маскою з вертикальних дротиків, що підсилює яскравість і насиченість зображення. Чим менше крок між отворами або щілинами (крок маски), тим чіткіше й точніше отримане зображення. Крок маски вимірюють у частках міліметра. В даний час найбільш поширені монітори з кроком маски 0,25-0,27 мм. Застарілі монітори можуть мати крок до 0,43 мм, що негативно позначається на органах зору при роботі з комп'ютером. Моделі підвищеної вартості можуть мати значення менш 0,25 мм.

Частота регенерації (відновлення) зображення показує, скільки разів протягом секунди монітор може цілком змінити зображення (тому її також називають частотою кадрів). Цей параметр залежить не тільки від монітора, але і від властивостей і налаштувань відеоадаптера, хоча граничні можливості визначає все-таки монітор.

Частоту регенерації зображення вимірюють у герцах (Гц). Чим вона вище, тим чіткіше й стійкіше зображення, тим менше стомлення очей, тим більше часу можна працювати з комп'ютером безупинно. При частоті регенерації порядку 60 Гц дрібне мерехтіння зображення помітно незброєним оком. Сьогодні таке значення вважається неприпустимим. Мінімальним вважають значення 75 Гц, нормативним – 85 Гц і комфортним – 100 Гц і більш.

Клас захисту монітора визначається стандартом, якому відповідає монітор із погляду вимог техніки безпеки. В даний час загальноновизнаними вважаються наступні міжнародні стандарти: MPR-II, TCO-92, TCO-95, TCO-99, TCO-03 (приведені в хронологічному порядку). Стандарт TCO-03 установив найжорсткіші норми по параметрах, що обумовлює якість зображення (яскравість, контрастність, мерехтіння, антиблікові властивості покриття) і обмеження на рівні електромагнітного випромінювання.

В даний час звичайні монітори усе активніше витісняються LCD моніторами (із жидкокристалічним екраном), у яких абсолютно плоский екран забезпечує відсутність геометричних перекручувань.

IBM – сумісний комп'ютер звичайно має одне рознімання рівнобіжного порту (LPT1), два рознімання послідовних портів (COM1 і COM2), 2 або 4 послідовних порти USB, що виводяться назовні (рисунок 1.5). До рівнобіжного порту звичайно приєднується принтер, а до одного з послідовних – миша. До послідовного порту також може приє-

днуватися зовнішній модем. До USB портів можуть підключатися миша, клавіатура, багатофункціональні пристрої, відеокамера і т.д. – усі залежить від наявних рознімів і сполучних проводів.



Рисунок 1.5 – Рознімання портів персонального комп'ютера

Обов'язковим пристроєм комп'ютера є **клавіатура**. Її основні функції – введення алфавітно-цифрових даних, а також команд управління – не мають потреби в підтримці спеціальними системними програмами (драйверами). Необхідне програмне забезпечення для початку роботи з комп'ютером уже є в мікросхемі ПЗП (постійний запам'ятовуючий пристрій) у складі базової системи введення-виведення (BIOS), і тому комп'ютер реагує на натискання клавіш відразу після включення. Проте для роботи з різними розкладками (алфавітами) необхідні відповідні драйвери.

Принцип дії клавіатури полягає в наступному.

1. При натисканні на клавішу (або комбінацію клавіш) спеціальна мікросхема, убудована в клавіатуру, видає так називаний скан-код.
2. Скан-код надходить у мікросхему, що виконує функції порту клавіатури. (Порти – спеціальні апаратно-логічні пристрої, що відповідають за зв'язок процесора з іншими пристроями.) Дана мікросхема знаходиться на основній платі комп'ютера усередині системного блоку.
3. Порт клавіатури видає процесорові переривання з фіксованим номером. Для клавіатури номер переривання – 9 (Interrupt 9, Int9).
4. Одержавши переривання, процесор відкладає поточну роботу і по номеру переривання звертається в спеціальну область оперативної пам'яті, у якій знаходиться так називаний вектор переривань. Вектор переривань – це список адресних даних із фіксованою довжиною запису.

Кожен запис містить адресу програми, що повинна обслужити переривання з номером, що збігається з номером запису.

5. Визначивши адресу початку програми, що обробляє виникле переривання, процесор переходить до її виконання. Найпростіша програма обробки клавіатурного переривання «захита» у мікросхемі ПЗП, але програмісти можуть «підставити» замість неї свою програму, якщо змінять дані у векторі переривань.

6. Програма-оброблювач переривання направляє процесор до порту клавіатури, де він знаходить скан-код, завантажує його у свої регістри, потім під управлінням оброблювача обумовлює, який код символу відповідає даному скан-коду.

7. Далі оброблювач переривань відправляє отриманий код символу в невелику область пам'яті, відому як буфер клавіатури, і припиняє свою роботу, сповістивши про це процесор.

8. Процесор припиняє обробку переривання і повертається до відкладеної задачі.

9. Уведений символ зберігається в буфері клавіатури доти, поки його не забере відтіля та програма, для якої він і призначався, наприклад, текстовий редактор. Якщо символи надходять у буфер частіше, ніж забираються відтіля, настає ефект переповнення буфера. У цьому випадку введення нових символів на якийсь час припиняється. На практиці в цей момент при натисканні на клавішу ми чуємо попереджуючий звуковий сигнал і не спостерігаємо введення даних.

Клавіатура (пульт управління й індексації) ПК складається з таких груп клавіш:

функціональні клавіші (F1-F12) знаходяться у верхній частині клавіатури у виді трьох тетрад;

алфавітно-цифрові клавіші – основне поле клавіш білого кольору в лівій частині клавіатури;

цифрові (калькуляторна клавіатура) – група з 17 клавіш у правій частині клавіатури;

службові і клавіші управління переміщенням курсору – група клавіш сірого кольору, розташованих біля алфавітно-цифрових.

Крім цього на клавіатурі в правому верхньому куті розташовані індикатори включення режимів: "Num Lock", "Caps Lock" і "Scroll Lock".

Функціональні клавіші можуть бути запрограмовані, виконуючи в різних програмах різні функції (наприклад, **[F1]** у багатьох програмах викликає довідкову інформацію).

Алфавітно-цифрові клавіші використовують для введення текстової інформації. Великі букви і спеціальні символи вводяться при одночасному натисканні **[Shift]** і потрібної клавіші. **[Caps Lock]** працює як кнопковий перемикач і дозволяє вводити тільки великі букви, але ніяк не впливає на клавіші зі спеціальними символами. Індикатор "Caps Lock" горить, якщо клавіатура знаходиться в стані "Caps Lock". **[Tab]** – табуляція служить для швидкого переміщення курсору по позиціях табуляції (стандартно на 8 позицій). **[Esc]** часто служить для виходу з режимів у різних програмах або для скасування чого-небудь. **[Ctrl]** і **[Alt]** самостійно не діють, а працюють тільки разом із натиснутою іншою клавішею. **[Backspace]** – стрілка вліво в правій верхній частині алфавітно-цифрової клавіатури – переміщує курсор на одну позицію назад (уліво) і витирає символ, що стоїть перед курсором.

Калькуляторна клавіатура використовується при уведенні великих масивів цифрової інформації. Для переключення в режим цифр натискають **[Num Lock]** (загоряється відповідний індикатор). Повторне натискання **[Num Lock]** переводить цифрову клавіатуру в режим керування курсором.

Службові клавіші **[Print Screen]**, **[Scroll Lock]** і **[Pause]** використовуються в основному в комбінації з **[Shift]**, **[Ctrl]** і **[Alt]**. Серед таких комбінацій є:

**[Shift + Print Screen]** – друкувати зображення екрана в одному екземплярі;

**[Ctrl + Print Screen]** – синхронний вивід даних на дисплей і принтер;

**[Ctrl + Pause]** (**[Ctrl + Break]**) – припиняє дії команди.

**[←]**, **[↑]**, **[→]**, **[↓]** символи для переміщення курсору на екрані. **[Home]** переміщує курсор у "початок" (лівий верхній кут екрана або початок рядка). **[End]** – у "кінець" рядка. **[PgUp]** або **[PgDn]** – на одну сторінку догори або униз відповідно. **[Insert]** служить для переключення режимів вставки і заміни при введенні даних. Натискання **[Delete]** приводить до вилучення символу в позиції курсору й переміщення залишеної частини рядка вліво.

Найбільше часто застосовуються такі комбінації клавіш:



**[Ctrl + Scroll Lock]** – переводить ПК у режим паузи, тобто ПК припиняє виконання програми до натискання будь-якої клавіші.

**[Ctrl + Alt + Delete]** – виконує скидання системи і її перезавантаження;

**[Ctrl + End]** (**[Home]**, **[←]**, **[→]**) – переміщає курсор у кінець (початок) тексту або вліво (вправо) на одне слово.

**Миша** – пристрій керування манипуляторного типу. Являє собою плоску коробочку з двома-трьома кнопками. Переміщення миші по плоскій поверхні синхронізовано з переміщенням графічного об'єкта (показника миші) на екрані монітора.

Принцип дії. На відміну від розглянутої раніше клавіатури, миша не є стандартним органом керування, і персональний комп'ютер не має для неї виділеного порту. Для миші немає і постійного виділеного переривання, а базові засоби введення й виводу, що розміщені в постійному запам'ятовуючому пристрої, (ПЗП) BIOS комп'ютера, не містять програмних засобів для обробки переривань миші.

У зв'язку з цим у перший момент після включення комп'ютера миша не працює. Вона потребує підтримки спеціальної системної програми – драйвера миші. Драйвер устанавлюється або при першому підключенні миші, або при установці операційної системи комп'ютера. Хоча миша і не має виділеного порту на материнській платі, для роботи з нею використовують один із стандартних портів, засоби для роботи, із якими є в складі BIOS. Драйвер миші призначений для інтерпретації сигналів, що надходять через порт. Крім того, він забезпечує механізм передачі інформації про положення й стан миші операційній системі і працюючим програмам.

Користувач керує переміщенням миші по площині і короткочасним натисканням правої й лівої кнопок. (Ці натискання називаються клацаннями.) Переміщення миші й клацання її кнопок є подіями з погляду її програми-драйвера. Аналізуючи ці події, драйвер устанавлює, коли відбулася подія, і в якому місці екрана в цей момент знаходився показник. Ці дані передаються в прикладну програму, із яким працює користувач у даний момент. По них програма може визначити команду, що мав на увазі користувач, і приступити до виконання.

Комбінація монітора й миші забезпечує найбільш сучасний тип інтерфейсу користувача, що називається графічним. Користувач спостерігає на екрані графічні об'єкти й елементи управління. За

допомогою миші він змінює властивості об'єктів і пускає в хід елементи управління комп'ютерною системою, а за допомогою монітора одержує від неї відгук у графічному виді.

Вся інформація, з якою працює користувач, звичайно, зберігається на *жорсткому диску (вінчестері)*, що розміщується в середині системного блоку й кабелями приєднується до контролера дисків (розміщеному на материнській платі або на карті розширення). Вінчестер характеризується обсягом і швидкістю пошуку інформації. Обсяги вінчестерів коливаються від 2 МБ до 200 і більш Гб.

Вінчестер зовні представляє собою герметично запаяну коробку. Усередині неї розміщені пакети дисків, магнітні голівки, що записують і зчитують інформацію, і електродвигуни, що розкручують диск і переміщують голівки. Диски вінчестера обертаються увесь час поки комп'ютер включений. Поверхня диска розділена на сектори й доріжки. Над кожною поверхнею розташовується голівка, призначена для читання, запису даних. При високих швидкостях обертання дисків (90 об/с) у зазорі між голівкою і поверхнею утворюється аеродинамічна подушка, і голівка парить над магнітною поверхнею на висоті, що складає трохи тисячних часток міліметра. При зміні сили струму, що протікає через голівку, відбувається зміна напруженості динамічного магнітного поля в зазорі, що викликає зміни в стаціонарному магнітному полі феромагнітних часток, що утворюють покриття диска. Так здійснюється запис даних на магнітний диск.

Операція зчитування відбувається у зворотному порядку. Намагнічені частки покриття, що проносяться на високій швидкості поблизу голівки, наводять у ній ЕДС самоіндукції. Електромагнітні сигнали, що виникають при цьому, підсилюються й передаються на обробку.

Управління роботою жорсткого диска виконує спеціальний апаратно-логічний пристрій – контролер жорсткого диска. У минулому він уявляв собою окрему дочірню плату, що підключали до одного з вільних слотів материнської плати. В даний час функції контролерів дисків виконують мікросхеми, що входять у мікропроцесорний комплект (чипсет), хоча деякі види високопродуктивних контролерів жорстких дисків як і раніше поставляються на окремій платі.

До основних параметрів жорстких дисків відносяться місткість і продуктивність. Місткість дисків залежить від технології їхнього виготовлення. В даний час більшість виробників жорстких дисків використовув-

ють винайдену компанією IBM технологію з використанням гігантського магніторезистивного ефекту ( GMR – Giant Magnetic Resistance). Теоретична межа місткості однієї пластини, виконаної за цією технологією, складає порядку 20 Гбайт. В даний час досягнутий технологічний рівень 6,4 Гбайт на пластину, але розвиток продовжується.

З іншого боку, продуктивність жорстких дисків менше залежить від технології їхнього виготовлення. Сьогодні всі жорсткі диски мають дуже високий показник швидкості внутрішньої передачі даних (до 30-60 Мбайт/с), і тому їхня продуктивність у першу чергу залежить від характеристик інтерфейсу, за допомогою якого вони зв'язані з материнською платою. У залежності від типу інтерфейсу розкид значень може бути дуже великим: від декількох Мбайт/с до 133 Мбайт/с для інтерфейсів типу IDE; до 300 Мбайт/с для інтерфейсів типу SATA; до 320 Мбайт/с для інтерфейсів типу SCSI; від 50 Мбайт/с і більш для сучасних інтерфейсів типу IEEE 1394.

Крім швидкості передачі даних із продуктивністю диска прямо зв'язаний параметр середнього часу доступу. Він визначає інтервал часу, необхідний для пошуку потрібних даних, і залежить від швидкості обертання диска. Для дисків із частотою обертання 5400 об/хв, середній час доступу складає 9-10 мкс, для дисків із частотою 7200 об/хв – 7-8 мкс. Вироби більш високого рівня забезпечують середній час доступу до даних 5-6 мкс.

Працюючи з ПК, користувачеві не потрібно здійснювати елементарні операції з диском і клавіатурою. Для цього служить спеціальна програма.

Для збереження великих обсягів інформації використовуються *зовнішні запам'ятовуючі пристрої (ВЗП)*, які мають меншу швидкість у порівнянні з ОЗП. У накопичувачах гнучких магнітних дисків (НГМД) носієм інформації є *дискета*. Це гнучкий пластик у формі диска, покритий магнітним шаром і поміщений у твердий пластмасовий конверт для захисту від фізичних ушкоджень. У персональному комп'ютері IBM використовуються 3,5-дюймові дискети обсягом 1,44 Мб – 2 Мб. Завдяки тому що дискета може існувати окремо від дисководу, можна використовувати в роботі кілька дискет, кожна з яких містить різноманітну інформацію і зчитується на тому самому дисководі. Дискета вставляється в щілину отвору дисководу (до складу системного пристрою IBM-сумісного комп'ютера входять два дисководи), механізм приводу захоплює гнучкий пластиковий диск за центральний отвір і

обертає його в середині конверта. Доступ до інформації здійснюється через прорізи в захисному конверті.

Щоб використовувати дискету для збереження інформації, необхідно її відформатувати або, іншими словами, відповідним чином розмітити – розбити на доріжки й сектори. Для цієї мети служать спеціальні програми. Дискети 3,5 дюйми можуть формуватися на 1,44 Мб. Крім того, можна відформатувати дискети на нестандартні формати – 1,7; 2 Мб.

**Периферійні пристрої** персонального комп'ютера підключаються до його інтерфейсів і призначені для виконання допоміжних операцій. Завдяки їм комп'ютерна система здобуває гнучкість і універсальність.

По призначенню периферійні пристрої можна підрозділити на:

- пристрої введення даних;
- пристрої виводу даних;
- пристрої збереження й захисту даних;
- пристрої обміну даними.

### **Пристрої введення знакових даних**

**Спеціальні клавіатури.** Клавіатура є основним пристроєм уведення даних. Спеціальні клавіатури призначені для підвищення ефективності процесу введення даних. Це досягається шляхом зміни форми клавіатури, розкладки її клавіш або методу підключення до системного блоку.

Клавіатури, що мають спеціальну форму, розраховану з урахуванням вимог ергономіки, називають ергономічними клавіатурами. Їх доцільно застосовувати на робочих місцях, призначених для уведення великої кількості знакової інформації. Ергономічні клавіатури не тільки підвищують продуктивність користувача і знижують загальне стомлення протягом робочого дня, але і знижують імовірність і ступінь розвитку ряду захворювань, наприклад, тунельного синдрому кистей рук і остеохондроза верхніх відділів хребта. Розкладка клавіш стандартних клавіатур далека від оптимальної. Вона збереглася з часів ранніх зразків механічних пишучих машин. В даний час існує технічна можливість виготовлення клавіатур з оптимізованою розкладкою, і існують зразки таких пристроїв (зокрема, до них відноситься клавіатура Дворака). Однак практичне впровадження клавіатур із нестандартною розкладкою знаходиться під сумнівом у зв'язку з тим, що роботі з ними треба учити-

ся спеціально. На практиці подібними клавіатурами оснащують тільки спеціалізовані робочі місця.

По методу підключення до системного блоку розрізняють провідні й безпроводні клавіатури. Передача інформації у безпроводних системах здійснюється інфрачервоним променем. Звичайний радіус дії таких клавіатур складає кілька метрів. Джерелом сигналу є клавіатура.

### **Пристрої командного управління**

**Спеціальні маніпулятори.** Крім звичайної миші існують і інші типи маніпуляторів, наприклад, трекболи, пенмауси, інфрачервоні миші.

Трекбол на відміну від миші встановлюється стаціонарно, і його кулька приводиться в рух долонею руки. Перевага трекбола полягає в тому, що він не має потреби в гладкій робочій поверхні, тому трекболи знайшли широке застосування в портативних персональних комп'ютерах.

Пенмаус являє собою аналог кулькової авторучки, на кінці якої замість пишучого вузла установлений вузол, що реєструє величину переміщення.

Інфрачервона миша відрізняється від звичайної наявністю пристрою безпроводного зв'язку із системним блоком.

Для комп'ютерних ігор і в деяких спеціалізованих імітаторах застосовують також маніпулятори важільно-натискного типу (джойстики) і аналогічні їм джойпади, геймпади і штурвально-педальні пристрої. Пристрої цього типу підключаються до спеціального порту, що мається на звуковій карті, або до порту USB.

### **Пристрої введення графічних даних**

Для введення графічної інформації використовують сканери, графічні планшети (дигітайзери) і цифрові фотокамери. Цікаво відзначити, що за допомогою сканерів можна вводити і знакову інформацію. У цьому випадку вхідний матеріал уводиться в графічному вигляді, після чого обробляється спеціальними програмними засобами (програмами розпізнавання образів).

Розрізняють планшетні, ручні і барабанні сканери, кольорові і чорно-білі.

**Планшетні сканери.** Планшетні сканери призначені для введення графічної інформації з прозорого або непрозорого листового матеріалу.

Принцип дії цих пристроїв полягає в тому, що промінь світла, відбитий від поверхні матеріалу (або який пройшов крізь прозорий матеріал), фіксується спеціальними елементами, названими приладами із рядовим зв'язком (ПЗЗ). Звичайно елементи ПЗЗ конструктивно оформлюють у вигляді лінійки, розташовуваної по ширині вхідного матеріалу. Переміщення лінійки щодо листа папера виконується механічним протяганням лінійки при нерухомій установці листа або протяганням листа при нерухомій установці лінійки.

Основними споживчими параметрами планшетних сканерів є:

- роздільна здібність;
- продуктивність;
- динамічний діапазон;
- максимальний розмір матеріалу, що сканується.

Роздільна здібність залежить від щільності розміщення приладів ПЗЗ на лінійці, а також від точності механічного позиціонування лінійки при скануванні. Типовий показник для офісного застосування: 600-4200 dpi (dpi – dots per inch – кількість крапок на дюйм). Для професійного застосування характерні показники 1200-3000 dpi.

Продуктивність сканера визначається тривалістю сканування листа папера стандартного формату і залежить як від досконалості механічної частини пристрою, так і від типу інтерфейсу, використаного для сполучення з комп'ютером.

Динамічний діапазон визначається логарифмом відносини яскравості найбільш світлих ділянок зображення до яскравості найбільш темних ділянок. Типовий показник для сканерів офісного застосування складає 1,8-2,0, а для сканерів професійного застосування – від 2,5 (для непрозорих матеріалів) до 3,5 (для прозорих матеріалів).

**Ручні сканери.** Принцип дії ручних сканерів в основному відповідає планшетним. Різниця полягає в тім, що протягання лінійки ПЗЗ у даному випадку виконується вручну. Рівномірність і точність сканування при цьому забезпечуються незадовільно, і роздільна здатність ручного сканера складає 150-300 dpi.

**Барабанні сканери.** У сканерах цього типу вхідний матеріал закріплюється на циліндричній поверхні барабана, що обертається з високою швидкістю. Пристрої цього типу забезпечують найвищу роздільну здатність (2400-5000 dpi) завдяки застосуванню не

ПЗЗ, а фото-електронних множників. Їх використовують для сканування вхідних зображень, що мають високу якість, але недостатні лінійні розміри (фотонегативів, слайдів і т.п.)

**Сканери форм.** Призначені для введення даних із стандартних форм, заповнених механічно або «від руки». Необхідність у цьому виникає при проведенні переписів населення, обробці результатів виборів і аналізі анкетних даних.

Від сканерів форм не потрібно високої точності сканування, але швидкодія відіграє підвищену роль і є основним споживчим параметром.

**Штрих – сканери.** Цей різновид ручних сканерів призначений для введення даних, закодованих у вигляді штрих – коду. Такі пристрої мають застосування в роздрібній торговельній мережі.

**Графічні планшети (дигітайзери).** Ці пристрої призначені для введення художньої графічної інформації. Існує декілька різних принципів дії графічних планшетів, але в основі всіх них лежить фіксація переміщення спеціального пера щодо планшета. Такі пристрої зручні для художників і ілюстраторів, оскільки дозволяють їм створювати екранні зображення звичними прийомами, напрацьованими для традиційних інструментів (олівець, перо, кисть).

**Цифрові фотокамери, Web-камери.** Як і сканери, ці пристрої сприймають графічні дані за допомогою приборів із зарядовим зв'язком (ПЗЗ), об'єднаних у прямокутну матрицю. Основним параметром цифрових фотоапаратів є роздільна здатність, що прямо зв'язана з кількістю чарунок (пікселів) ПЗЗ у матриці. Найкращі споживчі моделі в даний час мають до 8 мегапікселів у ПЗЗ і, відповідно, забезпечують вирішення зображення до 2448 x 3264 крапок. У професійних моделях ці параметри вище.

**Пристрої введення звукової і відео інформації** – мікрофони, тунери (для введення даних із телевізора).

### Пристрої виводу даних

Як пристрої виводу даних, додаткових до монітора, використовують друкувальні пристрої (принтери), що дозволяють одержувати копії документів на папері або прозорому носії. За принципом дії розрізняють матричні, лазерні, світлодіодні і струминні принтери.

**Матричні принтери.** Це найпростіші друкувальні пристрої. Дані виводяться на папір у виді відбитка, що утвориться при ударі циліндричних стрижнів ("голок") через барвну стрічку. Якість друку матричних принтерів прямо залежить від кількості голок у друкуючій голівці. Найбільше поширення мають 9-голчасті і 24-голчасті матричні принтери. Останні дозволяють одержувати відбитки документів, що не уступають по якості документам, виконаним на друкарській машинці.

Продуктивність роботи матричних принтерів оцінюють по кількості знаків, що друкуються у секунду (cps – characters per second). Звичайними режимами роботи матричних принтерів є: draft – режим чорного друку, normal – режим звичайного друку й режим NLQ (Near Letter Quality), що забезпечує якість друку, близьку до якості друкарської машинки. Матричні принтери найдешевші, дозволяють за один раз друкувати під копірку кілька копій документів, але повільні, шумні і самі ненадійні в роботі. Вони можуть мати широку й вузьку каретку для роботи з папером різного розміру.

**Лазерні принтери.** Лазерні принтери забезпечують високу якість друку, що не поступається, а в багатьох випадках і переважає поліграфічну. Вони відрізняються також високою швидкістю друку, що виміряється в сторінках у хвилину (ppm – page per minute). Як і в матричних принтерах, підсумкове зображення формується з окремих крапок.

Принцип дії лазерних принтерів наступний:

відповідно до даних, що надходять, лазерна голівка випускає світлові імпульси, що відбиваються від дзеркала і попадають на поверхню світлочутливого барабана;

горизонтальне розгорнення зображення виконується обертанням дзеркала;

ділянки поверхні світлочутливого барабана, що одержали світловий імпульс, здобувають статичний заряд;

барабан при обертанні проходить через контейнер, наповнений барвним складом (тонером), і тонер закріплюється на ділянках, що мають статичний заряд;

при подальшому обертанні барабана відбувається контакт його поверхні з паперовим листом, у результаті чого відбувається перенос тонера на папір;

лист паперу з нанесеним на нього тонером протягається через нагрівальний елемент, у результаті чого частки тонера спікаються й закріплюються на папері.



До основних параметрів лазерних принтерів відносяться: роздільна здатність (вирішення); продуктивність (сторінок у хвилину); формат використовуваного папера; обсяг власної оперативної пам'яті.

При виборі лазерного принтера необхідно також враховувати параметр вартості відбитка, тобто вартість видаткових матеріалів для одержання одного друкованого аркуша стандартного формату А4. До видаткових матеріалів відноситься тонер і барабан, що після друку визначеної кількості відбитків утрачає свої властивості. Як одиницю виміру використовують вартість друку однієї сторінки.

Основна перевага лазерних принтерів полягає в можливості одержання високоякісних відбитків. Моделі середнього класу забезпечують вирішення друку до 600 dpi, а професійні моделі – до 1200 dpi.

**Світлодіодні принтери.** Принцип дії світлодіодних принтерів схожий на принцип дії лазерних принтерів. Різниця полягає в тім, що джерелом світла є не лазерна голівка, а лінійка світлодіодів. Оскільки ця лінійка розташована по всій ширині сторінки, що друкується, відпадає необхідність у механізмі формування горизонтального розгорнення і вся конструкція виходить простіше, надійніше й дешевше. Типова величина вирішення друку для світлодіодних принтерів складає порядку 600 dpi.

**Струминні принтери.** У струминних друкувальних пристроях зображення на папері формується з плям, що утворюються при влученні крапель барвника на папір. Викид мікрокрапель барвника відбувається під тиском, що розвивається в друкуючій голівці за рахунок паротворення. У деяких моделях крапля викидається клацанням у результаті п'єзоелектричного ефекту – цей метод дозволяє забезпечити більш стабільну форму краплі, близьку до сферичного.

Якість друку зображення багато в чому залежить від форми краплі і її розміру, а також від характеру усмоктування рідкого барвника поверхнею папера. У цих умовах особливу роль грають в'язкісні властивості барвника й властивості папера.

До позитивних властивостей струминних друкувальних пристроїв варто віднести відносно невелику кількість механічних частин, що рухаються, і, відповідно, простоту й надійність механічної частини пристрою і його відносно низьку вартість. Основним недоліком, у порівнянні з лазерними принтерами, є нестабільність одержуваного вирішення, що обмежує можливість їхнього застосування в чорно-білому напівтоновому друці.

У той же час, сьогодні струминні принтери знайшли дуже широке застосування в кольоровому друці. Завдяки простоті конструкції вони

набагато перевершують кольорові лазерні принтери по показнику якість/ціна. При вирішенні вище 600 dpi вони дозволяють одержувати кольорові відбитки, що перевершують по якості кольорові відбитки, одержувані фотохімічними методами.

При виборі струминного принтера варто обов'язково мати у виду параметр вартості друку одного відбитка. При тім, що ціна струминних друкувальних пристроїв помітно нижче, ніж лазерних, вартість друку одного відбитка на них може бути в кілька разів вище.

**Плотери і графобудівники**, призначені для виготовлення креслень, діаграм і малюнків на папері різного формату. Плотери характеризуються числом кольорів, якими може бути виконане креслення, і форматом папера (A1, A2, A3, A4, A5 і ін.). Плотери бувають планшетного й барабанного типу.

## Пристрої збереження й захисту даних

Необхідність у зовнішніх пристроях збереження даних виникає у двох випадках

- коли на обчислювальній системі обробляється більше даних, чим можна розмістити на базовому жорсткому диску;
- коли дані мають підвищену цінність і необхідно виконувати регулярне резервне копіювання на зовнішній пристрій (копіювання даних на жорсткому диску не є резервним і тільки створює ілюзію безпеки).

В даний час для зовнішнього збереження даних виробляють кілька типів пристроїв, що використовують магнітні або магнітооптичні носії.

**Стримери** – це накопичувачі на магнітній стрічці. Їх відрізняє порівняно низька ціна. До недоліків стримерів відносять малу продуктивність (вона зв'язана, насамперед, із тим, що магнітна стрічка – це пристрій послідовного доступу) і недостатню надійність (крім електромагнітних наведень, стрічки стримерів спитують підвищені механічні навантаження і можуть фізично виходити з ладу).

Місткість магнітних касет (картриджів) для стримерів складає до кількох сотень Мбайт. Подальше підвищення місткості за рахунок підвищення щільності запису знижує надійність збереження, а підвищення місткості за рахунок збільшення довжини стрічки стримується низьким часом доступу до даних.

Пристрій зчитування з компакт-дисків (**CDROM**). Серед компакт-дисків виділяють наступні види:

- **CD-ROM** – Compact Disk-Read Only Memory (Компакт-диск – тільки для читання). Компакт-диск, що виробляється на спеціальному заводському устаткуванні методом штампування;

- **CD-R** – Compact Disk-Recordable (Записуваний компакт-диск). Записувані CD відносяться до носіїв Write Once, Read Multiple – однократного запису, багаторазового читання. Для запису використовуються спеціальні пристрої і спеціальне програмне забезпечення, а відтворюються вони звичайними пристроями читання компакт-дисків;

- **CD-RW**- Compact Disk-Re Writable (Перезаписуваний компакт-диск). Диск, на який можна не тільки записувати, але і перезаписувати дані.

Компакт-диски мають малі розміри і велику місткість (до 700 Мбайт), надійні й довговічні. Інформація на них записується й зчитується за допомогою лазерного променя.

Існує ще одна технологія збереження даних, називана **DVD**. Раніше абревіатура DVD означала *Digital Video Disc*, оскільки такі диски призначалися, у першу чергу, для збереження фільмів високої якості. В даний час ця абревіатура розшифровується як *Digital Versatile Disc* – цифровий універсальний диск. DVD-диски по розмірі і зовнішньому вигляді практично не відрізняються від звичайних компакт-дисків, однак, завдяки удосконаленій технології двостороннього запису, ємність їх збільшилась до 17 Гб інформації.

Для роботи з компакт-дисками перерахованих вище типів існують спеціальні приводи:

*CD Drive* – привод для читання компакт-дисків CD-ROM, записуваних дисків CD-R і перезаписуваних дисків CD-RW;

*CD-Recordable Drive* – привод для читання компакт-дисків, CD-R і CD-RW, і запису CD-R і CD-RW. Ці пристрої не можуть стирати дані з перезаписуваних дисків;

*CD-ReWritable Drive* – приводи, що можуть записувати і перезаписувати диски CD-RW.

Приводи CD-ROM усе активніше витісняються з ринку приводами CD-R і CD-RW, що дозволяють у домашніх умовах відповідно записувати компакт-диски і (CD-RW) створювати перезаписувані диски, з якими можна працювати, як із звичайними дискетами, тільки набагато більшої ємності.

Також із компакт-дисками можуть працювати і сучасні приводи DVD, оскільки фізичні розміри дисків збігаються й у приводах DVD використовується той же принцип зчитування інформації.

Пристрій зчитування з компакт-дисків звичайно компонується як внутрішній пристрій, що підключається або до спеціальної карти розширення, або до контролера вінчестера. Розрізняють CDROMи по швидкості: 8-, 12-, 24-, 32-, 40-, 48-, 52- швидкісні.

Основним параметром дисководів CD-ROM є швидкість читання даних. Вона вимірюється в кратних частках. За одиницю виміру прийнята швидкість читання в перших серійних зразках, що складала 150 Кбайт/с. Таким чином, дисковод із подвоєною швидкістю читання забезпечує продуктивність 300 Кбайт/с, з учетвереною швидкістю – 600 Кбайт/с і т. д. У даний час найбільше поширення мають пристрої читання CD-ROM із продуктивністю 32х-48х. Швидкість запису може бути вище, ніж швидкість читання. Залежності швидкості запису і швидкості читання не існує: швидкість запису визначається технологією виготовлення диска, швидкість читання – форматом запису й характеристиками пристрою читання.

**ZIP-накопичувачі.** ZIP-накопичувачі випускаються компанією Iomega, що спеціалізується на створенні зовнішніх пристроїв для збереження даних. Пристрій працює з дисковими носіями, які по розміру незначно перевищують стандартні гнучкі диски і мають місткість 100/250 Мбайт. ZIP-накопичувачі випускаються у внутрішнім і зовнішнім виконанні. У першому випадку їх підключають до контролера жорстких дисків материнської плати, а в другому – до стандартного рівнобіжного порту, що негативно позначається на швидкості обміну даними.

**Накопичувачі HiFD.** Основним недоліком ZIP-накопичувачів є відсутність їхньої сумісності зі стандартними гнучкими дисками 3,5 дюйми. Такою сумісністю володіють пристрої HiFD компанії Sony. Вони дозволяють використовувати як спеціальні носії місткістю 200 Мбайт, так і звичайні гнучкі диски. В даний час поширення цих пристроїв стримується підвищеною ціною.

**Накопичувачі JAZ.** Цей тип накопичувачів, як і ZIP-накопичувачі, випускається компанією Iomega. По своїх характеристиках JAZ-носій наближається до жорстких дисків, але на відміну від них є змінним. У залежності від моделі накопичувача на одному диску можна розмістити 1 або 2 Гбайт даних.

**Магнітооптичні пристрої.** Ці пристрої одержали широке поширення в комп'ютерних системах високого рівня завдяки своїй універсальності. З їхньою допомогою вирішуються задачі резервного копіювання, обміну даними і їхнього накопичення. Однак досить висока вартість пристроїв і носіїв не дозволяє віднести їх до пристроїв масового попиту.

У цьому секторі паралельно розвиваються 5,25- і 3,5-дюймові накопичувачі, носії для яких відрізняються в основному формою-фактором і місткістю. Останнє покоління носіїв формату 5,25" досягає місткості 5,2 Гбайт. Стандартна місткість для носіїв 3,5" – 640 Мбайт.

У форматі 3,5" недавно була розроблена нова технологія GIGAMO, що забезпечує місткість носіїв у 1,3 Гбайт, цілком сумісна зверху вниз із попередніми стандартами. У перспективі очікується поява накопичувачів і дисків форми-фактора 5,25", що підтримують технологію NFR (NearField Recording), що забезпечить місткість дисків до 20 Гбайт, а пізніше і до 40 Гбайт.

**Флеш– пам'ять (флеш-карта)** – різновид твердотільної напівпровідникової енергонезалежної пам'яті, що перезаписується.

Флеш-пам'ять може бути прочитана скільки завгодно раз, але писати в таку пам'ять можна лише обмежене число раз (звичайно близько 10 000). Причина в тім, що для запису в пам'ять необхідно спочатку стерти ділянку пам'яті, а ділянка може витримати лише обмежене число стирань.

Перевагою флеш-пам'яті над жорсткими дисками, CD-ROM-ами, DVD є відсутність частин, що рухаються. Тому флеш-пам'ять більш компактна, дешева (з урахуванням вартості пристроїв читання-запису) і забезпечує більш швидкий доступ. Недоліком, у порівнянні з жорсткими дисками, є відносно малий обсяг: обсяг найбільших флеш-карт складає близько 4 Гб.

Завдяки своїй компактності, дешевині і відсутності потреби в енергії, флеш-пам'ять широко використовується в портативних пристроях, що працюють на батарейках і акумуляторах, – цифрових фотокамерах і відеокамерах, цифрових диктофонах, MP3-плеєрах, КПК, а останнім часом ще й в програмувальних калькуляторах. Крім того, вона використовується для збереження важливої інформації з комп'ютера, а також для збереження убудованого програмного забезпечення в різних периферійних пристроях (маршрутизаторах, комунікаторах, принтерах, сканерах і т.ін.).

**Джерела безперебійного живлення (UPS)** – призначені для захисту ПК від перебоїв в енергозабезпеченні. До складу UPS входить акумуляля-

тор, на забезпечення якого переходить комп'ютер при збої електроживлення. У залежності від пристрою акумулятор забезпечує час від декількох хвилин до декількох годин для завершення коректної роботи комп'ютера.

### Пристрої обміну даними

**Мережна карта** дозволяє приєднати комп'ютер до локальної комп'ютерної мережі. Мережна карта характеризується типом інтерфейсу; найбільш розповсюдженими зараз є інтерфейси "Тонкий Ethernet", "Token Ring", "Товстий Ethernet".

**Модем.** Пристрій, призначений для обміну інформацією між віддаленими комп'ютерами по каналах зв'язку, прийнято називати модемом (МОдулятор + ДЕМОдулятор). При цьому під каналом зв'язку розуміють фізичні лінії (провідні, оптоволоконні, кабельні, радіочастотні), спосіб їхнього використання (комутируємі і виділені) і спосіб передачі даних (цифрові або аналогові сигнали). Розрізняють внутрішні й зовнішні модеми, факс-модеми і голосові модеми, радіомодеми, кабельні модеми та інші.

**Внутрішній модем** компонується як звичайна карта розширення і вставляється в рознімання материнської плати, а **зовнішній модем** – підключається до виведеного назовні рознімання послідовного порту.

**Факс-модем** дозволяє відправляти й приймати факсимільні повідомлення (зв'язуючи безпосередньо з факсами-апаратами), а **голосові модеми** – приймати або передавати голосові повідомлення (наприклад, виконуючи типові функції автовідповідача).

Цифрові дані, що надходять у модем із комп'ютера, перетворюються в ньому шляхом модуляції (по амплітуді, частоті, фазі) відповідно до обраного стандарту (протоколом) і направляються у відповідну лінію зв'язку. Модем-приймач, що розуміє даний протокол, здійснює зворотне перетворення (демодуляцію) і пересилає відновлені цифрові дані у свій комп'ютер. У такий спосіб забезпечується віддалений зв'язок між комп'ютерами й обмін даними між ними.

Від продуктивності модему залежить обсяг даних, переданих в одиницю часу. Від підтримуваних протоколів залежить ефективність взаємодії даного модему із суміжними модемами (імовірність того, що вони вступають у взаємодію один з одним при оптимальних налаштуваннях). Від шинного інтерфейсу в даний час поки залежить тільки простота уста-

новки й настроювання модему (надалі при загальному удосконалюванні каналів зв'язку шинний інтерфейс почне впливати і на продуктивність).

**Спеціалізовані пристрої.** У якості периферійні до комп'ютера можуть приєднуватися цілий ряд спеціальних пристроїв, що забезпечують професійні робочі місця. Прикладами таких пристроїв можуть бути касові термінали, електронні ваги, банкомати, верстаки й апарати з убудованими засобами автоматики і програмувальними пристроями, музичні пристрої. Кількість таких пристроїв постійно росте й змінюється.

**Багатофункціональні пристрої.** В даний час з'явився цілий ряд комбінованих периферійних пристроїв, що сполучають кілька функцій: принтер і ксерокс; принтер, сканер, ксерокс; фото й відеокамера...

Вибір конфігурації ПК залежить від задач, що передбачається вирішувати на комп'ютері. Найбільш вимогливі до потужностей комп'ютера графічні задачі, задачі з відео матеріалами. Відповідно підбирається і перелік периферійних пристроїв. Користувач повинен чітко уявляти собі можливості модифікації комп'ютера й швидкість удосконалювання електронних пристроїв.

### 1.3 Програмне забезпечення ПК

Наявність тільки апаратної частини ще не забезпечує працездатності комп'ютера. Необхідний певний набір програм, що дозволяє «оживити» машину і зробити її корисною користувачеві.

Сукупність програм для комп'ютера утворює **програмне забезпечення (ПО)**.

Більш строго, **програмне забезпечення ЕОМ** – це комплекс програм, документації й інструкцій з їхньої експлуатації, призначених для ефективного управління обчислювальним процесом, зменшення трудомісткості експлуатації комп'ютера, автоматизації процесу підготовки, створення і виконання програм при різних режимах роботи машини, спрощення зв'язку користувача з ЕОМ. Склад програмного забезпечення комп'ютера можна узагальнити за допомогою рисунка 1.6.

По функціональній ознаці розрізняють наступні види ПЗ:

- **системне;**
- **прикладне.**

Під **системним** (базовим) розуміється програмне забезпечення, що забезпечує роботоспроможність комп'ютера (операційні системи, мережне ПЗ, сервісні програми, програми технічного обслуговування),

а також засоби розробки нових програм (транслятори, редактори зв'язків, відладчики й ін.).

Основні функції **операційних систем (ОС)** полягають в управлінні апаратними й інформаційними ресурсами і процесами обчислювальних систем. Під процесом розуміється деяка послідовність дій, запропонована відповідною програмою і даними, що нею використовуються. ОС дозволяє відокремити інші класи задач від безпосередньої взаємодії з апаратурою. (Особливості окремих моделей комп'ютерів можна не враховувати).

**Без операційної системи комп'ютер працювати не може!**

В даний час існує велика кількість ОС, розроблених для ЕОМ різних типів. Наприклад, використовуються такі операційні системи, як DOS 6.22, Windows 95, 98, Windows 2000, Windows NT, Unix, OS/2, Linux та ін.



Рисунок 1.6 – Склад програмного забезпечення комп'ютера



**Мережне ПЗ** призначено для управління загальними ресурсами в розподілених обчислювальних системах: мережними накопичувачами на магнітних дисках, принтерами, сканерами, переданими повідомленнями і т.д. До мережного ПЗ відносять ОС, що підтримують роботу ЕОМ у мережних конфігураціях (так називані мережні ОС), а також окремі мережні програми (пакети), використовувані разом із звичайними, не мережними ОС. Наприклад, велике поширення одержали наступні мережні ОС: NetWare 4.1 (фірма Novell), Windows NT Server 3.5 (фірма Microsoft) і LAN Server 4.0 Advanced (фірма IBM).

Для розширення можливостей операційних систем і надання набору додаткових послуг використовуються **сервісні програми**. Їх можна розділити на наступні групи:

- інтерфейсні системи;
- оболонки операційних систем;
- утиліти.

**Інтерфейсні системи** є надбудовами над операційною системою і модифікують як користувальницький, так і програмний інтерфейси, а також реалізують додаткові можливості по управлінню ресурсами ЕОМ. До них відносяться Windows 3.11, Windows 3.11 for Work Groups.

**Оболонки операційних систем**, на відміну від інтерфейсних систем, модифікують тільки користувальницький інтерфейс, надаючи користувачеві якісно новий інтерфейс у порівнянні з реалізованим операційною системою. Такі системи спрощують виконання часто запитуваних функцій, наприклад, таких, як операції з файлами, як копіювання, перейменування й знищення, а також пропонують користувачеві ряд додаткових послуг. У цілому, програми-оболонки помітно підвищують рівень користувальницького інтерфейсу, найбільш повно задовольняючи потребам користувача. На ПК найбільш часто використовуються Norton Commander (NC), DOS Navigator, FAR.

**Утиліти** надають користувачам засоби обслуговування комп'ютера і його ПЗ.

Вони забезпечують реалізацію наступних дій:

- обслуговування магнітних дисків;
- обслуговування файлів і каталогів;
- надання інформації про ресурси комп'ютера;
- шифрування інформації;
- захист від комп'ютерних вірусів;
- архівація файлів і ін.

Існують окремі утиліти, використовувані для рішення однієї з перерахованих дій, і багатофункціональні комплекти утиліт. В даний час для ПК серед багатофункціональних утиліт одним із найбільш досконалих є комплект утиліт Norton Utilities. Існують його версії для використання в середовищі DOS і Windows.

**Програми (системи) технічного** обслуговування включають різноманітні контролюючі, діагностичні, настроюючі тести й програми для перевірки роботи ПК і окремих його пристроїв.

**Засоби розробки програм** використовуються для розробки нового програмного забезпечення як системного, так і прикладного.

**Прикладним** називається **ПЗ**, що використовується для розв'язання визначеної цільової задачі з проблемної області. Часто такі програми називають додатками.

Спектр проблемних областей у даний час досить широкий і містить у собі принаймні наступні: промислове виробництво, інженерну практику, наукові дослідження, медицину, управління (менеджмент), діловодство, видавничу діяльність, освіту і т.д.

З усієї різноманітності прикладного ПЗ виділяють групу найбільш розповсюджених програм (типові пакети й програми), які можна використовувати в багатьох областях людської діяльності.

**До типового прикладного ПЗ** відносять наступні програми:

- текстові редактори;
- табличні процесори;
- системи ілюстративної й ділової графіки (графічні процесори);
- системи управління базами даних;
- експертні системи;
- програми математичних розрахунків, моделювання й аналізу експериментальних даних;
- програми професійної спрямованості (бухгалтерські, для розрахунку бізнес-планів і ін.).

Пропоновані на ринку ПЗ додатки, у загальному випадку, можуть бути виконані як окремі програми або як інтегровані системи.

Інтегрованими системами звичайно є експертні системи, програми математичних розрахунків, моделювання й аналізу експериментальних даних, а також офісні системи. Прикладом могутньої і широко розповсюдженої інтегрованої системи є офісна система Microsoft Office, що включає наступні прикладні програми – текстовий редактор Word, табличний процесор Excel, система управління базою даних Access, систе-

ма для підготовки презентацій Power Point, система планування колективної роботи користувачів Outlook.

Одним із головних достоїнств пакета Microsoft Office є послідовне використання графічного інтерфейсу користувача фірми Microsoft.

Узагалі, *інтерфейс* – це набір правил, що визначають взаємодію кількох пристроїв. Фізично взаємодія здійснюється впливом робочого органу одного пристрою на органи управління іншого пристрою. У даному випадку маються на увазі два пристрої: користувач (людина) і Microsoft Office (програми). Робочими органами людини при роботі з Microsoft Office є очі й руки. З цих же позицій необхідно описати органи управління другого пристрою – програми. Насамперед, додатки Microsoft Office мають уніфікований інтерфейс. Додатки Microsoft Office надлишкові, тобто частково перекриваються по своїх функціональних можливостях, і тому в різних додатках є подібні функції. Ортогональність (ненадмірність) дозволяє заощаджувати ресурси комп'ютера, надмірність дозволяє заощаджувати ресурси користувача. Розібравшись із тим, що розуміється під уніфікацією інтерфейсу різних додатків, можна відразу ж помітити, що інтерфейси всіх додатків уніфіковані, але в різному ступені. Більше всього уніфікація інтерфейсів позначається в Word, Excel і PowerPoint, тобто в тих додатках, що постійно й безумовно входили й входять до складу Microsoft Office.

Таким чином, познайомивши з роботою одного з компонентів Microsoft Office, користувач уже зможе орієнтуватися в типових командах меню і відповідних їм кнопках панелей інструментів (зберегти, відкрити, друкувати, попередній перегляд, параметри сторінки, копіювати, вирізувати, вставити і т.д.). Причому, якщо користувач забув призначення кнопок, то досить затримати покажчик мишки над інструментальною кнопкою, щоб з'явилася спливаюча підказка з її назвою.

Крім того, у додатках Microsoft Office використовуються однакові прийоми роботи. Наприклад, усі первинні вікна містять пункт меню **Справка**, за допомогою якого можна витягти будь-яку інформацію, що цікавить користувача.

**Контекстні меню.** У Microsoft Office широко використовується інтерфейсний прийом, позитивний вплив якого на ефективність роботи важко переоцінити. Це контекстні меню. Щоб викликати контекстне меню, потрібно підвести покажчик миші до об'єкта, з яким користувач збирається працювати (це може бути виділений текст, елемент у списку, малюнок, чарунка таблиці і взагалі що завгодно), і один раз клацнути правою кнопкою миші. Поруч із покажчиком миші з'явиться меню з

командами, що застосовні до даного об'єкта в даному контексті. Використання контекстних меню скорочує кількість маніпуляцій, необхідних для вибору команди, і зменшує кількість помилок. Якщо в контекстному меню немає команди, що користувач хоче застосувати, то це значить, що команда застосовується в невідповідному контексті: не той об'єкт виділений, не той режим відображення встановлений, не ті значення мають параметри і т.д.

**Множинне виділення й групування.** Дуже багато операцій (наприклад, всі операції форматування) вимагають, щоб попередньо був зазначений об'єкт, до якого застосовується операція. Така вказівка об'єкта як аргумент майбутньої операції називається **виділенням**. Виділено може бути що завгодно: окремий символ, фрагмент тексту, малюнок, чарунка таблиці... Виділений об'єкт відображається особливим образом, щоб було видно, що він виділений. Наприклад, зображення виділеного тексту інвертується, навколо виділеного малюнка з'являються характерні маркери виділення і т.д. Існує безліч різних методів виділення об'єктів, специфічних для різних додатків. Але деякі прийоми виділення є загальними. Так, графічні об'єкти, (наприклад, малюнки) як правило, виділяються одним клацанням лівої кнопки миші, а текст або діапазон чарунок виділяється протаскуванням покажчика миші. На практиці часто зустрічається ситуація, коли ту саму операцію потрібно застосувати відразу до кількох об'єктів. Замість того щоб послідовно кілька разів виділяти черговий об'єкт і виконувати операцію, можна виділити відразу кілька об'єктів і застосувати операцію один раз. Таке виділення називається **множинним виділенням** (для малюнків – синонім групування). Звичайно множинне виділення можна виконати, послідовно виділяючи кілька об'єктів при натиснутій клавіші **[Ctrl]**. Далі для кожного додатка будуть зазначені специфічні методи виділення об'єктів.

При роботі всі додатки Windows можуть використовувати буфер обміну для копіювання або переміщення будь-яких об'єктів. У Office 97 у ньому можна зберігати тільки один об'єкт, а в Office 2000 – 12 об'єктів. Крім того, можна виконувати копіювання й переміщення об'єктів, минаючи буфер обміну – шляхом перетаскування об'єктів з одного вікна (папки) в інше (технологія OLE).

## Питання й завдання для самоперевірки

1. Що таке інформація, дані, знання?
2. Чим відрізняються бази даних і бази знань?
3. Що означають поняття «інформаційні технології», «інформаційна інфраструктура»?
4. Які тенденції у розвитку інформаційних технологій?
5. Поняття й склад інформаційної системи?
6. Що таке принцип відкритої архітектури?
7. Склад базового комплекту ПК?
8. Що входить у поняття периферійних пристроїв?
9. Які принтери використовуються при роботі ПК?
10. Як Ви розумієте запис у прайсі: P-III-1000/ASUS CUSL2/128Mb SDRAM/30,2Gb/1.44/SVGA 32Mb/CD-ROM Samsung 52x/17” SONY E220
11. Що означає термін «програмне забезпечення ПК»?
12. Структура програмного забезпечення?
13. Яке програмне забезпечення називається системним?
14. Призначення й функції операційних систем?
15. Що Вам відомо про прикладне програмне забезпечення?
16. Склад пакета Microsoft Office?

## Література

1. Войтюшенко Н.М., Остапєць А.І. Комп'ютерна техніка і програмування. Частина 1. Основи підготовки користувача ПК: Навчальний посібник по базовій підготовці для студентів денної і заочної форм навчання. – Донецьк, ДонДУ-ЕТ, 2001 – 150 с.
2. Глушаков С.В. Создание и запись CD/ С.В. Глушаков, В.Г.Мачула, Т.С. Хачиров. – Харьков: Фолио, 2005. – 391 с.
3. Інформатика: Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів/ За ред. О.І. Пушкаря. – К.: Видавничий центр “Академія”, 2003. – 704 с.
4. Компьютеры, сети, Интернет. Энциклопедия./Новиков Ю и др. – СПб.: Питер, 2002. – 928 с.
5. Основы современных компьютерных технологий: Учебное пособие/ под ред. проф. Хомоненко А.Д. – СПб.: КОРОНА принт, 1998 – 448с.
6. Резник Ю.А. Графика, звук, видео на ПК. Популярный самоучитель. – СПб.:Наука и Техника, 2003. – 336 с.
7. Стариченко Б.Е. Теоретические основы информатики: Учебное пособие для вузов. – 2-е изд. перераб. И доп. – М.: Горячая линия – Телеком, 2003. – 312 с.
8. Шафран Ю. Информационные технологии: В 2ч. Ч. 2: Офисная технология и информационные системы. – М.: Лаборатория базовых знаний, 2000 – 336с.

## Розділ 2. Операційні системи і сервісні програми

Найбільш розповсюдженими операційними системами для ПК на сьогодні є *Windows 2000*, *Windows XP*, *Linux*. Між *Windows* – продуктом корпорації Microsoft, і *Linux* – результатом багаторічної роботи співтовариства програмістів усього світу, йде досить жорстка боротьба на ринку операційних систем.

З одного боку, *Windows* пропонує найбільш легкий і інтуїтивно зрозумілий інтерфейс для користувачів, з іншого боку, *Linux* дає відкритий код, низьку вартість (вільне розповсюдження) у сполученні з високою стійкістю й функціональністю. Під ці операційні системи створена безліч прикладних програм і кожна з них завоювала величезну кількість прихильників.

Проте, у нашій країні більше поширені продукти фірми Microsoft, тому основна увага буде приділена роботі в операційній системі *Windows*.

### 2.1 Операційна система Windows

#### 2.1.1 Історична довідка і загальна характеристика Windows

У 1981р. на ринку з'являється перший комп'ютер IBM PC на базі процесора Intel 8086/88. У цьому ж році невелика фірма «Microsoft», заснована в 1975р., випускає операційну систему *MS-DOS 1.0*. Дана система виявилася на рідкість вдалою й непередливою стосовно потужностей ПК. Безупинно удосконалюючись, *MS-DOS* стала практично поза конкуренцією в класі IBM PC. У 1991 році з'явилася надбудова (інтерфейсна система) для *MS-DOS – Windows 3.1*, що стала надзвичайно популярною за рахунок використання графічного інтерфейсу й маніпулятора «миша».

Саме графічне представлення інформації додає велику наочність і забезпечує простоту спілкування користувача із системою, а маніпулятор «миша» є вказівним пальцем користувача на екрані.

Однак *Windows 3.1* була ще тільки операційним середовищем, що працювало на базі операційної системи *MS-DOS*. Злиття *MS-DOS* і *Windows* відбулося в 1995 році, коли і з'явилася *Windows-95*. Спадкоємицею *Windows-95* стала операційна система *Windows-98*, потім

*Windows – 2000, Windows XP.* Останні операційні системи створені на базі технології NT, вони поєднують достоїнства операційних систем попередніх поколінь: зручність, простоту в інсталяції й експлуатації, багатофункціональність і мають багато загальних рис.

*Windows-2000 (XP)* дозволяє:

- цілком використовувати повну потужність процесора;
- автоматично розпізнавати й набудовувати конфігурацію комп'ютера і нових периферійних пристроїв;
- забезпечувати повноцінний режим багатозадачності;
- використовувати довгі імена файлів (до 256 символів, включаючи пробіли), а також українську, російську або інші мови в іменах каталогів і файлів;
- використовувати технологію OLE, що дозволяє перетаскувати мишею об'єкти з одного додатка в інший, минаючи буфер обміну;
- надавати користувачеві зручну довідкову систему, більш ефективні драйвери екрана, широкий набір засобів для роботи в глобальній мережі, зокрема, Internet Explorer, оглядач телепередач, стандартну програму по плануванню задач, пакет нових службових програм, що забезпечують безпеку і надійну роботу ПК;
- показувати коротке меню з найбільш часто використовуваними пунктами (програмами, документами).

У *Windows* користувачеві доступний активний робочий стіл, який можна оживити динамічним змістом, наприклад, картою погоди, автоматично обновлюваною через Internet.

При одночасній роботі з великим числом програм у *Windows XP* використовується алгоритм угруповання задач, відповідно до якого однотипні програми поєднуються в логічну візуальну групу, що дозволяє розвантажити панель задач.

### 2.1.2 Робочий стіл Windows

При завантаженні операційної системи на екрані монітора з'являється **Робочий стіл**, на якому розташовані в першу чергу стандартні об'єкти (папки) *Windows*, а потім об'єкти, додані користувачем. (рисунк 2.1).

До стандартних об'єктів відносяться:

- **Мой компьютер** – містить у собі всі об'єкти ПК: диски, периферійні пристрої, панель управління.



Рисунок 2.1 – Робочий стіл *Windows 2000*

- **Корзина** – місце на жорсткому диску, куди переміщуються всі об'єкти, що видаляються. Поки корзина не вичищена, можна відновити вилучені файли або папки.
- **Сетевое окружение**. Якщо ПК працює в локальній мережі, то через цей значок можна побачити всі об'єкти, доступні в мережі.

Унизу **Робочого столу** знаходиться **Панель задач** із кнопками відкритих додатків.

Ліворуч на **Панелі задач** знаходиться кнопка **Пуск**, із якої можна починати будь-яку роботу. Клацання на ній видає на екран меню **Пуск** (головне меню), у якому ви знайдете усе, що може знадобитися для подальшої роботи (рисунок 2.2).

У *Windows* **Робочий стіл** можна представити також у вигляді Web-сторінки (сторінки служби World Wide Web, що містять графічну інформацію в Інтернеті).



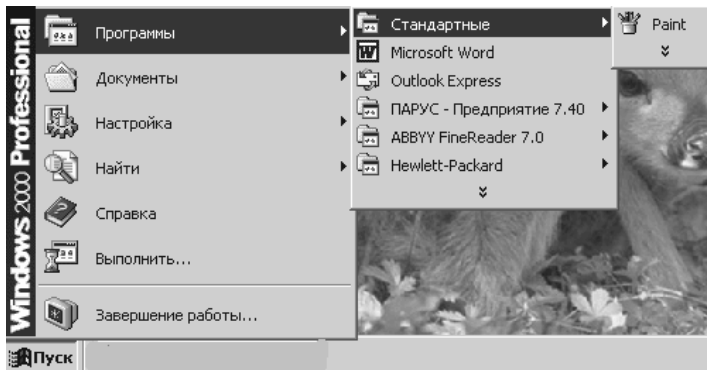


Рисунок 2.2 – Меню кнопки **Пуск** (короткий варіант)

### 2.1.3 Прийоми роботи з маніпулятором «миша»

Оскільки при роботі з Windows широко використовується «мишка», а деякі операції недоступні без неї, розберемо прийоми роботи з нею.

Маніпулятор миша має 2 – 3 кнопки і відображається на екрані у вигляді стрілки. «Мишка» може налаштовуватися (через панель управління) на роботу лівою або правою рукою. При роботі правою рукою основною буде ліва кнопка, а використання правої кнопки буде спеціально пояснюватися. Переміщення миші варто здійснювати по рівній поверхні столу або по спеціальній поверхні, при цьому на екрані монітора рухається покажчик миші (стрілка).

Натискання на кнопку миші називається *клацанням*. Одне натискання – *одинарне клацання*, швидке подвійне натискання – *подвійне клацання*.

*Одинарне клацання* лівою кнопкою використовується для:

- указівки на об'єкт;
- розкриття меню або списку й вибору їхніх пунктів;
- натискання на екранні кнопки.

Наприклад, клацнувши мишею по пункті **Справка** в меню, одержите довідкову інформацію по системі або конкретному діалоговому вікну.

**Подвійне клацанням** лівою кнопкою використовується для:

- розкриття об'єкта;
- запуску програми або документа.

Установивши покажчик на якому-небудь елементі (малюнку, лінії), ви можете перемістити цей елемент в інше місце. Для цього натискають кнопку миші і, не відпускаючи її, переміщують цей елемент у потрібне місце екрана. Потім кнопку відпускають. Така операція називається **перетаскуванням**.

**Перетаскування** застосовується для:

- переміщення об'єктів по екрану;
- виділення прямокутних областей;
- виділення фрагментів тексту й малюнків.

**Клацання правою кнопкою** найчастіше приводить до появи контекстного меню, у якому показуються команди, які можна застосовувати до даного об'єкта.

У деяких моделях мишок між кнопками мається коліща для зручності скролінга (просування за рамки вікна).

## 2.1.4 Вікна Windows

Усе, що є в комп'ютері, у *Windows* представлене у вигляді вікон, звідси і назва *Windows*. Усього можна виділити 4 типи вікон:

**1). Вікна об'єктів і додатків (первинні вікна).** З'являються, коли відкривають об'єкт (значок папки або диска) або запускають програму.

Типова структура такого вікна приведена на рисунку 2.3

1. **Кнопка системного меню** відкриває системне меню, за допомогою якого можна змінити розмір вікна, згортати його в значок, розгорнути на повний екран, переключитися в іншу або закрити поточну програму. Якщо користувач працює з мишею, то всі операції виконуються клацаннями миші.
2. **Рядок заголовка** містить ім'я прикладної програми або документа. Якщо одночасно відкрито кілька вікон, то заголовок відкритого вік-

на виділений іншим кольором. Заголовок вікна залежить від типу вікна, яке відкривається. Це може бути ім'я прикладної програми, ім'я документа, ім'я групи, каталогу файлів. Крім того, при необхідності вікно можна переміщати по екрані за рядок заголовка.

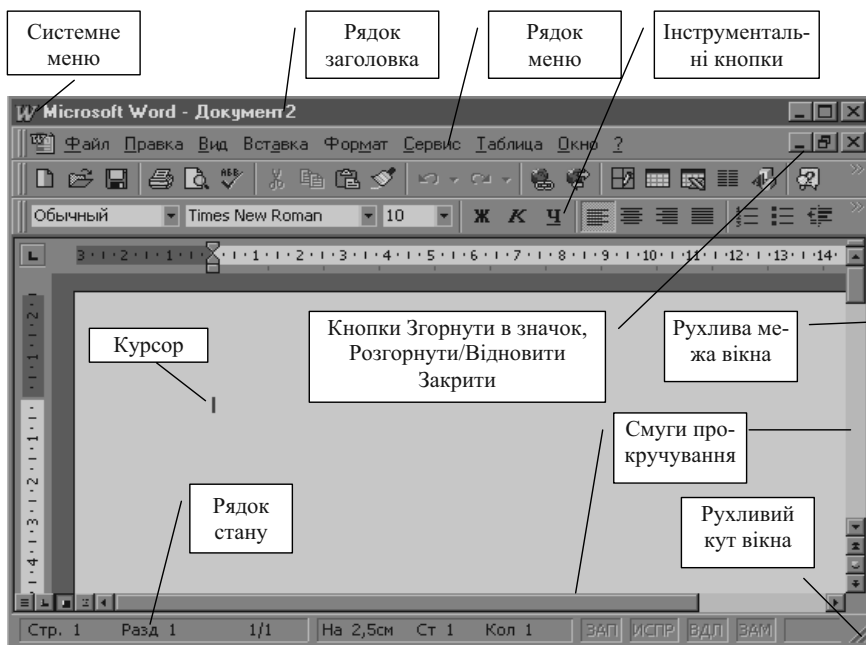


Рисунок 2.3 – Структура первинного вікна

3. У *рядку меню* відображений повний список доступних для роботи команд або дій. Даний елемент властивий тільки первинному вікну. *Інструментальні кнопки* відповідають окремим командам меню. Всі інструментальні кнопки, згруповані по функціональному призначенню у панелі інструментів. Виставити необхідні панелі, видалити зайві або створити свої можна через меню **Вид**, команду **Панелі інструментів**.
4. Кнопка *Згорнути в значок* – звертає вікно в значок і розміщає його на **Панелі задач**. При цьому програма залишається в оперативній пам'яті. Знову розгорнути вікно можна клацанням по значку. Кнопка *Розгорнути/Відновити* – розвертає вікно на весь екран і перетворюється в кнопку **Відновити**. Остання відновлює первісні розміри вікна.

Кнопка **Закрити** вікно – завершує роботу програми і видаляє її з оперативної пам'яті.

5. **Межа вікна** – рухливі межі вікна. При влученні на них покажчик миші приймає вигляд двуспрямованої стрілки ↔. Натиснувши на межі ліву кнопку миші, її можна переміщати в потрібному напрямку, зменшуючи або збільшуючи вікно.
6. **Кут вікна** – рухливий, використовується аналогічно межі вікна і змінює вікно одночасно у двох напрямках. Усі чотири кути первинного вікна рухомі.
7. **Смузи прокручування** – з'являються у випадку, коли інформація не уміщується у вікно. Довжина смуги прокручування відповідає всьому обсягові інформації. Клацання по кнопках із стрілками відповідають переміщенню в зазначеному стрілкою напрямку на один рядок по вертикалі (одну позицію – по горизонталі). Ці ж операції можна виконувати на клавіатурі за допомогою клавіш із стрілками (курсорні клавіші). Листання екранами (вікнами) можна виконувати клацанням між кнопкою зі стрілкою й бігунком (порожня кнопка на смузі прокручування), що відповідає діям клавіш [Page Up] і [Page Down]. Крім того, можна виконувати операцію, що не має аналога на клавіатурі, – перетаскувати бігунки по смузі прокручування в потрібне місце.
8. **Рядок стану** – відображає готовність програми до роботи і поточний стан оброблюваного документа.

**2). Вторинні вікна або вікна документів.** Вікна документів завжди з'являються у вікнах програм, у яких вони обробляються. Подвійне клацання по значку документа змушує Windows визначити по розширенню імені файлу, яка програма створила даний документ, завантажити цю програму, а потім сам документ. У випадку, коли система не може сама визначити програму – джерело, вона задає користувачеві питання про це. У вікнах документів відсутній рядок меню й панелі інструментів.

**3). Діалогові вікна** – призначені для з'ясування побажань і установок користувача. Ці вікна не містять меню, кнопок зміни розмірів вікна, не мають рухливих меж. У діалогових вікнах використовується ряд стандартних елементів: списки, випадаючі списки, прапорці, індикатори вибору, кнопки.

Приклад діалогового вікна приведений на рисунку 2.4

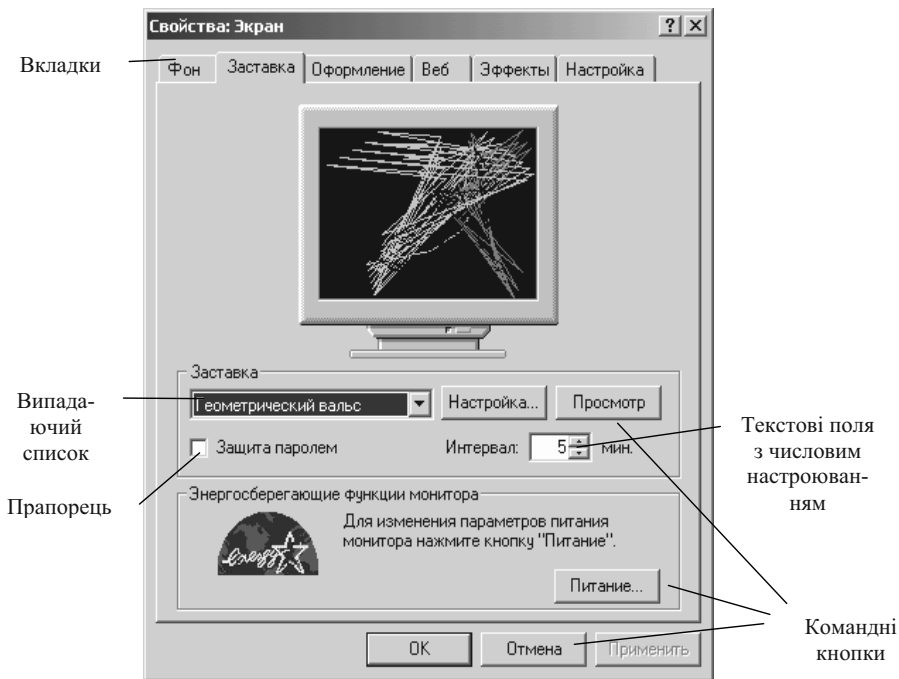





Рисунок 2.4 – Приклад діалогового вікна з вкладками

Діалогове вікно може мати вкладки, на яких розташовуються настроювання визначених властивостей і параметрів. Клацання по назві вкладки викликає листок даної вкладки на передній план. У діалогових вікнах використовуються такі елементи, як: **командні кнопки** (Настройка, Просмотр, ОК, Отмена ...) – для виконання зазначених дій; **текстові поля** – для введення запитуваної інформації; **списки і випадючі списки**, (із кнопкою праворуч від текстового поля) – для вибору даних або файлів із наявного списку; **прапорці** (прямокутні поля) – для установки (після клацання з'являється галочка) або відключення зазначених у підписі параметрів; **селекторні кнопки** вибору (кілька круглих полів) для вибору альтернативного параметра, у обраному після клацання з'явиться крапка; **текстові поля з числовим настроюванням** (із двома кнопками – нагору і вниз – праворуч від тексту) – для настроювання числових параметрів.

**4). Інформаційні вікна** – призначені для інформування користувача і мають тільки одну керуючу кнопку – **ОК**.

Для освоєння основних операцій роботи з вікнами рекомендується виконати конкретний приклад.

### Приклад 2.1.

1. Запустіть *Windows*.  
Після включення комп'ютер завжди автоматично запускає *Windows*.
2. Запустіть програму **Проводник** за допомогою одного з варіантів:
  - клацніть на кнопці **Пуск** → укажіть мишкою м. **Програми** → папка **Стандартные** → клацнути по назві **Проводник**;
  - клацнути правою кнопкою миші по кнопці **Пуск** → клацнути по назві **Проводник**.
3. Розгорніть вікно програми **Проводник** на весь екран.  
Цю операцію можна виконати одним із способів:
  - виконати подвійне клацання лівою кнопкою мишки в зоні заголовка вікна (там, де написаний **Проводник**)
  - клацнути лівою кнопкою мишки на кнопці **Розгорнути** - 
4. Відновіть розміри вікна (із повноекранного у віконний).  
Цю операцію можна виконати одним із способів:
  - подвійним клацанням мишки в зоні заголовка вікна
  - клацанням лівої кнопки мишки на кнопці **Відновити** – 
5. Згорніть вікно в значок.  
Клацніть на кнопці **Мінімізації** –   
Програма згорнеться в значок, що розміститься на **Панелі задач**.
6. Розгорніть значок програми у вікно.  
Відшукайте значок програми **Проводник** на **Панелі задач** і клацніть на ньому. Програма у вигляді вікна знову з'явиться на **Робочому столі**.
7. Змініть розмір вікна програми.  
Укажіть мишкою на межу або кут рамки вікна. Після того як покажчик прийме форму двуспрямованої стрілки, натисніть ліву кнопку мишки і, не відпускаючи її, перемістіть межу. Потім кнопку відпустіть.
8. Перетягніть вікно по **Робочому столу** на нове місце.  
Помістіть покажчик мишки в рядок заголовка вікна, натисніть ліву кнопку і, не відпускаючи її, переміщайте мишку разом із вікном по **Робочому столу**. Потім кнопку відпустіть.
9. Запустіть програми *Мой компьютер, Paint, Word Pad*.  
Запуск програми *Мой компьютер* можна виконати одним із способів:
  - виконати подвійне клацання мишкою на значку програми *Мой компьютер* на **Робочому столі**;

- виконати клацання правою кнопкою мишки на значку програми *Мой компьютер* і у контекстному меню, що з'явиться вибрати команду **Розгорнути**.

Запуск програм *Paint* і *Word Pad* здійснюється послідовно через головне меню: кнопка **Пуск** → к. **Программы** → п. **Стандартные** → клацнути по назві потрібної програми.

10. Зробити активною по черзі кожен з програм: *Проводник*, *Мой компьютер*, *Paint*, *Word Pad*.

Вікно активної програми відрізняється більш темним кольором рядка заголовка.

Роботу можна виконати одним із способів:

- якщо видно хоча б частину необхідного вікна на **Робочому столі**, клацніть на ній;
- якщо видно значок вікна на **Панелі задач**, клацніть на ньому;


11. Натисніть одночасно клавіші [**Alt + Tab**] і побачите список усіх запущених у даний момент програм. Послідовно натискайте [**Tab**], поки не побачите свою програму в рамці, тоді відпустіть [**Alt**].

12. Розташуйте усі відкриті вікна програм на Робочому столі в межах видимості.

Розташувати вікна на **Робочому столі** так, щоб вони всі були видні, можна:

- змінюючи вручну їхні розміри і перетягуючи по столу;
- використовувавши команди: **Каскадом**, **Сверху вниз** або **Слева направо** з контекстного меню, викликаного клацанням правої кнопки мишки на **Панелі задач**.

13. Закрийте усі вікна.

По черзі в кожному вікні клацніть лівою кнопкою миші на кнопці **Закрити** – 

14. Завершіть роботу з *Windows*.

**Ніколи не завершуйте роботу з Windows простим вимиканням комп'ютера!**

Виходьте з *Windows* у такий спосіб:

- клацніть на кнопці **Пуск**;
- клацніть по пункті меню **Завершення роботи**;
- у діалоговому вікні, що з'явилося, установіть перемикач команди **Виключить комп'ютер**;
- клацніть на кнопці **Да**.

**Примітка** – У випадку зависання комп'ютера (відсутня реакція на дії користувача) потрібно спробувати вийти із завислої програми, а у випадку невдачі – перезавантажити машину однією з наступних дій:

1. Спільно натиснути клавіші **[Alt]+[Ctrl]+[Del]** – буде викликане вікно **Безопасность системы** → кнопка **Диспетчер задач** → на вкладці **Приложения** будуть показані всі додатки, що завантажені і їхній стан. Якщо стан «не відповідає», то виділити рядок із таким додатком і клацнути по кнопці **Снять задачу**. Зависла програма буде закрита, при повторному відкритті в ній можна продовжити роботу.

2. Спільно натиснути клавіші **[Alt]+[Ctrl]+[Del]** – буде викликане вікно **Безопасность системы** → кнопка **Завершение работы**.

3. Натиснути кнопку **RESET** на системному блоці.

4. Виключити комп'ютер, а потім увімкнути його повторно.

## Питання й вправи для самостійного виконання

1. Що таке багатозадачна операційна система?
2. Типи вікон використовувані в *Windows*?
3. Що таке Панель задач? Де вона може знаходитися на екрані і що може розташовуватися на ній?
4. Способи перезавантаження ПК при зависанні?
5. Назвіть основні можливості операційної системи *Windows*.
6. Уключіть комп'ютер і знайдіть меню **Пуск**.
7. Запустіть програми *Блокнот* і *Калькулятор*.
8. За допомогою **Панелі задач** розташуйте програми на екрані каскадом і поруч.
9. Згорніть вікно програми *Блокнот*, розгорніть його на весь екран і поверніть програму у вікно за допомогою відповідних кнопок.
10. Змініть розміри вікна, перетаскуючи його за межі.
11. Змініть положення вікна, перетаскуючи його за рядок заголовка.
12. Закрийте усі вікна.
13. Завершіть роботу з *Windows*.

### 2.1.5 Папки й файли. Програми *Мой компьютер* і *Проводник*

Сучасний комп'ютер здатний зберігати в собі величезні масиви інформації. Цю інформацію необхідно упорядковувати з метою швидкого пошуку, читання, переміщення й додавання нової. Будь-яка зв'язана інформація оформляється у файл.

**Файл** – упорядкований набір даних, що має унікальне ім'я і зберігається на одному з пристроїв.



У файлі можуть зберігатися текст, малюнок, таблиця, програма, відеосюжет або музика. І таких файлів згодом стає дуже багато. Один *Windows* розміщає на вінчестері більш 600 файлів. Щоб мати можливість швидко розібратися з потрібними файлами, усі файли групуються в папки.

**Папка** – особливий файл, умістом якого є вкладені папки й файли.

Уся сукупність файлів і папок на комп'ютері утворює файлову систему. **Файлова система** – схема запису інформації, що розміщується на фізичному диску. Файлова система забезпечує можливість доступу до конкретного файлу незалежно від того, у якому місці диска він реально записаний, і дозволяє знайти вільне місце при створенні нового файлу. Системи *MS-DOS* і *Windows 95* використовують файлову систему FAT 16. Операційна система *Windows 98* використовує системи FAT 16 і FAT32, *Windows 2000, XP* – NTFS.

Кожен файл або папка знаходиться на одному з пристроїв і в одній з папок. Структура розміщення папок (каталогів) і файлів на фізичному пристрої називається **файловою структурою**. Вона створюється й обслуговується засобами керування файловою системою.

Сам пристрій – дисковод, вінчестер або CD-ROM – є своєрідною папкою. Кожен файл або папка має ім'я, що формується довільним образом із будь-яких букв, цифр, розділових знаків (до 256 символів), крім символів /, \, «, >, <, :, \*, ?, |. Ім'я файлу, але не папки, містить ще і розширення, що складається не більш ніж із трьох символів, що відокремлюються від імені крапкою. Наприклад: *Babytype.exe*.

Розширення дозволяє виконувати визначені дії над файлами. Так, файли з розширеннями *exe*, *com*, *bat* є здійсненними, тобто подвійне клацання по файлу запускає відповідну програму. Багато програм дають створеним у них документам власні розширення. Наприклад, тестовий редактор *Word* розширення *.doc*, табличний процесор *Excel* – *.xls*, система управління базою даних *Access* – *.mbd* і т.д., що дає можливість системі автоматично визначати, за допомогою якої програми створений документ, і при подвійному клацанні по файлу документа відкривати його в потрібній програмі.

**Правило присвоєння імен: в одній папці не можуть знаходитися файли з однаковими іменами.**

Каталог (папка) найвищого рівня на поточному диску називається **кореневим каталогом**. Усі інші папки вважаються вкладеними в нього у вигляді багаторівневої ієрархічної структури. Місце для кореневого каталогу звичайно, приділяється в строго визначеній області диска,

тому число файлів і папок у кореневому каталозі, як правило, обмежене. Вкладені каталоги таких обмежень не мають.

У різних папках можуть зберігатися файли з однаковими іменами. Це зв'язано з тим, що кожний записаний файл одержує адресу (повне ім'я файлу), що вказує на його місце розташування. Адреса починається з імені пристрою – буква з двокрапкою. Наприклад:

**A:** – гнучкий диск,

**C:, D:, E:** – вінчестер(и),

**F:** – CD-ROM.

Потім послідовно вказуються папки і підпапки, і, нарешті, ім'я файлу.

Наприклад:

*C:\Windows\Мои документи\Звіт за перший квартал* – файл «Звіт за перший квартал» знаходиться в папці *Мои документи*, що у свою чергу поміщена в папку **Windows**, розташовану на диску **C:**

Подвійне клацання мишкою по папці розкриває папку, по значку файлу з документом – при розпізнанні розкриває документ, по значку файлу з програмою – запускає програму на виконання.

Необхідний для роботи файл може бути глибоко захований у папки, що створює значні незручності. Для прискорення процесу завантаження можна використовувати ярлик.

**Ярлик** – це покажчик адреси необхідного файлу. Значок ярлика відрізняється від значка реального файлу маленькою стрілкою в лівому нижньому куті.

Характеристики папки або файлу можна подивитися, виділивши його клацанням миші, а потім вибравши пункт меню **Файл – Свойства**.

У *Windows* для роботи з файлами й папками призначені програми *Мой компьютер* і *Проводник*.

Обидві програми видають на екран файли, папки, дисководи у виді значків, але в різних форматах.

Програма *Мой компьютер* для дисків і папок, що переглядаються, відкриває самостійні вікна (рисунок 2.5).

Вікно програми *Проводник* розділено на дві половини. Ліва половина вікна призначена для перегляду й відкриття папок, розміщених у вигляді дерева. У правій половині виводиться вміст папок, що відкриваються. Поруч із значком папки на дереві може знаходитися "+" або "-". Плюс означає, що в цій папці є ще папки, не показані у виді гілок. Мінус говорить про те, що усі вкладені папки показані у виді гілок (рисунок 2.6).

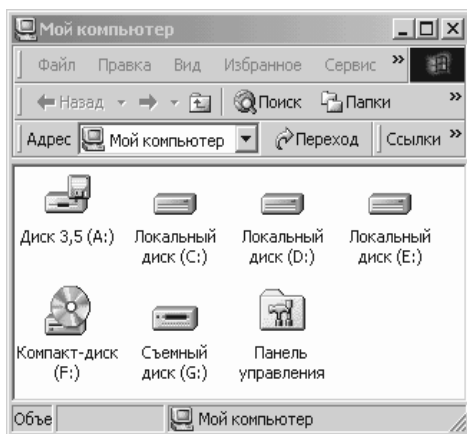


Рисунок 2.5 – Вікно програми *Мой компьютер*

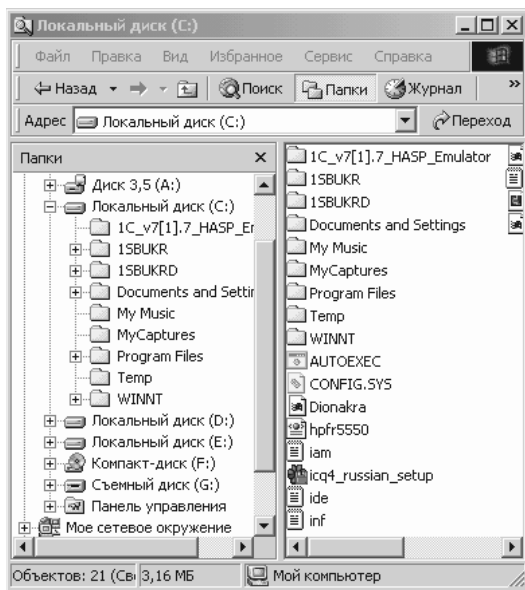


Рисунок 2.6 – Вікно програми *Проводник*

## 2.1.6 Файлові операції в Windows

К файловим операціям відносяться наступні:

- створення нової папки;
- копіювання;
- переміщення;
- перейменування;
- видалення.

Усе різноманіття маніпуляцій з файловою системою комп'ютера зводиться до цих п'яти операцій. Файлові операції можуть бути виконані декількома способами.

### Створення папок, ярликів

Папку можна створювати на **Робочому столі**, будь-якому диску або в іншій папці. Для цього в необхідному місці викличте клацанням правої кнопки миші контекстне меню і виберіть у ньому послідовно команди: **Создать** і **Папка**. Папці, що з'явилася, дайте ім'я і завершіть роботу натисканням [**Enter**].

Ярлик можна створити як в одній з папок, так і на **Робочому столі** одним із способів:

1. Необхідно схопити й перетягнути правою кнопкою мишки значок файлу або папки, для яких створюється ярлик, на місце його створення, вибравши у контекстному меню, що з'явиться, команду **Создать ярлык**. Вхідні значки файлів і папок можуть знаходитися в будь-якому вікні програм *Мой компьютер*, *Проводник*, *функції Поиск* і т.д.
2. Викликати клацанням правої кнопки мишки контекстне меню в місці створення ярлика (**Робочий стіл**, вікно), вибрати в меню команди **Создать, Ярлык**. Використовуючи в діалоговому вікні, що з'явилася, кнопку **Обзор**, відшукати вхідний об'єкт (файл, папку) для створення ярлика.

Ярлики можна копіювати, видаляти, перейменовувати і т.д. Виконання цих операцій не відображається на самих програмах.

### Копіювання й переміщення файлів (папок)

Для переміщення й копіювання файлів з однієї папки в іншу використовується той самий прийом: перетаскування лівою або правою

кнопками миші. Відмінність копіювання від переміщення в тім, що при копіюванні файл залишається на своєму місці, а в іншу папку переміщається його копія. При переміщенні файл на «старому» місці видаляється.

Попередньо ви повинні розкрити потрібні папки на **Робочому столі** так, щоб в одному з вікон ви бачили той файл (папку), яку ви збираєтеся копіювати, а в іншому – папку, у котру буде скопійований або переміщений файл.

Якщо ви перед своїми очима бачите, що і куди ви збираєтеся копіювати, схопіть цікавлячий вас файл або папку і перетягніть його мишкою. Якщо при перетаскуванні буде натиснута клавіша **[Ctrl]** (її треба натиснути раніш, а відпустити пізніше кнопки миші), то відбудеться копіювання, про що свідчить «плюсік» на значку, що переноситься. У протилежному випадку відбудеться переміщення.

Є невелике виключення. Якщо джерело й приймач файлу (папки), що копіюється, знаходяться на різних пристроях, то клавішу **[Ctrl]** можна не натискати. За замовчуванням відбудеться копіювання. Правда, при переміщенні мишею значків програм (файлів із типом **.exe**) відбудеться не копіювання файлів, а створення ярлика.

**Щоб змусити Windows видати на екран меню операцій, що допоможе вам уникнути помилкових дій, завжди перетаскуйте об'єкти правою кнопкою мишки.**

Копіювання і переміщення супроводжується діалоговим вікном із листочками, що перелітають з однієї папки в іншу.

Копію об'єкта можна **відправити** на флоппі-диск, на робочий стіл, по електронній пошті:

контекстне меню об'єкта → к. **Отправить** → указати куди.

## **Переименовання файлів (папок)**

Часто виникає необхідність дати файлові або папці інше ім'я. Для цього досить клацнути по об'єкту правою кнопкою миші, у контекстному меню, що з'явилося, вибрати команду **Переименовать** і ввести нове ім'я.

## **Видалення файлів (папок). Відновлення з Корзини**

Рано або пізно файли й папки приходиться видаляти, тому що місткість дисків вашого комп'ютера обмежена. Для видалення потрібно

лише виділити файл клацанням миші і натиснути клавішу **Delete** на клавіатурі. Файл буде видалений.

Усі файли, що видаляються, не «знищуються», а переміщуються до **Корзини**. Це та ж сама папка, яку, щоправда, не можна ні перемістити, ні перейменувати.

Переміщення в **Корзину**, зрозуміло, робиться про усякий випадок, щоб, раптом спохватившись, ви могли знайти й відновити видалений файл або папку.

Подвійним клацанням **Корзину** можна відкрити й заглянути в неї. Зсередини вона виглядає так само, як звичайна папка. Ви можете скопіювати або перемістити з неї будь-який видалений файл. Але якщо вам треба просто відновити видалений файл, то клацніть по ньому правою кнопкою й у контекстному меню виберіть команду **Восстановить**. Файл буде відновлений у тій же папці, із якої він був видалений.

Варто розуміти, що розміри **Корзини** не безмежні. І якщо вона заповнена, то файли, що найбільше запилилися, будуть дійсно вилучені, а на їхнє місце прийде свіже сміття. Клацнувши правою кнопкою по значку **Корзини** і вибравши з контекстного меню пункт **Свойства**, її можна настроїти. А саме: задати розмір **Корзини** (у відсотках від загальної ємності вінчестера, де вона зберігається), указати, що файли варто знищувати, не поміщаючи в **Корзину**, установити або скасувати запит підтвердження на видалення (переміщення) у **Корзину**.

## Пошук файлів

Якщо ви забули, куди «поклали» той або інший файл, то можна скористатися функцією пошуку файлів і папок. Натисніть кнопку **Пуск** і виберіть пункт **Найти/Файлы и папки...** У лівій частині діалогового вікна, що з'явилося (рисунок 2.7), варто вказати, що і де шукати.

Якщо ви не знаєте, як точно називався файл або потрібно знайти відразу кілька файлів із схожими іменами, то можна використовувати шаблон – ім'я файлу з використанням « \* », що як би заміняє будь-яку кількість букв.

### Наприклад:

- |        |   |
|--------|---|
| Совре* | – усі файли, ім'я яких починається на Совре ; |
| *.txt  | – усі файли з розширенням .txt ;              |
| *.*    | – усі файли.                                  |

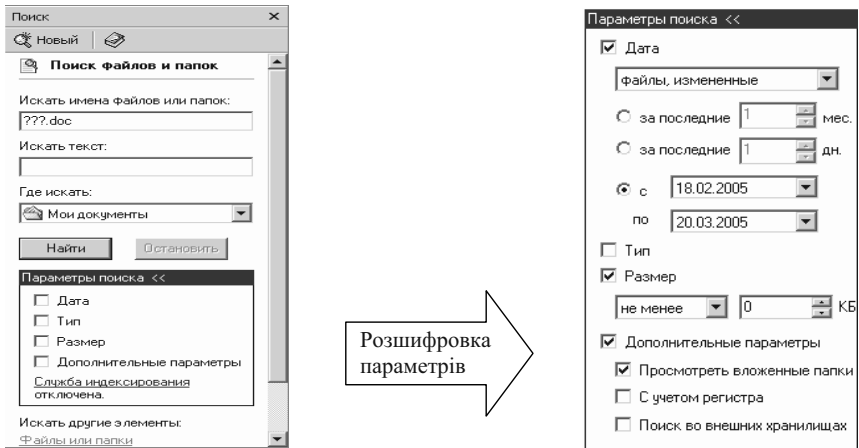


Рисунок 2.7 – Завдання параметрів пошуку

У випадку пошуку імен об'єктів, що мають визначену кількість букв в імені, для завдання шаблону використовується знак ?, він замінює тільки один символ в імені.

Наприклад:

?????.\* – файли з п'ятьма знаками в імені, розширення будь-яке;

Ал????.doc – файли з п'ятьма знаками в імені, із них два перших знаки Ал, а три невідомі, розширення .doc.

### Групові операції

Іноді треба скопіювати або видалити не один, а кілька файлів. Це робиться за допомогою групових операцій.

Для того щоб зробити групову операцію, попередньо виділяється група об'єктів (файлів, папок), а потім здійснюється стандартна операція.

Щоб виділити вибірково кілька об'єктів, просто натисніть **[Ctrl]** і, не відпускаючи її, клацаннями виберіть потрібні об'єкти.

Щоб виділити файли, що йдуть у списку підряд, можна клацнути на першому файлі зі списку, а потім із натиснутою клавішею **[Shift]** – наприкінці списку. Виділиться список файлів.

Виділену групу можна копіювати, перемішувати й видаляти.

## Приклад 2.2.

1. На Робочому столі створити папку з ім'ям **Праця**.
  - клацніть на фоні Робочого столу правою кнопкою мишки і виберіть у меню, що з'явиться, послідовно команди **Создать і Папка**;
  - наберіть слово *Праця* в полі імені папки і натисніть **[Enter]**.
2. Скопіювати в папку **Праця** три будь-яких файли з папки тренажера **Baby**.
  - подвійним клацанням мишки відкрийте папку **Праця**, якщо вона закрита;
  - відкрийте папку **Baby**, послідовно відкриваючи вікна **Мой компьютер**, диск **C:**, **Stud**;
  - у папці **Baby** виділіть три будь-яких файли, клацаючи на них лівою кнопкою мишки і тримаючи при цьому натиснутою клавішу **[Ctrl]**;
  - правою кнопкою мишки схопіть будь-який з виділених файлів і перетаскуйте його в папку **Праця**. При цьому переміщатися будуть одночасно усі виділені файли. У меню, що з'явиться, виберіть команду **Копировать**.
3. Видалити один із файлів папки *Праця* в **Корзину**.

Видалити файл можна одним із способів:

  - виділити його клацанням лівої кнопки мишки, а потім натиснути клавішу **[Delete]** на клавіатурі;
  - перетягнути файл правою або лівою кнопкою мишки в **Корзину**;
  - клацнути на значку файлу правою кнопкою мишки і, у меню, що з'явиться, вибрати команду **Удалить**.
4. Відновити вилучений файл із **Корзини**.

Відновити файл із **Корзини** можна одним із способів:

  - подвійним клацанням мишки на значку відкрийте **Корзину** і перетягніть із неї мишкою файл на попереднє місце;
  - клацніть на значку вилученого файлу в **Корзині** правою кнопкою мишки й у меню, що з'явиться, виберіть команду **Восстановить**. Файл буде відновлений у папці, із якої був вилучений;
  - закрийте вікно **Корзини**.
5. Скопіювати в папку **Праця** з папки **Мої документи** усі файли з розширенням **.doc** і довжиною імені рівної трьом символам.

Для виконання завдання доцільно скористатися функцією **Найти** з меню кнопки **Пуск**.



- клацніть мишкою на кнопці **Пуск** і виберіть послідовно команди **Найти** і **Файлы и папки**;
- у діалоговому вікні, що відкрилося, виконати:
  - а) у полі **Файлы** наберіть шаблон для пошуку необхідних файлів: **???.doc** (рисунк 2.8)

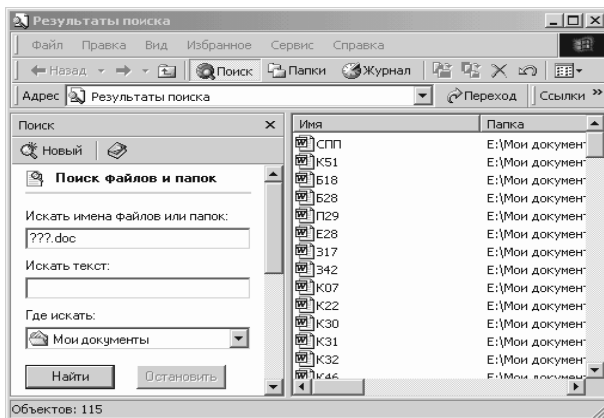
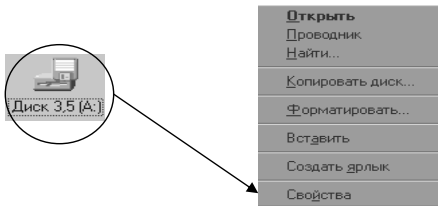


Рисунок 2.8 – Діалогове вікно пошуку файлів

- б) у полі **Где искать** наберіть ім'я папки: **Мои документы** або скористайтеся для заповнення цього поля списком, що розкривається;
    - клацніть на кнопці **Найти**. Імена знайдених файлів будуть виведені у вікні пошуку;
    - виберіть файли для копіювання одним із способів:
      - в) у вікні пошуку клацніть на значку першого файлу, натисніть клавішу **[Shift]** і, не відпускаючи її, клацніть на останньому файлі списку. Усі файли будуть виділені;
      - г) у пункті меню **Правка** діалогового вікна функції **Поиск** виберіть команду **Выделить все**.
    - схопіть правою кнопкою мишки будь-який з виділених файлів і перетягніть його в папку **Праця**. Переміщатися будуть усі виділені файли;
    - у меню, що з'явиться, виберіть команду **Копировать**.
6. Визначити наявність вільного місця на диску **A:**
- уставте дискету в дисковод;

- відкрийте вікно програми *Мой компьютер* подвійним клацанням на її значку;
- клацніть правою кнопкою на значку диска **A:** ;
- у меню, що з'явиться, виберіть команду **Свойства**;



- дані про заповнювання диска будуть представлені в діалоговому вікні, що відкрилося (рисунок 2.9);

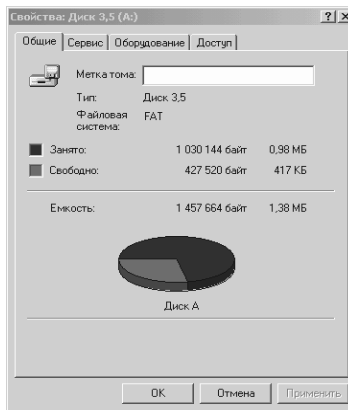
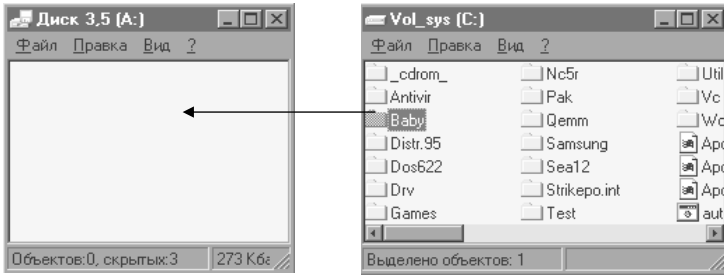


Рисунок 2.9 – Діалогове вікно властивостей об'єктів

## 7. Скопіювати на дискету папку **Baby**.

- уставте дискету в дисковод, якщо її там немає;
- відкрийте вікно програми *Мой компьютер*;
- у вікні *Мой компьютер* подвійним клацанням мишки на значку диска **A:** відкрийте його вікно;
- відкрийте вікно диска **C:** і відшукайте в ньому папку **Baby**;
- розташуйте вікна дисків **A:** і **C:** поруч на **Робочому столі**, закривши інші вікна;
- правою кнопкою мишки перетягніть папку **Baby** у вікно диска **A:** ;



- у меню, що з'явиться, виберіть команду **Копировать**. Папка **Baby** з усіма файлами, що утримувалися в ній, буде скопійована на дискету;
  - закрийте усі вікна на **Робочому столі**.
8. На **Робочому столі** створити ярлик програми *Paint*.  
Створювати ярлики можна різними способами. Розглянемо деякі з них.
- а). Використання контекстно – залежного меню **Робочого столу**:
- клацніть на **Робочому столі** правою кнопкою мишки;
  - у меню, що з'явиться, виберіть послідовно команди **Создать, Ярлик** (рисунок 2.10);

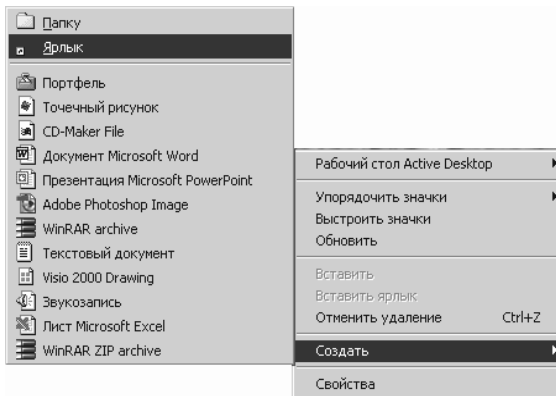
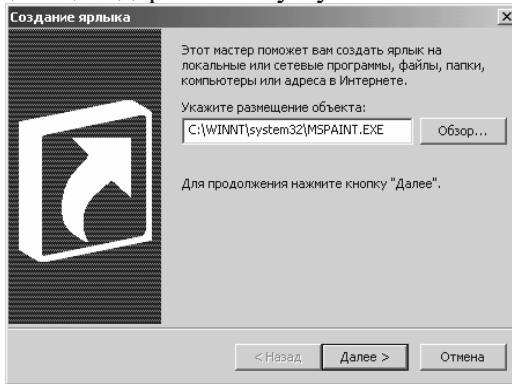


Рисунок 2.10 – Вибір команд для створення ярлика

- у діалоговому вікні “Создание ярлика”, що з'явиться, клацніть на **кнопці Обзор**;

- на диску C:\ відшукайте папку Windows і подвійним клацанням мишки відкрийте її. При необхідності, використовуйте полосу прокручування, щоб побачити весь уміст папки C:\. Для Windows 2000 необхідно ще відкрити папку System32.

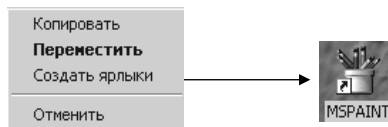


- двічі клацніть на значку файлу *MSPAIN.T* ;
- у діалоговому вікні клацніть на кнопці **Далее**;
- наберіть назву ярлика у відведеному для нього полі;
- клацніть на кнопці **Готово**.

На Робочому столі з'явиться ярлик програми *Paint*.

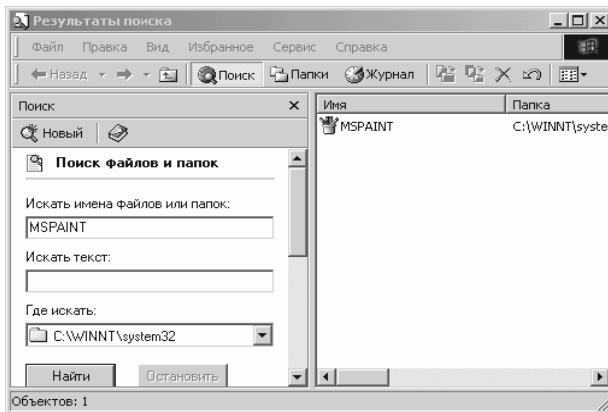
б). Використання програми *Мой компьютер*:

- відкрийте вікно програми *Мой компьютер* подвійним клацанням мишки на його значку;
- двічі клацніть на значку диска **C:**. З'явиться вікно, що відображає всі папки й файли диска **C:** ;
- подвійним клацанням мишки відкрийте папку **Windows**. З'явиться вікно з її вмістом;
- відшукайте в папці **Windows** папку **System32**, а в ній файл *MSPAIN.T*;
- клацніть правою кнопкою мишки на значку файлу *MSPAIN.T* і, не відпускаючи її, витягніть файл на Робочий стіл;
- відпустіть кнопку мишки, а потім лівою кнопкою, у меню, що з'явиться, виберіть команду **Создать ярлык**.



Ярлик з'явиться на **Робочому столі**.

- закрийте усі відкриті вікна на **Робочому столі**.
- в). Використання функції **Найти**:
  - клацніть на кнопці **Пуск** ;
  - у меню, що з'явиться, виберіть послідовно команди **Найти, Файлы и папки**;
  - а потім у полі імені наберіть ім'я шуканого файлу, тобто *MSPAINТ* , а в полі **Папка** – розкрийте список, що випадає, і виберіть останній пункт **Обзор**, послідовно розкриваючи папки, знайдіть потрібну і клацніть **ОК**;
  - клацніть на кнопці **Знайти**;
  - знайдений файл буде виведений у вікні пошуку;



- витягніть правою кнопкою миші значок файлу на **Робочий стіл**;
  - у меню, що з'явиться, виберіть команду **Создать ярлык**;
  - закрийте вікно пошуку.
9. Перемістити в папку **Праця** ярлик програми *MSPAINТ*.
- подвійним клацанням мишки відкрийте папку **Праця**;
  - правою кнопкою мишки перетягніть ярлик **MSPAINТ** у папку **Праця**. Відпустіть кнопку і, у меню, що з'явиться, виберіть команду **Переместить**.
- Для переміщення можна використовувати і ліву кнопку мишки, але при цьому не буде виводитися меню операцій.
10. **Переименовати папку Праця в Работа**.
- клацніть правою кнопкою на значку папки **Праця**;

- у контекстному меню, що з'явиться, виберіть команду **Переименовать**;
  - у рядку, що містить назву папки, з'явиться миготливий курсор вставки. Натисніть клавішу [**Пробіл**] для видалення старої назви, а потім уведіть у поле, що звільнилося, нове ім'я папки (**Робота**) і натисніть [**Enter**].
11. Переглянути й записати властивості папки **Робота**.
    - клацніть правою кнопкою мишки на значку папки **Робота**;
    - у меню, що з'явиться, виберіть команду **Свойства**;
    - в діалоговому вікні, що з'явиться, перегляньте властивості папки: тип, розмір, дату створення і т.д. ;
    - закрийте вікно властивостей.
  12. Видалити папку **Робота** в **Корзину**.
    - клацніть на значку папки будь-якою клавішею мишки і, тримаючи її натиснутою, перетягніть папку в **Корзину**.

## Питання й вправи для самостійного виконання

1. Дайте тлумачення поняттям **папка, файл, ярлик, повне ім'я файлу**.
2. Назвіть, які операції з інформацією можна виконувати безпосередньо в операційній системі Windows?
3. Що таке шаблони імен папок і файлів і як вони будуються?
4. Завантажте програму *Мой компьютер*.
5. Уставте дискету в дисковод **A:** і перегляньте її вміст.
6. Створіть папку **Приклад** на диску **A:**
7. Перемістіть створену папку на диск **C:**
8. Скопіюйте в папку **Приклад** три будь-яких файли з будь-якої іншої папки.
9. Перегляньте властивості папки **Приклад**.
10. Закрийте програму *Мой компьютер* і всі, стосовні до неї вікна.
11. Відкрийте програму *Проводник* і переглянете вміст папки **Windows**, розташованої на диску **C:**
12. Використовуючи команду **Упорядочить значки** в меню **Вид**, відсортуйте файли в алфавітному порядку, потім відшукайте усі файли папки **Windows**, що починаються з букви **C**.
13. Скопіюйте п'ять поруч розташованих знайдених файлів у папку **Приклад**.
14. Видаліть папку **Приклад**.
15. Закрийте програму *Проводник*.
16. Створіть на **Робочому столі** ярлики до всіх дисків із вікна програми *Мой компьютер*.
17. Використовуючи функцію **Поиск**, відшукайте усі файли, що містять слова «інформація» і до двох із них створіть ярлики на **Робочому столі**.

18. Перегляньте один із файлів, використовуючи створений для нього ярлик. Закрийте програму, яка відкрилася для переглядання файлу.
19. Видаліть всі створені ярлики.

## 2.1.7 Вбудовані додатки Windows

Крім зазначених функцій по організації роботи з папками і файлами **Windows** містить папку **Стандартные** з додатками різного характеру – по обслуговуванню комп'ютера, ігри й розваги, засоби Internet, калькулятор, блокнот, графічний редактор. Як приклад коротко розглянемо роботу з графічним редактором *Paint* і калькулятором.

### Калькулятор

Windows надає послугам користувача зручний і корисний додаток – калькулятор, що забезпечує як прості розрахунки, так і інженерні. Завантажити калькулятор можна:

кнопка **Пуск** → м. **Программы** → п. **Стандартные** → п.п. **Калькулятор**.

Калькулятор може бути представлений у звичайному виді (рисунок 2.11) або для інженерних розрахунків (рисунок 2.12).

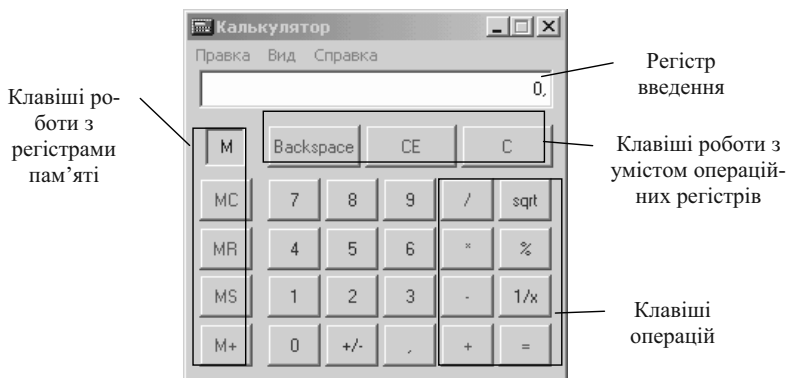


Рисунок 2.11 – Звичайний калькулятор

Виконання операцій за допомогою програми Калькулятор нічим не відрізняється від виконання аналогічних дій на звичайному калькуляторі. Крім того, у даної програми є відмінна підказка: клацання пра-

вою кнопкою миші по будь-якій кнопці або полю калькулятора викликає питання «Що це таке?». Якщо потім по ньому клацнути лівою кнопкою, то з'явиться конкретне роз'яснення про це.



Рисунок 2.12 – Калькулятор для інженерних розрахунків

### Робота з пам'яттю

- Занесення відображуваного числа в пам'ять – кнопка **MS**. Кожне нове число, занесене в пам'ять, замінює попереднє.
- Виклик числа з пам'яті – кнопка **MR**.
- Очищення пам'яті – кнопка **MC**.
- Щоб скласти відображуване число з числом, що зберігається в пам'яті, натисніть кнопку **M+**.
- Виклик результату з пам'яті – кнопка **MR**.

Виконання інженерних обчислень починається з наступних операцій:

м. **Вид** → к. **Інженерний** → вибрати систему числення → вибрати одиниці відображення → виконувати розрахунки.

Можливе виконання операцій над числами в наступних системах числення: десятковій, шістнадцятирічній, восьмирічній і двійковій. Для чисел у шістнадцятирічній, восьмирічній і двійковій системах числення доступні чотири розміри відображення: 8 байт (64-бітне представлення), 4 байти (32-бітне представлення), 2 байти (16-бітне представлення)



і 1 байт (8-бітне представлення). Для чисел у десятковій системі числення доступні три розміри відображення: градуси, радіани і гради.

При виборі іншої системи числення відбувається перетворення числа в цю систему.

Примітка -

- При перетворенні нецілого десяткового числа в іншу систему числення його дробова частина відкидається.
- Шістнадцятирічні, восьмирічні і двійкові числа, перетворені в десяткові, відображаються як цілі.

При переключенні між звичайним і інженерним режимом реєстр уведення калькулятора очищається. Щоб перевести число з одного режиму в інший, варто зберегти його в пам'яті:

1. Натисніть кнопку **MS**, щоб занести число в пам'ять.
2. У меню **Вид** виберіть потрібний режим.
3. Натисніть кнопку **MR**, щоб викликати число з пам'яті.

Примітка -

Числа в шістнадцятирічному, восьмирічному і двійковому форматах при переході від інженерного до звичайного режиму перетворюються в десятковий формат.

У режимі калькулятора **Інженерний** можна виконати **статистичні** обчислення:

1. Уведіть перше число і натисніть кнопку **Sta**, щоб відкрити вікно **Статистика**.
2. Натисніть кнопку **RET**, щоб повернутися у вікно калькулятора, а потім натисніть кнопку **Dat**, щоб зберегти це значення.
3. Уведіть інші числа, натискаючи кнопку **Dat** після введення кожного з них.
4. Натисніть кнопку **Ave**, **Sum** або **s**.
  - **Ave** обчислює середнє значення для чисел, збережених у вікні **Статистика**, кнопка **Sum** обчислює суму цих чисел, а кнопка **s** – незміщене стандартне відхилення.
  - Після того як усі дані введені, їхній список можна переглянути, натиснувши кнопку **Sta**.
  - У нижній частині вікна **Статистика** показана кількість збережених значень. Можна видалити будь-яке значення зі списку, натиснувши кнопку **CD**, або видалити всі значення, натиснувши кнопку **CAD**. Натиснувши кнопку **LOAD**, можна

замінити число, відображуване на калькуляторі, числом, обраним у вікні **Статистика**.

## Графічний редактор Paint

### Завантаження редактора:

кнопка **Пуск** → м.**Программы** → п.**Стандартные** → п.п.**Paint**.

Буде завантажене вікно графічного редактора (рисунок 2.13).

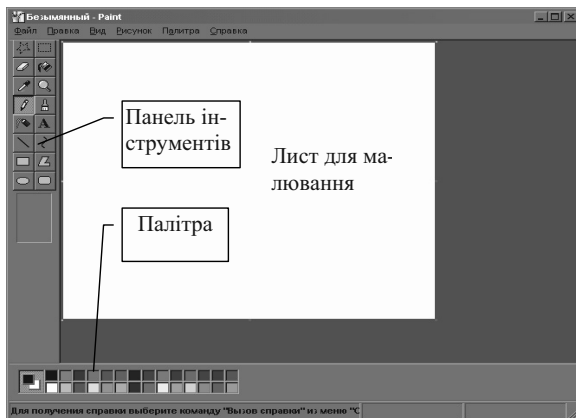


Рисунок 2.13 – Вікно графічного редактора *Paint*

У вікні редактора, крім стандартних елементів первинного вікна *Windows* розташований лист для малювання, панелі інструментів і палітра. Якщо останніх нема, то вони виставляються через меню **Вид**. (У всіх вікнах *Windows* і Microsoft Office у пункті меню **Вид** розташовані команди, що відповідають за зовнішній вигляд вікна.)

Розмір листа для малювання можна установити через меню **Рисунки** → команда **Атрибути** (рисунок 2.14).

Вибір кольору фону здійснюється клацанням правої кнопки миші по палітрі, а кольору малювання – клацанням лівої кнопки миші. Потім клацанням миші вибирається інструмент для малювання – геометричні фігури або олівець. Показчик миші встановлюється в потрібному місці листа для малюнка, зажимається ліва кнопка й миша переміщується в потрібному напрямку (наприклад, вправо й униз). Одержите обрану фігуру або лінію.

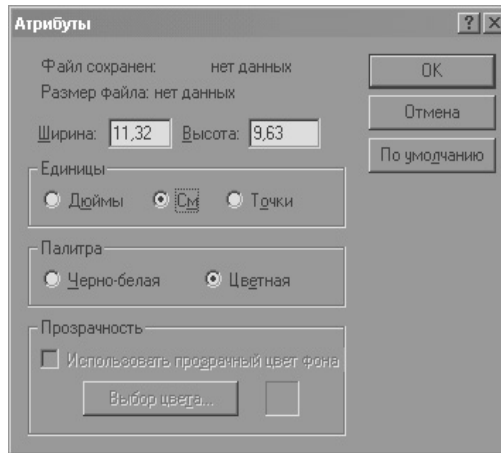


Рисунок 2.14 – Діалогове вікно по установці атрибутів рисунка

При невдалому нанесенні ліній або фігур можна користуватися ластиком (кнопки панелей інструментів при затримці на них покажчика мишки підписуються).

Малюнки можна розфарбовувати за допомогою інструмента **Заливка**, підписувати – за допомогою інструмента **Надпись**. Більш детальний опис можливостей даного графічного редактора мається в довідці.

Намалюйте будь-який малюнок (наприклад, як на рисунку 2.15) із використанням геометричних фігур, олівця, розфарбуйте, підпишіть його і збережіть з ім'ям *Мій малюнок* у папці *Мої документи*. При наявності колірного друку малюнок буде виглядати більш веселим.

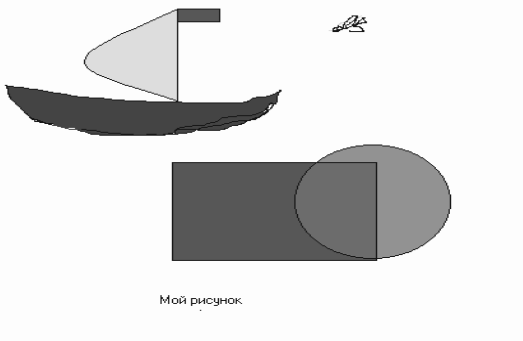


Рисунок 2.15 – Мій малюнок

Для збереження треба виконати:

м. **Файл** → к. **Сохранить как** → в діалоговому вікні (файлері) дати ім'я файлу і вказати, в якій папці його зберегти (рисунок 2.16) → кнопка **Сохранить**.

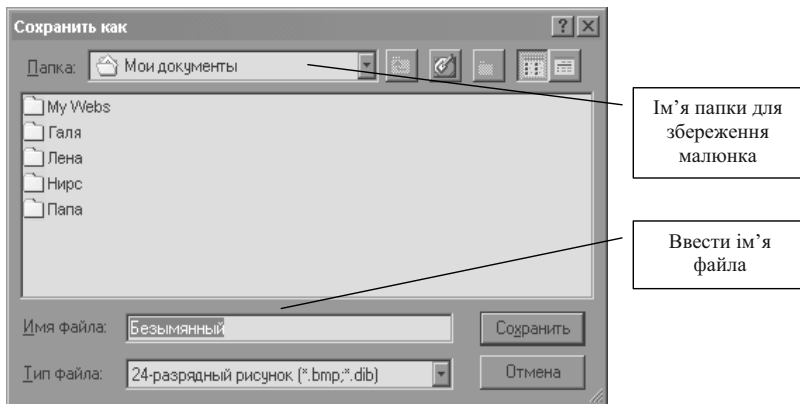


Рисунок 2.16 – Діалогове вікно збереження документа

Надалі він ще буде використовуватися в нашій роботі. Природно, що при завантаженні малюнка у графічний редактор (м. **Файл** → к. **Открыть** → у діалоговому вікні вказати ім'я потрібної папки і файлу в ній → кнопка **Открыть**) можна його змінювати й зберігати вже у відредагованому виді. При цьому можна уточнювати дрібні деталі малюнка шляхом виведення сітки:

м. **Вид** → к. **Масштаб** → п. **Крупный**, а потім ще раз

м. **Вид** → к. **Масштаб** → п. **Показать сетку**.

Повернення до звичайного виду йде через те ж меню й команду, п. **Обычный**.

## Питання й завдання для самостійного виконання

1. Назвіть, які убудовані в *Windows* додатки Ви знаєте?
2. Призначення різних видів калькулятора?
3. Виконайте за допомогою калькулятора наступні обчислення:  
$$Y = (X^2 + X^3 + 1/X) - (\ln X + e^X)$$
, де  $X=2$ ; 7; 3,5
4. Які види робіт можна виконувати в додатку *Paint*?
5. Як виставити на екран панель інструментів, палітру, сітку для промальовування дрібних деталей малюнка?

6. Як змінити розміри листа малюнка?
7. Створіть малюнок довільного змісту, у якому необхідно обов'язково використувати олівець або пензлик, геометричні фігури, заливання й написи. Збережіть малюнок у папці **Мої малюнки**.
8. Скопіюйте фрагмент малюнка і вставте його в той же малюнок, збережіть новий варіант малюнка з іншим ім'ям.
9. Завантажите в редактор будь-яку готову картинку ( можна з папки **ClipArt** ) і внесіть у неї довільні зміни. Збережіть новий варіант картинки в папці **Мої малюнки**.

## 2.2 Сервісні програми

### 2.2.1 Віруси й боротьба з ними

В даний час комп'ютерні віруси одержали дуже широке поширення й антивірусна боротьба доставляє рядовому користувачеві великий «головний біль». Хоча вірусні атаки трапляються не дуже часто, загальне число вірусів занадто велике, а збиток від “хуліганських” дій вірусу в системі може виявитися значним. Існують віруси, що можуть привести до втрати програм, знищити дані, стерти необхідну для роботи комп'ютера інформацію, записану в системних областях пам'яті, привести до серйозних збоїв у роботі комп'ютера. У результаті цих дій Ви можете назавжди втратити дані, необхідні для роботи і понести істотний моральний і матеріальний збиток. “Епідемія” комп'ютерного вірусу у фірмі (неважливо – великій або маленькій) може цілком дестабілізувати її роботу. При цьому може відбутися збій у роботі, як окремих комп'ютерів, так і комп'ютерної мережі в цілому, що спричинить за собою втрату інформації, необхідної для нормальної роботи й утрату часу, що буде витрачено на відновлення даних і приведення комп'ютерів і/або мережі в робочий стан.

Тому важливо розуміти способи поширення й характер прояву вірусів, а головне, навчитися грамотно застосовувати антивірусні програми для ефективної боротьби з вірусами.

### Характеристика вірусів

*Вірус* являє собою програму, що самовідтворюється, що здатна упроваджувати свої копії (не обов'язково цілком співпадаючі з оригіналом) у файли, системні області, обчислювальні мережі і т.д. і приво-

дити до порушення нормального функціонування комп'ютера. Копії вірусної програми також зберігають здатність подальшого поширення. Віруси прийнято класифікувати по наступних ознаках:

- середовищу існування;
- способу зараження середовища існування;
- способу активації;
- деструктивним можливостям;
- особливостям алгоритму.

По середовищу існування віруси розділяють на файлові, завантажувальні мережні. Файлові віруси впроваджуються у файли, найчастіше виконувані, або файли документів текстових редакторів і робочих книг табличних процесорів. Завантажувальні віруси впроваджуються в завантажувальний сектор диска або в сектор системного завантажника жорсткого диска. Мережні віруси поширюються по комп'ютерній мережі. Існують також файлово-завантажувальні віруси, що заражають файли і завантажувальні сектори.

Спосіб зараження середовища існування залежить від самого середовища. Зокрема, тіло файлового вірусу може при зараженні розміщатися наприкінці, початку, середині або хвостовій (вільній) частині останнього кластера файлу. Найбільше просто реалізується упровадження вірусу в кінець файлу типу `com`. Найбільш складна імплантація вірусу в середину файлу, оскільки для цього повинна бути відома структура файлу, що заражається, щоб можна було впровадитися, приміром, в область стека. При впровадженні завантажувального вірусу (через малі розміри середовища існування) використовується розміщення голови тіла замість завантажувального сектора диска або сектора системного завантажника, а хвіст вірусу і наступний за ним завантажувальний сектор розміщаються в інших кластерах або секторах.

По способу активації віруси підрозділяють на резидентні й нерезидентні. Резидентний вірус при зараженні залишає в оперативній пам'яті резидентну частину, що потім перехоплює звертання операційної системи до об'єктів зараження – файлам, завантажувальним секторам і т. п., і впроваджується в них. Резидентні віруси зберігають свою активність аж до вимикання або перезавантаження комп'ютера. Нерезидентні віруси є активними обмежений час і активізуються у визначені моменти, наприклад, при запуску заражених виконуваних програм або при обробці документів текстовим редактором. Деякі нерезидентні віруси залишають в оперативній пам'яті невеликі резидентні програми.

По деструктивних можливостях віруси розділяють на нешкідливі, безпечні, небезпечні і дуже небезпечні. Нешкідливі віруси виявляються тільки в тім, що зменшують обсяг пам'яті на диску в результаті свого поширення. Безпечні віруси, крім відзначеного прояву, породжують графічні, звукові й інші ефекти. Небезпечні віруси можуть привести до порушень нормальної роботи комп'ютера, наприклад, до зависання або до неправильного друку документа. Дуже небезпечні віруси можуть привести до знищення програм і даних, стиранню інформації у системних областях пам'яті і навіть приводити до виходу з ладу частин жорсткого диска, що рухаються, при введенні в резонанс.

По особливостях алгоритмів розрізняють наступні віруси:

**компаньйони-віруси** (companion) – це віруси, що не змінюють файли. Алгоритм роботи цих вірусів полягає в тому, що вони створюють для EXE-файлів файли-супутники, що мають те ж саме ім'я, але з розширенням .COM, наприклад, для файлу XCOPY.EXE створюється файл XCOPY.COM. Вірус записується в COM-файл і ніяк не змінює EXE-файл. При запуску такого файлу DOS першим знайде й виконає COM-файл, тобто вірус, що потім запустить і EXE-файл;

**віруси -“хробаки”** (worm) – віруси, що поширюються в комп'ютерній мережі і, так само як і компаньйони-віруси, не змінюють файли або сектори на дисках. Вони проникають у пам'ять комп'ютера з комп'ютерної мережі, обчислюють мережні адреси інших комп'ютерів і розсилають по цих адресах свої копії. Такі віруси іноді створюють робочі файли на дисках системи, але можуть узагалі не звертатися до ресурсів комп'ютера (за винятком оперативної пам'яті);

**“паразитичні”** – усі віруси, що при поширенні своїх копій обов'язково змінюють уміст дискових секторів або файлів. У цю групу відносяться усі віруси, що не є “хробаками” або “компаньйони”;

**“студентські”** – у край примітивні віруси, часто нерезидентні й мають велике число помилок;

**“стелс”-віруси** (віруси – невидимки, stealth), що представляють собою досить досконалі програми, які перехоплюють звертання операційної системи до уражених файлів або секторів дисків і “підставляють” замість себе незаражені ділянки інформації. Крім цього, такі віруси при звертанні до файлів використовують досить оригінальні алгоритми, що дозволяють “обманювати” резидентні антивірусні монітори;

**“поліморфік”-віруси** (що самостійно шифруються або віруси-примари, polymorphic) – віруси, які досить важко виявляються, тому, що не мають сигнатур, тобто не містять жодної постійної ділянки коду. У більшості випадків два зразки того самого поліморфік-віруса не будуть мати жодного збігу. Це досягається шифруванням основного тіла вірусу й модифікаціями програми-розшифровувача;

**“макро-віруси”** – віруси цього сімейства використовують можливості макро-мов, убудованих у системи обробки даних (текстові редактори, електронні таблиці і т.д.). В даний час найбільш поширені макро-віруси, що заражають текстові документи редактора Microsoft Word.

Основні симптоми вірусної поразки наступні:

- Уповільнення роботи деяких програм.
- Збільшення розмірів файлів (особливо виконуваних).
- Поява не існуючих раніше “дивних” файлів.
- Зменшення обсягу доступної оперативної пам'яті (у порівнянні зі звичайним режимом роботи).
- Раптово виникаючі різноманітні відео і звукові ефекти.

При всіх перерахованих вище симптомах, а також при інших “дивних” проявах у роботі системи (хитлива робота, часті “самостійні” перезавантаження та інше) настійно рекомендується негайно зробити перевірку Вашої системи на наявність вірусів за допомогою антивірусної програми.

## **Класифікація антивірусних програм**

**Антивірусними** називаються програми, призначені для захисту даних від руйнування, для виявлення й видалення комп'ютерних вірусів. Розрізняють наступні різновиди антивірусних програм:

- фільтри, або сторожа;
- детектори;
- доктори, або фаги;
- ревізори;
- імунізатори, або вакцини.

Фільтр являє собою резидентну програму, що контролює небезпечні дії, характерні для вірусних програм, і запитує підтвердження на їхнє виконання. До таких дій відносяться наступні:



- зміна файлів виконуваних програм;
- розміщення резидентної програми;
- прямий запис на диск по абсолютній адресі;
- запис у завантажувальні сектори диска;
- форматування диска.

Достоїнством програм-фільтрів є їхнє постійне відстеження небезпечних дій, що підвищує ймовірність виявлення вірусів на ранній стадії їхнього розвитку. З іншого боку, це ж є і недоліком, тому що приводить до відволікання користувача від основної роботи для підтвердження запитів на виконання підозрілих операцій.

Детектори забезпечують пошук і виявлення вірусів в оперативній пам'яті і на зовнішніх носіях. Розрізняють детектори універсальні й спеціалізовані. Універсальні детектори у своїй роботі використовують перевірку незмінності файлів шляхом підрахунку й порівняння з еталоном контрольної суми. Недолік універсальних детекторів зв'язаний з неможливістю визначення причин перекручування файлів. Спеціалізовані детектори виконують пошук відомих вірусів по їх сигнатурі (повторюваній ділянці коду). Недолік таких детекторів полягає в тому, що вони нездатні виявляти усі відомі віруси. Детектор, що дозволяє виявляти кілька вірусів, називають полідетектором.

Доктором називають антивірусну програму, що дозволяє виявляти й знешкоджувати віруси. При знешкодженні вірусів середовище знаходження може відновлюватися, або не відновлюватися. Програми-доктори, що дозволяють відшукувати й знешкоджувати велике число вірусів, називають поліфагами. До їхнього числа належать програми *Aidstest*, *Doctor Web* і *Norton AntiVirus*, *AVP*, які отримали широке розповсюдження.

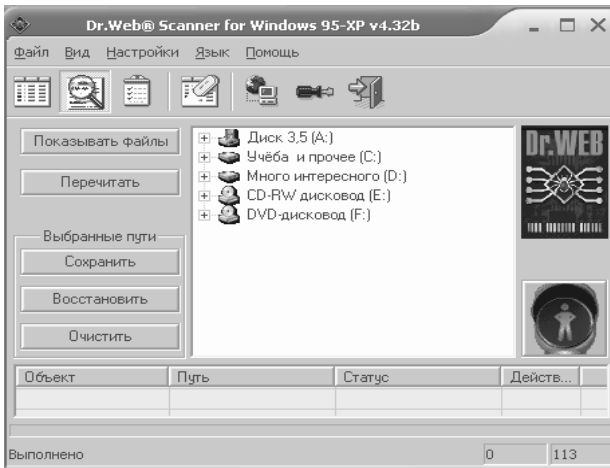
Ревізор являє собою програму, що запам'ятовує вхідний стан програм, каталогів і системних областей і періодично порівнює поточний стан з вхідним. Порівняння може виконуватися по ряду параметрів, таких як довжина і контрольна сума файлу, дата й час зміни і т.п. Достоїнством ревізорів є їхня здатність виявляти стелс-віруси і зміни в програмах, що вносяться вірусами. До числа ревізорів відноситься добре відома програма *ADInf*.

Імунізатор являє собою резидентну програму, призначену для запобігання зараження ряду відомих вірусів шляхом їхньої вакцинації. Суть вакцинації полягає в модифікації програм або диска таким чином, щоб це не відображалося на нормальному виконанні програм і в той же

час віруси сприймали їх як уже заражені і тому не намагалися впровадитися. Істотним недоліком таких програм є їх обмежені можливості по запобіганню зараження від великого числа різноманітних вірусів.

## Характеристика антивірусних програм

Серед широкої безлічі антивірусних програм у вітчизняного користувача найбільшу популярність придбали програми – поліфаги **Doctor Web** (рисунк 2.17) і **AVP** (рисунк 2.18). Обидві програми побудовані як програми оболонки, орієнтовані на роботу в конкретному середовищі, і ядро, що не залежить від середовища. Подібна організація антивірусних програм дозволила:



Рисунк 2.17 – Вікно програми *Doctor Web*

- використовувати єдині файли вірусних баз поза залежністю від типу використовуваної операційної системи: DOS, Windows 95-2000, OS/2, Novell Netware;
- підключати ядро до різних оболонок і додатків, що дає можливість простої інтеграції антивірусної перевірки з багатьма прикладними задачами;
- реалізувати механізм автоматичного поповнення вірусних баз і відновлення версій оболонки і ядра через мережу Інтернет.

Програма Doctor Web призначена для боротьби з поліморфними вірусами, здатна виявляти зміни у власному тілі. У комплект програм для Windows входить поліфаг *Dr.Web* і резидентний сторож *SpIDer Guard*, *SpIDer Mail*.

Програма – поліфаг виявляє й видаляє фіксований набір відомих вірусів у пам'яті, файлах і системних областях дисків комп'ютера.

Резидентний сторож (програма *Monitor*), знаходячись у пам'яті комп'ютера, постійно контролює вірусоподібні ситуації, зроблені різноманітними програмами з диском і пам'яттю. *SpIDer Guard* перехоплює звертання до файлів і системних областей дисків, здійснюючи перевірку на наявність у них комп'ютерних вірусів "на льоту". При виявленні вірусу *SpIDer Guard* починає дії по знешкодженню (лікуванню, видаленню, переміщенню в задалегідь задану область) або блокуванню інфікованого файлу (заборона доступу до інфікованого файлу). Дії можуть починатися у автоматичному (без утручання користувача) або напівавтоматичному режимах. У напівавтоматичному режимі користувач самостійно визначає тип конкретної дії з інфікованим файлом. Таким чином, при активізованому сторожі, доступ до файлів і/або системним областям дозволяється тільки у випадку, якщо віруси не виявлені, або їх удалося знешкодити.

*SpIDer Mail* для робочих станцій Windows – поштовий антивірусний сторож. Програма перехоплює звертання будь-яких поштових клієнтів комп'ютера до поштових серверів по протоколах POP3/SMTP, виявляє й знешкоджує поштові віруси до одержання листів поштовим клієнтом із сервера або до відправлення листа на поштовий сервер.

Починаючи з версії 4.20 у комплект програм входить Планувальник *Dr.Web*, що дозволяє робити запуск антивірусних програм і перевірку пристроїв збереження інформації за графіком, що задається користувачем. Планувальник *Dr.Web* (програма *Drwebsch.exe*) дозволяє робити запуск програм за графіком. Кожному пунктові графіка, який називається завданням, можна сопоставити ім'я антивірусної програми, що запускається, (*Drweb32.exe*, *Drwebwcl.exe*, *Drweb386.exe* або *Drweb.exe*), передати їй необхідні параметри і задати час виконання. Завдання бувають виконувані однократно, представляючи із себе фактично відкладену перевірку, і періодичними. Період виконання завдання може бути заданий рівним 1 години, 1 дневі, 1 тижневі, 1 місяцеві і 1 року.

Для запуску перевірки необхідно виділити у вікні програми потрібні диски і клацнути по зеленій кнопці з чоловічком. Після закінчення перевірки програма запросить наступний диск. По закінченні роботи буде виданий звіт про виявлені й вилучені віруси.

За допомогою меню **Options** можна уточнити параметри роботи програми – які файли перевіряти, лікувати їх чи ні, тестувати пам'ять і т.п.

Dr.Web Модуль автоматичного відновлення для Windows – дозволяє зареєстрованим користувачам одержувати відновлення вірусних баз і інших файлів комплексу, а також робить їхню автоматичну установку.

Велике поширення одержала також програма -поліфаг *AVP*, що може бути встановлена резидентно. По своїх можливостях виявлення вірусів вона порівнянна з програмою *Doctor Web*. Визначеними накладними витратами резидентної установки програми *AVP* є природне уповільнення в роботі комп'ютера. На рисунку 2.18 показано вікно програми *AVP*, в якому треба установити запитувані параметри (наприклад, виставити прапорець **Диск 3,5(A:)** і **ОК**).

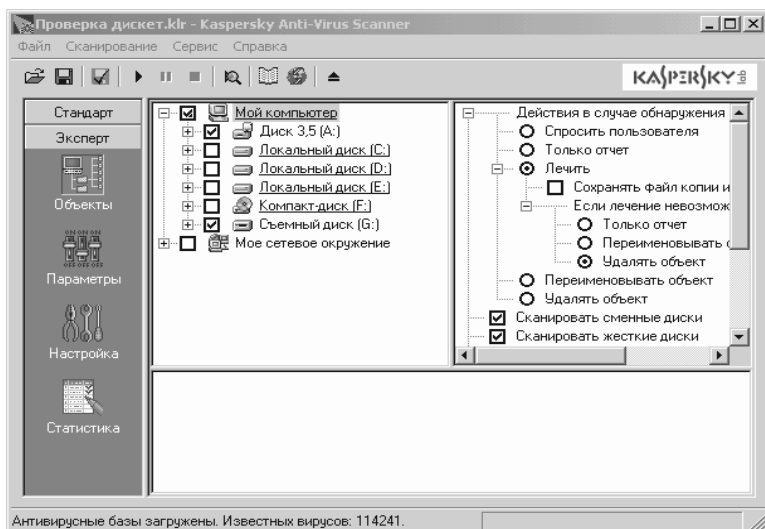


Рисунок 2.18 – Вікно антивірусної програми *AVP*

У ході роботи *AVP* сканує: оперативну пам'ять (DOS, XMS, EMS), файли, включаючи архівні й упаковані, системні сектори, що мі-

стятъ Master Boot Record, завантажувальний сектор (Boot-сектор) і таблицю розбивки диска (Partition Table).

Основні особливості AVP (набудовуються за допомогою режимів **Объекты, Параметры, Настройка**):

- Детектування й видалення величезного числа найрізноманітніших вірусів, у тому числі: поліморфних або вірусів, що самошифруються; стелс-вірусів або вірусів-невидимок; нових вірусів; макро вірусів, що заражають документи Word і таблиці Excel.
- Сканування усередині упакованих файлів.
- Сканування усередині архівних файлів.
- Сканування об'єктів на гнучких, локальних, мережних і CD-ROM дисках.
- Евристичний модуль *Code Analyzer*, необхідний для детектування невідомих вірусів.
- Пошук у режимі надлишкового сканування.
- Перевірка об'єктів на наявність у них змін.
- “AVP Monitor” – резидентний модуль, що знаходиться постійно в оперативній пам'яті комп'ютера і відслідковує усі файлові операції у системі. Він дозволяє знайти й видалити вірус до моменту реального зараження системи в цілому.
- Зручний користувальницький інтерфейс.
- Створення, збереження й завантаження великої кількості різних настроювань.
- Механізм перевірки цілісності антивірусної системи.
- Могутня система допомоги.

Для підтримки якості роботи програми необхідно періодично обновляти бази даних:

**м. Сервис → к. Обновить антивирусные базы.**

При роботі в глобальній мережі, відкiля активно намагаються проникнути в комп'ютер віруси, бажано виконувати перевірку декількома програмами. Можна використовувати програму *Ad-aware* (рисунок 2.19), що виявляє й видаляє складні, змінливі варіанти відомого Datamining, агресивних паразитів, Keyloggers, традиційних Trojans, і ін.

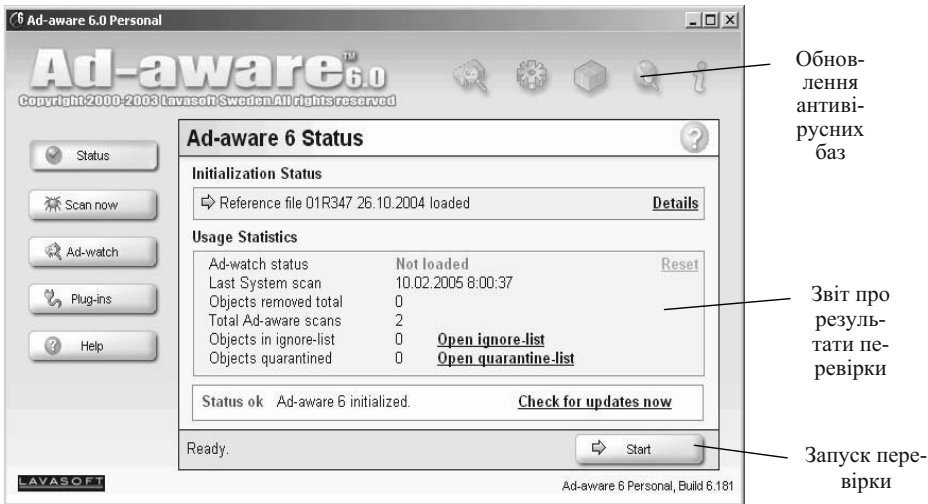


Рисунок 2.19 – Вікно антивірусної програми *Ad-aware*

## 2.2.2 Обслуговування дисків

Основними операціями по обслуговуванню магнітних дисків є наступні: усунення дефектів на дисках, оптимізація розміщення інформації на дисках із метою прискорення доступу до неї, чищення магнітних дисків від непотрібної інформації.

### Усунення дефектів на дисках

Дефекти, що мають місце на магнітних дисках, розділяють на логічні й фізичні. Логічні дефекти полягають у порушенні файлової структури диска або вмісту системної області диска – завантажувального запису й таблиці розміщення файлів. Причинами появи логічних дефектів можуть бути збої у роботі комп'ютера, неправильні дії користувача або деструктивні дії комп'ютерних вірусів. При цьому можлива поява так званих загублених кластерів (недоступних із жодної папки) або сполучених файлів (що мають загальні кластери). У результаті логічних дефектів може виникати засмічення дискового простору, мати місце неможливість доступу до елементів файлової структури диска, неправильна обробка інформації через взаємовплив файлів.

Фізичні дефекти виявляються в неможливості правильного читання і/або запису даних на окремих ділянках магнітного диска через механічні ушкодження, незадовільної якості або старіння магнітного покриття диска. Вчасно виявлені фізичні дефекти небезпеки не представляють, оскільки кластери з дефектними секторами позначаються як дефектні і надалі не використовуються. Нові, але не виявлені фізичні дефекти можуть привести до втрати значної частини даних. Особливо небезпечні фізичні дефекти в системній частині диска, тому що при цьому можуть виявитися недоступними цілі фрагменти файлової структури.

Для пошуку й усунення дефектів на магнітних дисках застосовуються спеціальні утиліти, що одержали назву **дисккових сканерів-коректорів**. Серед таких утиліт широке поширення одержали *Microsoft ScanDisk*, що входить до складу Windows, і *Norton Disk Doctor* із комплекту Norton Utilities. Утиліти мають однакову схему роботи: спочатку виконується перевірка файлової структури диска для пошуку й усунення логічних дефектів, потім проводиться перевірка поверхні диска для пошуку й усунення фізичних дефектів.

Знайдені ланцюжки вільних кластерів відповідно до реакції користувача перетворюються у файли або об'являються вільними. Сполучені файли можуть бути перерозміщені для поділу. При виявленні фізичного дефекту розташовані на дефектній ділянці дані по можливості переміщуються в інше місце. Природно, при цьому частина даних може виявитися втраченою.

Для запуску сканера-коректора *Microsoft ScanDisk* необхідно:

викликати контекстне меню для об'єкта, що перевіряється → к. **Свойства** → вкладка **Сервис** → кнопка **Выполнить проверку** (у результаті відкривається стартове вікно сканер-коректора) → виставити необхідні прапорці (рисунок 2.20) → кнопка **Запуск**.

Установка прапорця **Автоматически исправлять системные ошибки** означає автоматичне виправлення виявлених помилок сканер-коректором без видачі запитів користувачеві.

Після закінчення роботи сканера-коректора з'являється панель, що відображує звіт про результати.

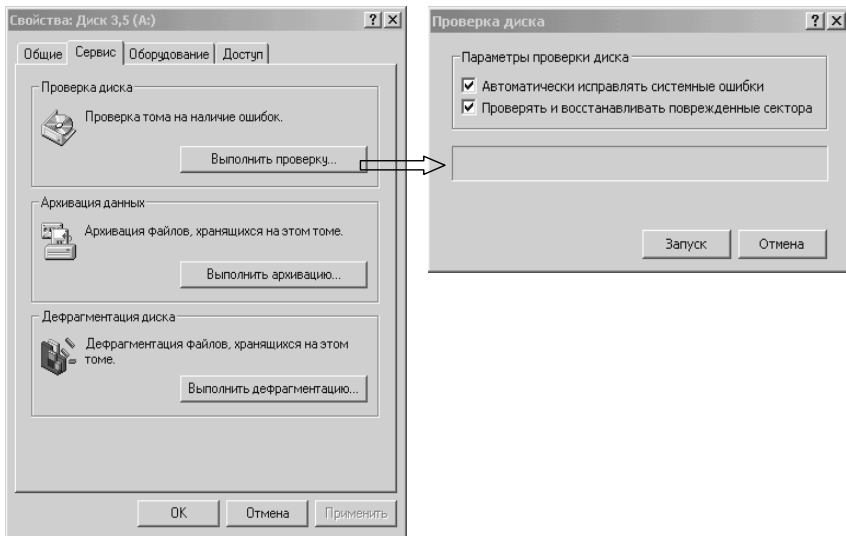


Рисунок 2.20 – Вікна запуску сканер-коректора для перевірки дисків

## Прискорення доступу до даних на диску

Прискорення доступу до даних на диску досягається шляхом дефрагментації (усунення фрагментації) файлів, а також розміщення дескрипторів файлів таким чином, щоб забезпечити більш швидкий доступ до найчастіше використовуваних файлів.

Фрагментація файлів означає розміщення суміжних кластерів файлу на декількох несуміжних фрагментах ( екстентах ) дискового простору. Причиною, що викликає фрагментацію, є та обставина, що пам'ять під файли приділяється в міру потреби. Тому при виділенні додаткових кластерів файлові вони можуть бути розміщені на несуміжній ділянці диска. Суть дефрагментації складається в такому перерозміщенні файлів, щоб вони займали безперервні ділянки дискової пам'яті. Корисним додатковим ефектом проведення дефрагментації є підвищення безпеки збереження даних, оскільки нефрагментовані файли легше відновлювати.

Для прискорення доступу до деякого файлу його варто перемістити на початок папки. У цьому випадку операційній системі буде потрібно виконати меншу кількість операцій зчитування й аналізу дескрипторів. Звідси витікає, що файли в папках бажано розміщати в порядку убування частот їхнього використання.

Крім того, для прискорення доступу до магнітних дисків часто використовувані компоненти файлової системи розміщують на початку, а



рідко використовувани – ближче до кінця дискового простору. Це зв'язано із застосовуваним напрямком переміщення блоку магнітних голівок від зовнішніх циліндрів до циліндрів біля осі обертання диска. До часто використовуваних компонентів файлової системи відносяться, насамперед, папки і потім часто використовувані файли .

Найбільший внесок у прискорення доступу до магнітних дисків із числа зазначених способів вносить усунення фрагментації файлів.

Для проведення дефрагментації дисків можна використовувати утиліту **Speed Disk** із комплекту Norton Utilities.

У складі Windows мається дефрагментатор **Disk Defragmenter**. Зазначена програма забезпечує дефрагментацію диска й перевірку наявності логічних дефектів.

Для запуску програми **Disk Defragmenter** досить виконати:

кнопка **Пуск** → м. **Программы** → к. **Стандартные** → п. **Служебные программы** → п.п. **Дефрагментация диска** (відкриється діалогове вікно – рисунок 2.21) → вибрати магнітний диск і натиснути потрібну командну кнопку ( **Анализ** або **Дефрагментация**).

Аналогічного результату можна досягти, виділивши об'єкт, а потім меню **Файл** → к. **Свойства** → вкладка **Сервис** → кнопка **Выполнить дефрагментацию**.

Результат аналізу, а потім дефрагментації виводиться в графічному вигляді. При невисокому ступені фрагментації проводити дефрагментацію диска недоцільно через малу ефективність.

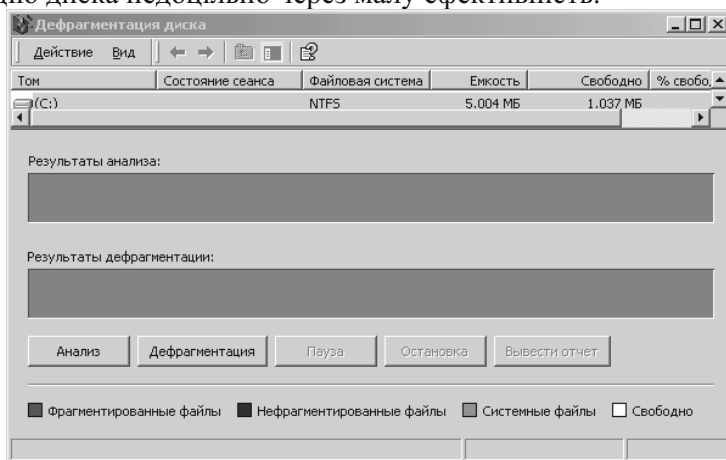


Рисунок 2.21 – Вікно програми **Disk Defragmenter**

## Чищення магнітних дисків і форматування дискет

Чищення магнітних дисків являє собою процес видалення, наприклад, стандартним способом, різного роду непотрібних файлів для вивільнення дискового простору. До числа непотрібних файлів, що з'являються в процесі роботи на комп'ютері, можна віднести тимчасові, резервні, файли, що не використовуються, і файли-дублікати.

В операційній системі Windows до тимчасових файлів відносяться файли з розширенням **.tmp** і усі файли й папки, розміщені в підпапках TEMP, TMP, Temporary Internet Files папок WINNT, Local Settings.

До зайвих файлів відносяться файли, що залишаються після збійних ситуацій, звичайно їхні імена починаються з тільди «~» і повторюють частину імені файлів, що працювали під час збою. Ці файли можуть з'являтися й у звичайних папках, їх потрібно видаляти.

Рідко використовувані файли визначаються по даті останнього доступу до файлів. Тимчасові, наприклад, через:

кнопку **Пуск** → м. **Найти** → к. **Файлы и папки** → виставити **Параметры** → виставити прапорець **Дата**.

При роботі з дискетами іноді виникає необхідність цілком очистити дискету і заново підготувати її до роботи. Найбільш надійний спосіб для цього – форматування.

У процесі форматування вся інформація з дискети цілком стирається і виконується розбивка поверхні гнучкого диска на доріжки й сектори, перевірка якості поверхні, формування змісту дискети.

Виконати дану роботу можна, виділивши значок диска **A:\** у вікні програми **Мой компьютер** або **Проводник**, а потім м. **Файл** → к. **Форматирование** → установити параметри у вікні **Форматирование: Диск 3,5 [A:]** (рисунок 2.22) → кнопка **Начать**.

Ведення архіву програмного забезпечення і користувальницьких файлів припускає рішення двох задач: резервного копіювання й архівування.

Основне призначення програм резервного копіювання, наприклад, **MICROSOFT BACKUP** – створення копій вхідної інформації на резервних носіях: стримерах, жорстких дисках або дискетах. Звичайно програми такого типу використовуються при періодичному збереженні на резервних носіях великих обсягів інформації і робота з ними спеціально регламентується керівництвом офісу.

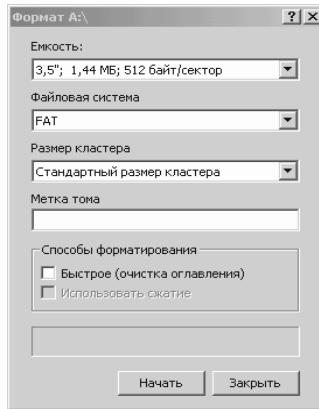


Рисунок 2.22 – Вікно програми форматування дискет

### 2.2.3 Програми архівації

Призначення програм – архіваторів – заощаджувати місце на диску за рахунок стиску одного або декількох вхідних файлів в архівний файл. Програми – архіватори використовуються у випадках: наявності на дисках комп'ютера великих обсягів інформації, що знадобиться тільки в майбутньому; переносу між комп'ютерами масивів інформації за допомогою дискет; створення резервних копій файлів у стиснутому виді. У результаті роботи програм – архіваторів створюються архівні файли (архіви).

В основі роботи програм-архіваторів лежить процедура пошуку й перекодування однакових фрагментів вмісту файлів. Механізм використання даної процедури ілюструється наступним прикладом. Нехай маєтся файл, що містить безліч однотипних слів: "комп'ютер", "комп'ютера", "комп'ютерна", "комп'ютеризація" і т.п. Якщо сполучення 9 букв "комп'ютер" замінити простою буквеною комбінацією "шщ", то зазначена система слів перетвориться в систему: "шщ", "шща", "шщная", "шщизація" і т.п. Ефект від зменшення розміру вхідного тексту при такій заміні очевидний. У реальних програмах- архіваторах процедура пошуку й перекодування набагато складніше.

Відома безліч програм – архіваторів: *ARJ*, *PKZIP/PKUNZIP*, *PAK*, *LHA*, *ICE*, *RAR* під операційну систему DOS; *WinARJ*, *WinZIP*, *WinRAR* під операційну систему Windows. Ці програми відрізняються форматом архівних файлів, швидкістю роботи, ступенем стиску файлів

в архіві, інтерфейсом користувача. Якщо розширення створюваного архівного файлу не зазначено явно, то мається на увазі розширення, що корелює з ім'ям програми архіватора. Наприклад, файли з розширенням ARJ утворюються програмою архівації *ARJ*.

Типові функції програм архівації файлів складаються в приміщенні вхідних файлів в архів, витягу файлів з архіву, видаленні файлів з архіву, перегляді змісту архіву, верифікації (перевірки) архіву і т.ін.

У посібнику описана реалізація типових функцій програм архівації на прикладі архіваторів *WinZIP* і *WinRAR*.

## Архіватор WinRAR. Розміщення файлів в архіві


Можливості *WinRAR*: повна підтримка архівів RAR і ZIP; оригінальний високоефективний алгоритм стиску даних; спеціальний алгоритм мультимедіа – стиску; оболонка з підтримкою технології перетягнути-і-залишити (drag & drop); керування архівами інших форматів (CAB, ARJ, LZH, TAR, GZ, ACE, UUE); підтримка безперервних (solid) архівів, у яких ступінь стиску може бути на 10 – 50% більше, чим при звичайних методах стиску, особливо при упакуванні значної кількості невеликих схожих файлів; підтримка багатотомних архівів; створення архівів, що саморозпаковуються (SFX) звичайних і багатотомних архівів; відновлення фізично ушкоджених архівів; інші додаткові функції, наприклад, шифрування, додавання архівних коментарів і ін.

Варто помітити, що формат архіву RAR значно краще оптимізований для складних задач із використанням великої кількості файлів і гигабайтних дискових просторів.

Насамперед, потрібно запустити *WinRAR* одним із способів:

1. Двічі клацнути мишею або натиснути [Enter] на значку **WinRAR**.
2. Кнопка **Пуск** → м. **Программы** → п. **WinRAR** → і виберіть п.п **WinRAR**.

При запуску *WinRAR* показує у своєму вікні (рисунок 2.23) список файлів і папок у поточній папці, а користувачеві необхідно показати файли, призначені для архівації. У полі пристрою необхідно вказати диск, на якому розташовані папки й файли, що підлягають включенню в архів.

Для переходу в батьківську папку використовуються клавіші **BackSpace**, [Ctrl+PgUp], маленька кнопка  під панеллю інструментів або варто двічі клацнути на папці “...”у списку файлів.

Для переходу в іншу папку потрібно натиснути [Enter], [Ctrl+PgDn] або двічі клацнути мишею на цій папці.

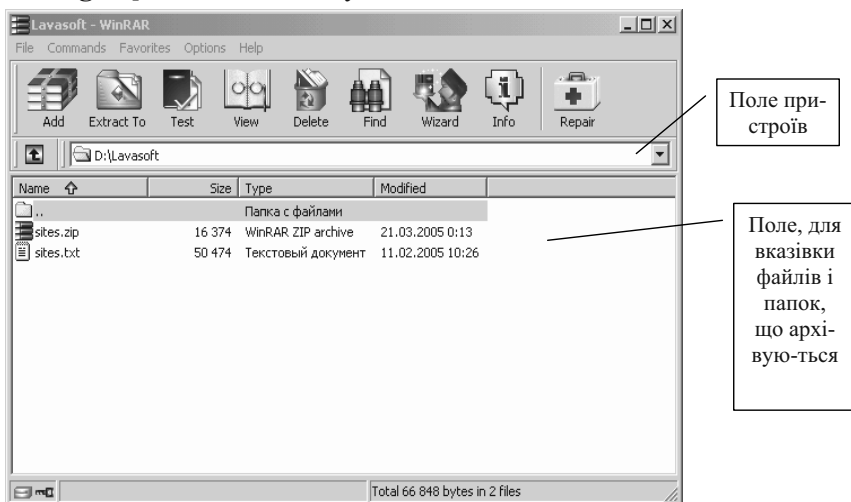


Рисунок 2.23 – Вікно архіватора WinRAR

У полі папок і файлів, що архівуються, варто виділити об'єкти для архівації і потім клацнути по інструментальній кнопці **Add** або м. **File** → к. **Add**. У діалоговому вікні (рисунок 2.24), що з'явилося, на вкладці **General** варто вказати ім'я й шлях до архіву. Для уточнення місця розташування архіву можна використовувати кнопку **Browse**. За допомогою кнопки **Profiles** можна уточнити тип файлу (для електронної пошти, резервний, ZIP-архів, том на дискеті). У полі **Update mode** задається операція над об'єктами, що поміщаються в архів. Селекторні кнопки вибору дозволяють уточнити формат створюваного архіву. У поле **Compression method** задається ступінь стиску архіву, а в поле **Volume size bytes** – розмір створюваного тому (наприклад, для дискети 1467434), у блоці **Archiving options** за допомогою прапорців указується тип і параметри створюваного архіву (звичайний – **solid** – з розширенням.rar, що саморозвертається – **SFX** – із розширенням.exe, архівувати мультимедійні файли і т.д.)

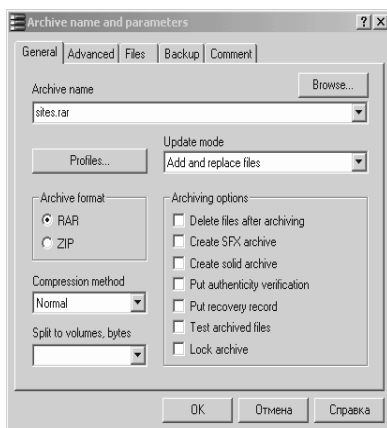


Рисунок 2.24 – Вікно, викликуване командою **Add** архіватора *WinRAR*

Під час архівації відображається вікно зі статистикою. Якщо потрібно перервати процес стиску, клацніть на кнопці **Отмена**. По закінченні архівації вікно статистики зникне, а створений архів стане поточним виділеним файлом.

Додавати файли в існуючий архів RAR можна також за допомогою перетаскування. Виділіть архів у вікні **WinRAR** і натисніть клавішу **Enter** (або двічі клацніть мишею) на його імені – *RAR* прочитає архів і покаже його вміст. Тепер можна додати файли в архів, просто перетягнувши їх з іншої програми у вікно **WinRAR**.

Приклад 2.3. Створити простий і архів, що саморозгортається, для папки **Мої документи** на **Робочому столі**.

#### Виконання:

1. Запустити програму *WinRAR* → у вікні програми зі списку поля пристроїв, що розкривається, вибрати **диск С:** → виділити **папку Мої документи** → клацнути по кнопці **Add** → у діалоговому вікні (рисунок 2.23) клацнути по кнопці **Browse** → у вікні **Find archive**, що з'явиться, у полі **Папка** вказати **Робочий стіл**, ім'я файлу залишити **Мої документи.rar** → кнопка **Открыть** → у вікні **Archive name and parameters** виставити прапорець **Create solid archive** → **OK** → закрити вікно архіватора.

2. Для створення архіву, що саморозвертається, запустити програму *WinRAR* → у вікні програми зі списку поля пристроїв, що розкривається, вибрати **диск С:** → виділити папку **Мої документи** → кла-

кнунути по кнопці **Add** → у діалоговому вікні (рисунок 2.24) клацнути по кнопці **Browse** → у вікні **Find archive**, що з'явиться, у полі папка вказати **Робочий стіл**, ім'я файлу залишити *Мої документи* → кнопка **Открыть** → у вікні **Archive name and parameters** виставити прапореч **Create SFX archive** (розширення архіву стане **.exe**) → **OK**.

Якщо обидва архіви створюються одночасно, то замість закриття вікна архіватора варто виконати підкреслену послідовність дій з п.2.

## Витяг файлів з архіву WinRAR

Щоб витягти файли за допомогою програми *WinRAR*, спочатку потрібно відкрити архів у *WinRAR*. Це можна зробити декількома способами:

- двічі клацнути мишею або натиснути клавішу **[Enter]** на файлі архіву в середовищі Windows (у *Проводнике* або на **Робочому столі**). Якщо *WinRAR* був зв'язаний з типами файлів-архівів під час установки (що робиться за замовчуванням), то архів буде відкритий у *WinRAR*;
- двічі клацнути мишею або натиснути **[Enter]** на файлі архіву у вікні **WinRAR**;
- перетягнути архів на значок або вікно **WinRAR**. Перед тим як це зробити, переконайтеся, що у вікні **WinRAR** не відкритий інший архів, інакше архів, що перетягується, буде доданий у відкритий.

При відкритті архіву у вікні **WinRAR** виводиться його вміст (рисунок 2.25).

Варто виділити ті файли й папки, які необхідно витягти. Це можна зробити клавішами керування курсором або лівою кнопкою миші при натиснутій клавіші **[Shift]** (як у *Проводнике* й інших програмах *Windows*), клавішами **[Пробіл]** або **[Ins]**. Клавіші **[+]** і **[-]** на цифровій клавіатурі дозволяють виділяти й знімати виділення з групи файлів за допомогою шаблонів.

Виділивши один або кілька файлів, клацніть на кнопці **Extract to** угорі вікна **WinRAR**.

Якщо ви хочете витягти файли не в поточну папку, введіть у вікні, що з'явилося, потрібний шлях і натисніть **Ok**.

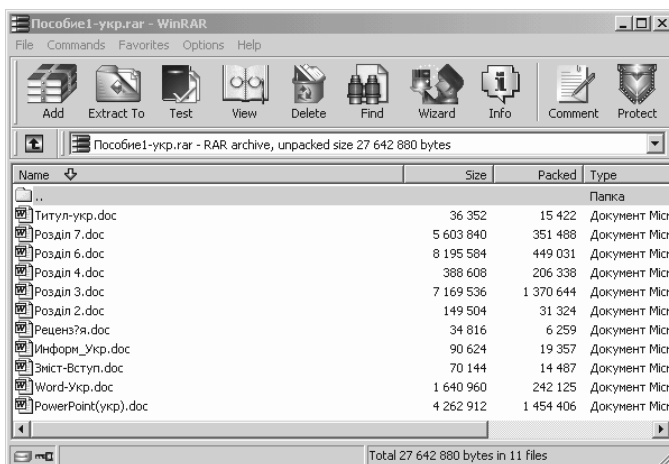


Рисунок 2.25 – Вікно **WinRAR** для витягу файлів і папок з архіву

**Приклад 2.4.** Розгорнути архів *Мої документи.rar* на диску **D:**, а архів, що саморозвертається – на **Робочому столі**.

#### Виконання.

1. Подвійне клацання миші по значку *Мої документи.exe* розгорне архів прямо на **Робочому столі** (з'явиться папка **Мої документи**).

2. Двічі клацнути мишею по значку *Мої документи.rar* → виділити папку **Мої документи** → клацнути по кнопці **Extract to** → у вікні праворуч вибрати папку диска **D:** (вона повинна з'явитися у полі **Destination path**) → **OK**.

### **Робота з архіватором WinZIP**

Архіватор *WinZIP* підтримує архіви ARJ, LZH, ARC і надає досить зручний і очевидний сервіс для користувача (рисунок 2.26).

Основне меню архіватора складається з пунктів меню **File, Actions, Options, Help**, що містять згруповані по функціональному призначенню команди архіватора.

Команди пункту меню **File** спрямовані на роботу з архівом, як файлом. Так, використання команди **New Archive** дозволяє створити на диску новий архів. Після вибору команди на екрані з'являється діалогове вікно, в якому необхідно вказати: диск і директорій, у якому буде створюватися архів і ім'я цього архіву (аналогом виклику даної команди є натискання кнопки **NEW** піктографічного меню).



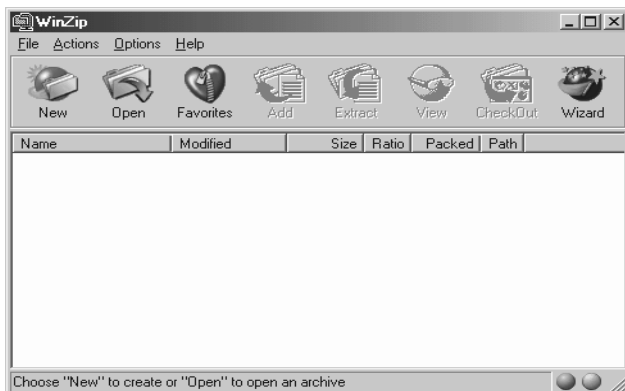


Рисунок 2.26 – Вікно архіватора *WinZIP*

Команда **Open Archive** виконує перегляд архіву. З використанням полів діалогового вікна команди вибирається необхідний архів. Команду можна виконати натисканням кнопки **Open** піктографічного меню. У результаті у вікні архіватора *WinZIP*, буде приведений список усіх файлів, що знаходяться в архіві. У цьому списку зазначені імена файлів, що знаходяться в архіві (стовпчик Name), дата й час їхнього створення (стовпчика Date і Time), розмір вхідних файлів (стовпчик Size), ступінь стиску файлів програмою *WinZIP* (стовпчик Ratio), розмір файлів після стиску (стовпчик Packed) і місце перебування файлів в архіві (стовпчик Path)

**Close Archive** команда закриття архіву, відкритого у вікні архіватора *WinZIP*.

Вибір команди **Information** дозволяє одержати інформацію про архів (ім'я і місцезнаходження архіву; розмір і кількість файлів в архіві; коефіцієнт стиску архіву; дата й час останньої модифікації архіву).

Сукупність команд **Move...**, **Copy...**, **Rename...**, **Delete Archive** реалізують відповідні дії з архівним файлом (переміщення архіву, створення копії архіву, перейменування й видалення архіву).

Команда **Mall Archive** дозволяє здійснити пересилання архіву по електронній пошті.

Другий пункт **Actions** основного меню програми архіватора *WinZIP* містить команди обробки вмісту архіву.

Застосування команди **Add** – поміщає файли в архів. При виборі цієї команди на екрані з'являється діалогове вікно **Add** (рисунок 2.27).

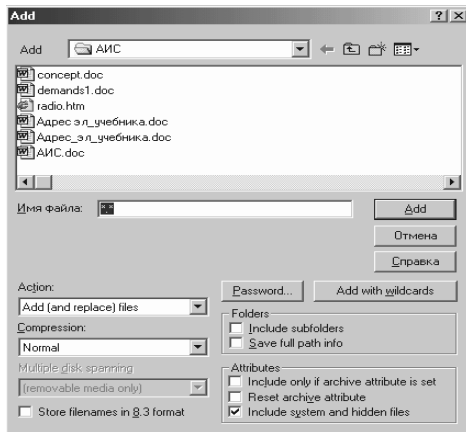


Рисунок 2.27 – Вікно команди Add архіватора WinZIP

У полі **Add from directory/drive** цього вікна задається місцезнаходження файлів, що поміщаються в архів.

У полі **Select files** вибираються файли, що поміщаються в архів.

У полі **Add Files** задається шаблон для маркірування (виділення) групи файлів.

Поля **Action** і **Compression** відповідно задають спосіб приміщення файлів в архів (копіювання, пересилання, пересилання нових версій і ін.) і ступінь стиску файлів (максимальне, нормальне, швидке і дуже швидке).

Прапорець **Recurse subdirectories** встановлює опцію збереження структури директоріїв разом із піддиректоріями при приміщенні в архів інформації.

Команда **Delete** дозволяє коректувати вміст архіву за рахунок видалення файлів, що знаходяться в ньому. Перед використанням команди рекомендується попередньо виділити файли, що видаляються.

Команда **Extract** забезпечує витяг файлів з архіву. При виборі команди на екрані з'являється діалогове вікно. У його полі **Extract to** задається ім'я папки, куди потрібно помістити файли, що витягаються з архіву. З використанням поля **Files** визначаються імена файлів, що витягаються.

Команда **View** виконує перегляд вмісту файлу, що знаходиться в архіві.

При виборі команди, користувач має можливість скористатися стандартним додатком, у якому був створений вхідний файл. Перегляд файлу здійснюється подвійним клацанням курсору миші по імені файлу або вибором кнопки **View** піктографічного меню.

Команди **Select all i Deselect all** встановлюють і знімають маркування для усіх файлів, що знаходяться в архіві.

Команда **Test** здійснює верифікацію (перевірку цілісності) умісту архіву.

Команда **Check Out** дозволяє перенести обрані файли на **Робочий стіл**.

З використанням команд третього пункту меню **Options** користувач має можливість змінити розмір і стиль використовуваних у вікні програми *WinZIP* шрифтів (команда **Font**), захистити архів паролем (команда **Password**) або відсортувати (упорядкувати) збережену в архіві інформацію (команда **Sort**). Упорядкування можна зробити в лексикографічному порядку зростання імен або розширень файлів; розміру вхідних або стиснутих файлів; ступеня стиску; дати і часу переміщення до архіву.

Команди четвертого пункту основного меню **Help** описують можливості роботи архіватора *WinZIP*, пояснюють техніку роботи з ним і містять інформацію про фірму розроблювача й процедури придбання архіватора.

#### Приклад 2.5.

Створити за допомогою програми *WinZIP* на **Робочому столі** архів п'яти перших файлів папки **Мої документи**. Архівові дати ім'я *Мій архів*.

#### Виконання.

Запустити програму *WinZIP* → клацнути по кнопці **New** → у полі **Create** указати **Робочий стіл** → у поле **Ім'я файлу** вказати *Мій архів* → **ОК** → у вікні **Add** у полі **Add** вибрати потрібний диск **C:** → нижче розкрити папку **Мої документи** і виділити 5 перших файлів у ній (їх імена з'являться у полі **Ім'я файла**) → клацнути по кнопці **Add** → закрити вікно архіватора.

#### Приклад 2.6.

Розгорнути *Мій архів* на диску **D:** у папку **aaa**.

### Виконання.

Подвійним клацанням миші по файлу *Мій архів* викликати вікно **WinZIP-архів.zip** → виділити файли, що витягаються з архіву, → клацнути по кнопці **Extract** → у вікні **Extract** (рисунок 2.28) клацанням по кнопці зі стрілкою праворуч від поля **Extract to** відкрити перелік пристроїв і папок → виділити диск **D:** → клацнути по кнопці **New folder** → задати ім'я **aaa** → **OK** → клацнути по кнопці **Extract** → закрити вікно архіватора. На диску **D:** буде створена папка **aaa** й у неї розгорнутий архів.

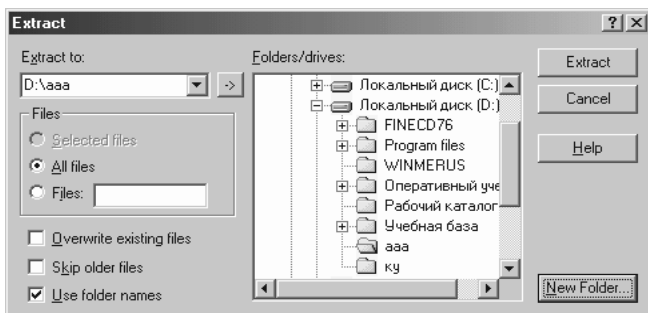


Рисунок 2.28 – Вікно Extract архіватора WinZIP

## 2.2.4 Запис даних на компакт-диски

Сьогоднішні програмні засоби здебільшого представляють об'ємні програмні комплекси, крім того, аудіо і відео інформація має також значні обсяги. Тому сучасним засобом переміщення таких даних є компакт-диски й користувач повинен уміти користуватися ними.

Розглянемо дві популярні програми для виконання зазначених робіт – **WinOnCD** і **Nero**. Дані програми допомагають користувачеві виконувати цілий ряд робіт, починаючи від простого копіювання диска на диск, кінчаючи створенням своїх аудіо й відео альбомів. Але в цьому розділі будуть розглянуті стандартні роботи з копіювання даних на диски. Запис на диск здійснюється у два етапи – програми створюють у пам'яті образ диска, а потім записують його на CD. Пам'ятаємо, що диски CD-R, DVD-R – диски однократного запису, диски CD-DW, DVD-RW дозволяють виконувати багаторазовий запис на диск. Крім того, при роботі можливий запис декількома порціями – сесіями, диски з декількома сесіями називають мультисесійними.

## Програма WinOnCD

Після завантаження програми відкривається вікно вибору типу проекту (рисунок 2.29), на вкладках якого позначені всі можливі типи робіт.

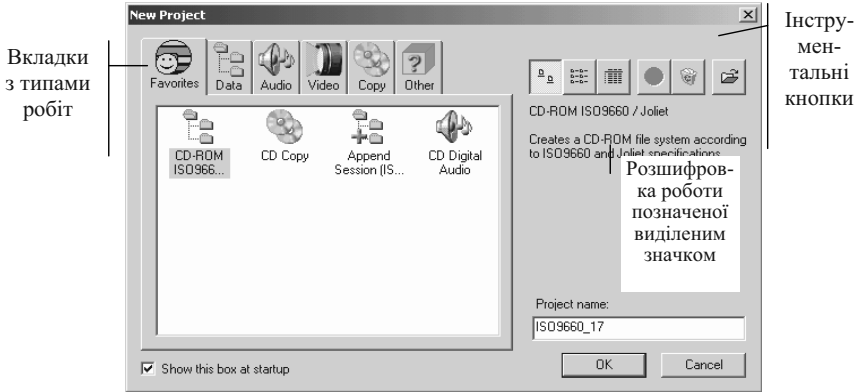


Рисунок 2.29 – Первинне вікно програми WinOnCD

## Перезапис із диска на диск

Уставити компакт-диск у дисковод → викликати вкладку **Сору** → виділити значок **CD Сору** → кнопка **ОК** → у вікні редактора проекту, що з'явилося, (рисунок 2.30) клацнути по кнопці **Record**. Програма

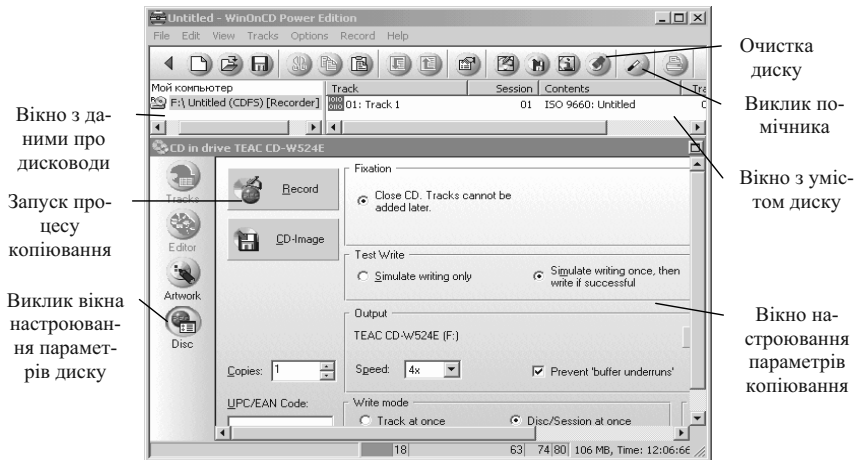


Рисунок 2.30 – Вікно редактора проекту програми WinOnCD

просканує диск, про що дасть повідомлення в інформаційному вікні, і при благополучному результаті видасть вікно **Make CD** (рисунок 2.31 А). Варто клацнути по кнопці **ОК** → з'явиться наступне вікно **Make CD** із графічним зображенням процесу створення образу диска (рисунок 2.31 Б).

Після створення образу диска програма автоматично вивантажує диск і запитує чистий CD. Після завантаження чистого диска автоматично починається перезапис образу диска на CD (рисунок 2.31 В).

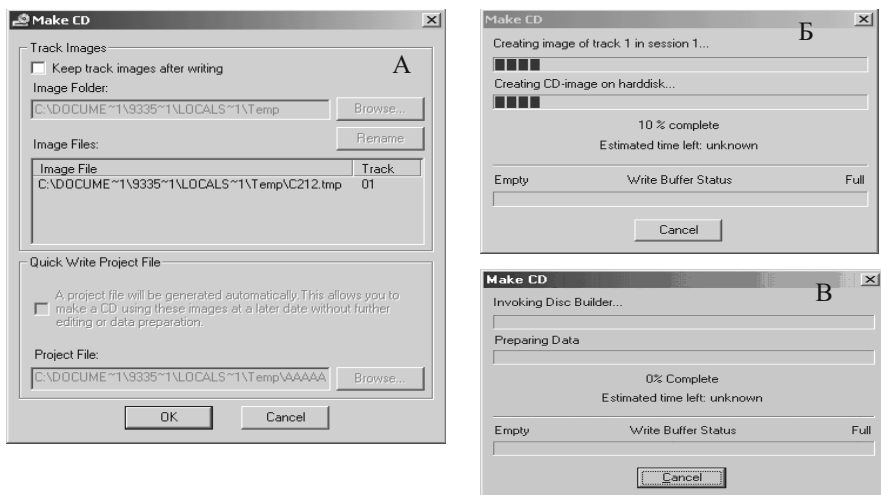


Рисунок 2.31 – Вікна створення образу диска

## Очищення диска

Диски багаторазового використання CD-RW можна очищати й використовувати для запису нових даних. Для цього у вікні редактора проекту (рисунок 2.30) клацнути по кнопці **Очистка**. Буде видане вікно для уточнення й підтвердження операції (рисунок 2.32) → клацнути по кнопці **Blank** → **ОК** → очікувати закінчення процесу.

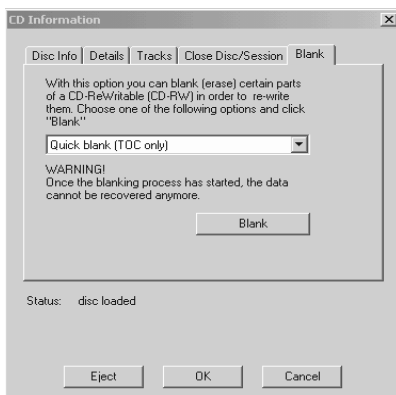


Рисунок 2.32 – Вікно підготовки до очищення диска

## Запис даних з жорсткого диска на CD

Для копіювання файлів із жорсткого диска вибрати вкладку **Data** (рисунок 2.29) → значок **CD-ROM ISO9660/Joliet** → **OK**. У вікні редактора проекту, що відкриється, (рисунок 2.33) верхнє вікно являє собою вікно програми *Проводник*, де користувач вибирає потрібні файли й папки для копіювання. Нижнє вікно представляє аналог вікна провідника, але для диска, що формується. У верхнім вікні необхідно виділити потрібні папки й файли (груповою операцією або послідовно) і перетягнути них у нижнє вікно. Кнопка **Make CD** → кнопка **Record** → у вікні **Make CD** кнопка **OK** → в інформаційному вікні буде графічно показаний процес запису на диск.

Примітка 1. Майте на увазі, що об'єкти, які перетаскуються, потраплять у ту папку, що виділена у вікні CD. Тому, щоб не плутатися при переносі декількох папок, не забувайте робити активним кореневий каталог – це там, де значок диска й номер. До речі, саме цю мітку диска Ви побачите в *Проводнику* Windows, коли диск буде вставлений у CD-ROM. Змінити її можна прямо тут або у властивостях проекту – це рядок **Volume ID** на закладках **Joliet** або **ISO 9660 Volume Descriptor**.

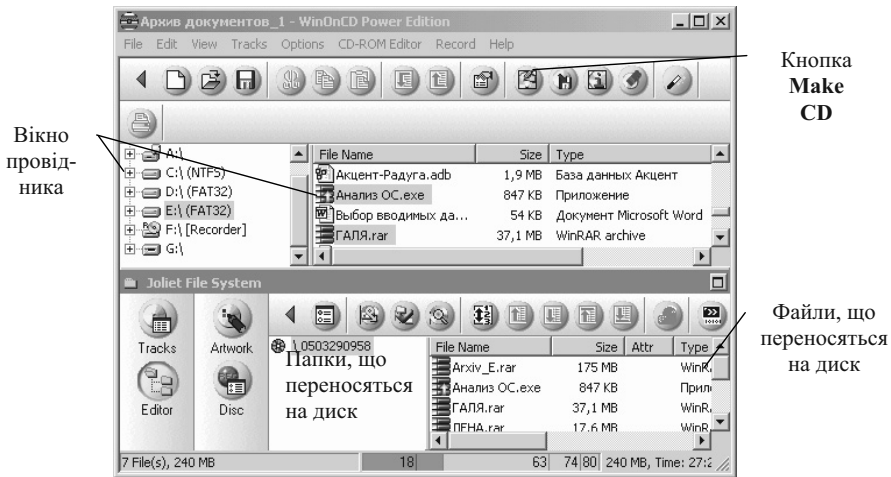


Рисунок 2.33 – Вікно редактора проекту для запису даних на диск

**Примітка 2.** При записі на диск є обмеження на довжину імені файлу або папки (8 символів) і символи, що застосовуються в імені. При занадто довгому імені програма запропонує обрізати його.

Якщо даних не вистачає для заповнення цілого диска (650-700 Мб) – залишіть перемикачі (рисунок 2.34) як є, потім запишете наступну сесію, тобто мультисесійний диск. Якщо набирається більш 500 Мб – диск краще закрити.

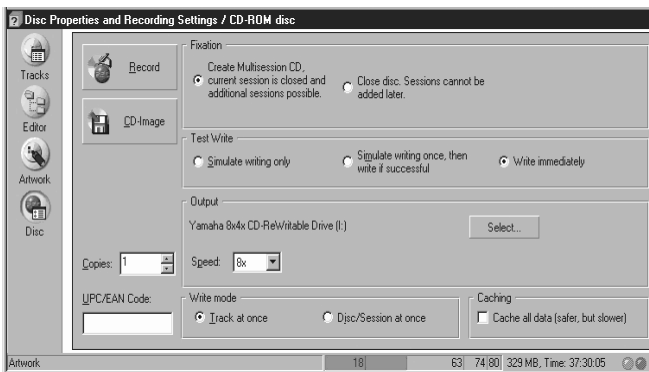


Рисунок 2.34 – Вкладка **Свойства диска**, викликається кнопкою **Disc**

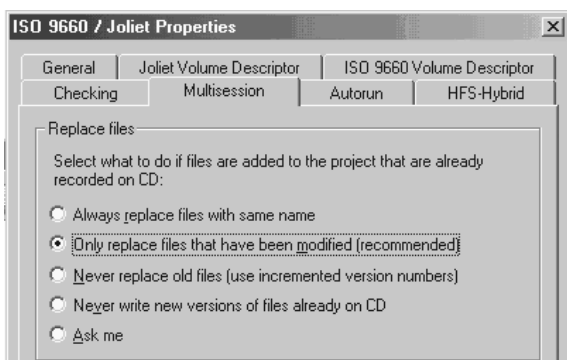


## Запис декількох сесій (мультисесія / Append Session (ISO9660 / Joliet))

Якщо у Вас є диск із даними, а перша сесія на ньому вже закрита, але не закритий сам диск – можете дописати його, тобто створити мультисесійний CD. Це проект типу Append Session (ISO9660 / Joliet).

Якщо попередня записана сесія була у форматі ISO9660 / Joliet, то імпорт даних відбувається автоматично – усе, що вже було на диску, Ви побачите у вікні редактора проекту. Для дозапису даних необхідно у верхнім вікні виділити потрібні об'єкти і перетягнути у відповідну папку диска нижнього вікна, потім клацнути по кнопці **Record**, далі процес запису виконується аналогічно попередній роботі.

Єдина відмінність: у властивостях проекту на закладці Multisession потрібно вказати, що робити, якщо й у тієї й в іншій сесії існують файли або каталоги зі співпадаючими іменами.



Розмір доріжки закриття сесії, у залежності від розміру останньої, складає від 13 до 20 Мб. Усього сесій на диску може бути 20 – так, що економія від запису мультисесійного CD досить відносна – в остаточному підсумку загальний розмір записаних даних може скласти менш 500 Мб. Великий мінус цього виду запису в тім, що TOC (таблиця розміщення даних) пишеться не цілком, як на звичайному диску ISO9660 / Joliet, а частинами – у кожній сесії. Не всі приводи підтримують цю специфікацію, а ті що підтримують – із кожною записаною сесією читають диск усе гірше. Мало того, якщо Ви щось зробили не так, то попередні сесії можуть бути на диску не видні зовсім.

## Програма Nero

Завантаження програми: кнопка **Пуск** → к. **Програми** → п. **Nero** → п.п. **Nero StartSmart** → завантажиться стартове вікно програми для версії 6 (рисунок 2.35). Для більш ранніх версій можна відразу завантажити **Nero Express**: кнопка **Пуск** → к. **Програми** → п. **Ahead Nero** → п.п. **Nero Express** (рисунок 2.36).



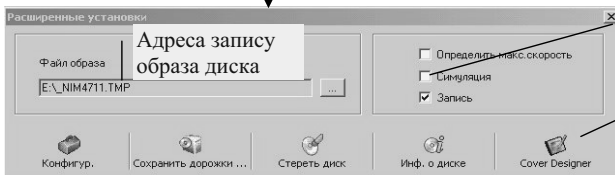
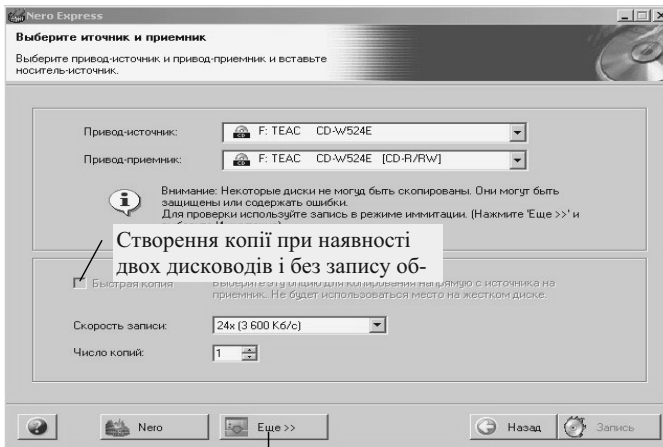
Рисунок 2.35 – Стартове вікно програми *Nero* версії 6

Нижче буде описана робота з програмою *Nero Express* версії 5. Для копіювання даних з диска на диск необхідно вставити диск, що копіюється, в дисковод → вибрати п. **Копіювання всього диска** → у вікні, що відкриється (рисунок 2.37) перевірити виставлені параметри і клацнути по кнопці **Запись**.

Клацання по кнопці **Ещё** відкриває діалогове вікно з розширеними установками, через яке можна запустити очистку диска, уточнити налаштування програми, одержати інформацію про диск, що копіюється, або зберегти інформацію окремими порціями – доріжками.



Рисунок 2.36 – Завантажувальне вікно Nero Express



Для пе-  
ревірки дис-  
ку  
Виклик редактора  
обкладинок диску

Рисунок 2.37 – Вікна установок і запуску копіювання диска на диск

Процес створення образу диска, а потім запису на новий диск по-  
казується графічно у відповідних вікнах.

Для запису даних на диск необхідно вставити диск у дисковод → вибрати п. **Данные** → у діалоговому вікні (рисунок 2.38) клацнути по кнопці **Добавить** → у вікні, що імітує *Проводник* виділити необхідні файли і папки → клацнути по кнопці **Добавить** → підкреслене повторювати поки не будуть обрані всі потрібні об'єкти для копіювання → кнопка **Завершить** → кнопка **Далее** → перевірити фінальні установки запису в наступному вікні (при необхідності установити прапорці **Разрешить добавление записей на диск** и **Проверить данные после записи на диск**) → кнопка **Запись** → почекати виконання запису і перевірки даних (рисунок 2.39) → після повідомлення програми, що процес виконаний, кнопка **Далее** → буде видане вікно з питаннями по продовженню роботи (рисунок 2.40).

У цілому в даній версії програми російський інтерфейс відмінно допомагає в роботі.

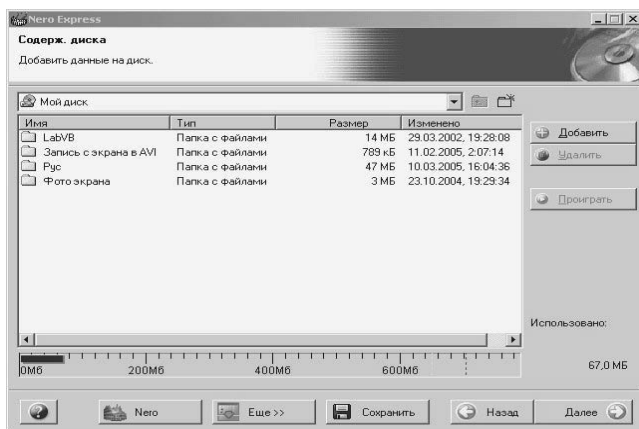
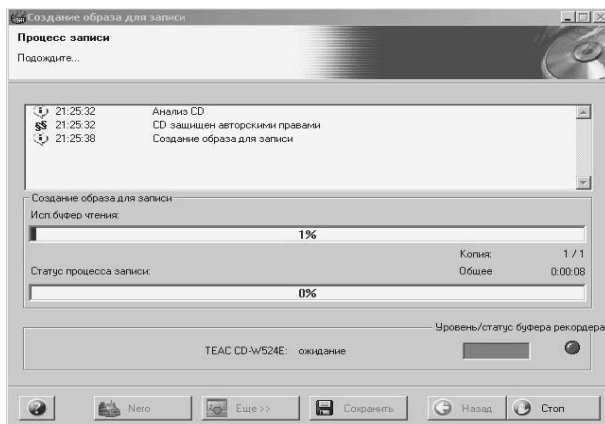


Рисунок 2.38 – Вікно підготовки записуваних даних на диск



процесу

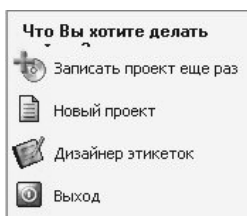


Рисунок 2.40 – Фрагмент вікна з питаннями про продовження роботи

## Питання й завдання для самостійного виконання

1. Що таке комп'ютерний вірус і способи його поширення?
2. Які різновиди вірусів Ви знаєте?
3. Охарактеризуйте різновиди антивірусних програм і приведіть приклади.
4. Виконайте перевірку двох дискет за допомогою антивірусної програми.
5. Виконайте антивірусну перевірку диска **D:** .
6. Які сервісні програми по обслуговуванню дисків Ви знаєте?
7. Відформатуйте дискету.
8. Виконайте перевірку диска **D:** .
9. Виконайте дефрагментацію заповненої робочої дискети.
10. Які файли необхідно періодично видаляти при чищенні дисків?
11. Призначення архівів і основні принципи їхнього створення?
12. Дайте характеристику відомих Вам програм – архіваторів.

13. Створіть на диску **D:** папку **Архіви** і помістіть в неї архіви усіх файлів з розширенням **.doc** із папки **Мої документи**, створені за допомогою архіватора *WinRAR* і *WinZIP*. Порівняйте розміри отриманих архівів і вхідних файлів.
14. Створіть у папці **Мої документи** архів, що саморозвертається, самого *архіватора WinRAR*.
15. Чим відрізняються архіви, що саморозвертаються і звичайні архіви?
16. Розгорніть rar-архів із папки **Архіви** на диску **D:** у папці **Навчальна**, а zip-архів на диску **C:** у папці **Тренажер**.
17. Для чого використовуються накопичувачі на оптичних дисках? З якими видами дисків Ви знайомі?
18. Створіть архів усіх документів, що Ви створили за останній рік і запишіть його на компакт-диск.

## Література

1. Андердал Б. Самоучитель Windows 98 – Спб: Питер, 1998 – 368с.
2. Войтюшенко Н.М., Остапець А.І. Комп'ютерна техніка і програмування. Частина 1. Основи підготовки користувача ПК: Навчальний посібник по базовій підготовці для студентів денної і заочної форм навчання. – Донецьк, ДонДует, 2001 – 150 с.
3. Глушаков С.В. Создание и запись CD/ С.В. Глушаков, В.Г.Мачула, Т.С. Хачиров. – Харьков: Фолио, 2005. – 391 с.
4. Информатика: Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів/ За ред. О.І. Пушкаря. – К.: Видавничий центр “Академія”, 2003. – 704 с.
5. Компьютерная документация по программам Dr. Web, AVP, WinRAR, WinZIP, Windows.
6. Компьютеры, сети, Интернет. Энциклопедия./Новиков Ю и др. – СПб.: Питер, 2002. – 928 с.
7. Основы современных компьютерных технологий: Учебное пособие/ под ред. проф. Хомоненко А.Д. – СПб.: КОРОНА принт, 1998 – 448с.
8. Шафран Ю. Информационные технологии: В 2ч. Ч. 2: Офисная технология и информационные системы. – М.: Лаборатория базовых знаний, 2000 – 336с.

## Розділ 3. Системи обробки текстів

Кожна людина у повсякденному житті зіштовхується з документами і різною текстовою інформацією. Тому навчання прикладним програмам природно почати з програм обробки текстів – текстових редакторів, систем розпізнання текстів і перекладачів.

### 3.1 Текстовий редактор WORD

Ефективність застосування комп'ютерів для підготовки текстів привели до створення безлічі прикладних програм для обробки документів. Такі програми називаються редакторами текстів. Усі редактори текстів можна розділити на наступні групи:

- Редактори текстів програм розраховані на редагування програм на тій чи іншій мові програмування, вони убудовані в систему програмування на деякій мові. Прикладом служать редактори, убудовані в систему програмування *Turbo C*, *Turbo Pascal* і т.д
- Текстові редактори для документів простої структури – програмний засіб для введення й модифікації текстових файлів або текстових документів простої структури, що не мають поділу на параграфи, глави і т.п. Приклади програм – *Блокнот*, *WordPad*, *MultiEdit*.
- Текстові редактори для документів складної структури – орієнтовані на роботу з текстами, що складаються з розділів, глав, параграфів із складним оформленням. Серед найбільш розповсюджених у світі редакторів виділяються *Microsoft Word*, *Word Perfect*, *WordStar*.
- Видавничі системи – використовуються для підготовки рекламних буклетів, оформлення журналів і книг. Вони дозволяють створювати складні документи найвищої якості. Прикладами таких систем є *Aldus PageMaker*, *Venture Publisher*.
- Редактори наукових документів – спеціальні редактори для документів з інтенсивним використанням складних математичних, хімічних формул, спеціальних символів, матриць, складних діаграм. Представниками таких редакторів є *ChiWriter*, *MathOr*, *MathWord*, *TEX*.




Текстовий редактор *Microsoft Word* призначений для створення й обробки документів складної структури. Крім тексту документи можуть включати малюнки, формули, таблиці, діаграми. Частина з них може включатися в текст безпосередньо, частина – тільки через зв'язки з динамічною реакцією на зміни.

#### Завантаження Microsoft Word:

кнопка **Пуск** → меню **Програми** → підміню **Microsoft Word**.

На екрані з'явиться вікно Microsoft Word з усіма стандартними елементами первинного вікна. Під рядком меню з'являються інструментальні кнопки панелей інструментів, що відповідають операціям, які використовуються найчастіше. Інструментальні кнопки згруповані в панелі інструментів по функціональній ознаці. Виставити потрібні панелі і прибрати непотрібні можна через меню **Вид**, команда **Панелі інструментов** позначити (зняти позначку) відповідні прапорці. Аналогічного результату можна досягти, клацнувши правою кнопкою миші в області панелей інструментів і позначити (зняти позначку) відповідні прапорці.

Крім того, у меню **Вид** зосереджені всі команди, що визначають зовнішній вигляд екрана текстового редактора й тексту, що на ньому розташовується. Перед початком роботи рекомендується виконати такі підготовчі операції:

- перейти в режим **Разметка страницы** (меню **Вид** → команда **Разметка страницы**);
- установити масштаб 75% або 100% (меню **Вид** → команда **Масштаб** або кнопка  на панелі інструментів **Форматирование**);
- вибрати шрифт і розмір для набору тексту (меню **Формат** → команда **Шрифт** → у полі списку **Шрифт** вибрати шрифт **Times New Roman** → у полі списку **Размер** вибрати **14** пунктів → **ОК** або кнопки  і  на панелі інструментів **Форматирование**); виставити лінійку (меню **Вид** → команда **Линейка**).

*Лінійка Word* – це не пасивна прикраса екрана над і ліворуч від документа, а багатофункціональний інструмент, що дозволяє ефективно виконувати різноманітні операції форматування абзаців.

Для контролю тексту на випадкові помилки можна установити перевірку правопису:



м. Сервис → к. Параметры → вкладка Правописание → в розділі Орфография виставити прапорець Автоматически проверяют орфографию → у розділі Грамматика → виставити прапорці Автоматически проверяют грамматику, Также проверяют орфографию → ОК.

### 3.1.1 Уведення, редагування і ручне форматування тексту в Microsoft Word

Текст у *Word* являє собою послідовність абзаців, кожний абзац являє собою послідовність слів, кожне слово являє собою послідовність букв (символів). Якщо є необхідність почати текст усередині абзацу з нового рядка або зсунути його в новому рядку щодо початку вправо, варто використовувати комбінацію клавіш **[Shift+Enter]** (недрукований символ ↵).

Для редагування тексту використовуються клавіші **[Insert]** – перемикач режимів вставки/заміни, **[Delete]** – стирає символ після курсору, **[Backspace]** – стирає символ перед курсором.

Щоб розірвати рядки, необхідно поставити курсор (клацнути мишкою) на місце розірвання і натиснути клавішу **[Enter]**. Щоб об'єднати рядки – варто поставити курсор після першого для об'єднання рядка і натиснути клавішу **[Delete]**.

Приклад 3.1. Наберіть наступний текст:

Як почати роботу, не читаючи посібників. Якщо ви віддасте перевагу вивченню нових програм, працюючи з ними, запустіть Word і озирніться. Більшість необхідних команд і інструментів легко знайти на панелях інструментів Стандартная й Форматирование або в меню *Word*. Якщо під час роботи буде потрібна допомога, викликайте помічника, уведіть питання і натисніть кнопку Знайти.

Зберегти набраний текст можна такими засобами:

1. Меню **Файл** → команда **Сохранить как** → у діалоговому вікні, що відкрилося, зазначити потрібну папку (**Мої документи**) і ім'я файла (*Навчальний текст*) → кнопка **Сохранить**. Команда **Сохранить как** завжди запитує ім'я файлу.

2. Меню **Файл** → команда **Сохранить**. Для тексту, набраного вперше, результат буде аналогічний першому пункту. Для відкритого файлу – зберігання піде без питань по старому маршруту.
3. Кнопка **Сохранить** панелі інструментів **Стандартная**, що відповідає команді **Сохранить**.

Відкрити файл із текстом для подальшої роботи можна за допомогою:

1. Меню **Файл** → команда **Открыть** → зазначити в діалоговому вікні, що відкриється, папку й ім'я файлу → кнопка **Открыть**.
2. Кнопка **Открыть** панелі інструментів **Стандартная**, що відповідає команді **Открыть**.
3. Меню **Файл** → дивитися в списку останніх оброблюваних текстів під списком команд.

Текстовий редактор Microsoft Word дозволяє друкувати документ в остаточному вигляді. Ця можливість забезпечується наявністю режиму *попереднього перегляду*, у якому відображується, як буде виглядати документ після друку. У випадку незадовільного результату користувач може продовжити форматування документа.

Перехід у режим попереднього перегляду: меню **Файл** → команда **Предварительный просмотр** або відповідна кнопка на панелі інструментів **Стандартная**. Вихід із режиму: клацнути по кнопці **Закрыть** на панелі інструментів.

Друк документів:

1. Інструментальна кнопка **Печать** у вікні попереднього перегляду.
2. Інструментальна кнопка **Печать** на панелі інструментів **Стандартная**.
3. Меню **Файл** → команда **Печать** → відзначити потрібний індикатор вибору і настроїти прапорці → **ОК**.

Останній варіант універсальний і припускає можливість друкування як усього тексту, так і окремих сторінок або виділених фрагментів тексту в потрібній кількості примірників.

Звичайно після набору тексту виникає потреба у форматуванні (зміні) тексту або його зовнішнього вигляду. Щоб *Word* знав, до якого фрагмента тексту відносяться дії користувача, необхідно виділити таку ділянку. Методи виділення тексту зведемо в таблицю 3.1. При

виділенні будемо використовувати поняття поля виділення. *Поле виділення* розташовано лівіше тексту, покажчик миші в ньому змінює напрямок і показує на текст ➤.

Таблиця 3.1 – Виділення елементів тексту.

Що виділяти	Як виділяти
Рядок	Поставити покажчик миші в поле виділення навпроти потрібного рядка і клацнути лівою кнопкою. Рядок буде виділено чорним кольором.
Декілька рядків	Поставити покажчик миші в поле виділення навпроти першого (останнього) рядка, при клацанні затиснути ліву кнопку і тягти до кінця діапазону, потім відпустити кнопку. Рядки будуть виділені чорним кольором.
Абзац	Поставити покажчик миші в поле виділення навпроти потрібного абзацу і виконати подвійне клацання лівою кнопкою. Абзац буде виділений чорним кольором.
Декілька абзаців	Поставити покажчик миші в поле виділення навпроти першого (останнього) абзацу, при подвійному клацанні затиснути ліву кнопку і тягти уздовж потрібної кількості абзаців. Абзаци будуть виділені чорним кольором.
Весь документ	Клацнути в поле виділення при натиснутій клавіші <b>[Ctrl]</b> .
Слово	Подвійне клацання по слову.
Словосполучення	Покажчик миші поставити на початок словосполучення, затиснути ліву кнопку і тягти до кінця словосполучення.

Зняти виділення – клацнути в будь-якому місці тексту.

Приклад 3.2. Відформатуйте введений у Прикладі 3.1 текст наступним чином:

1) перший рядок відокремити від тексту, змінити тип шрифту на Arial, розмір 18, вид шрифту – напвіжирний, отриману назву тексту центрувати й відокремити від іншого тексту порожнім рядком;

Виконання: клацнути мишкою після першого речення; натиснути клавішу **[Enter]** два рази; виділити перше речення – покажчик мишки поставити в поле виділення навпроти речення і клацнути лівою кнопкою; за допомогою інструментальної кнопки, установимо тип шрифту Arial (рисунок 3.1),

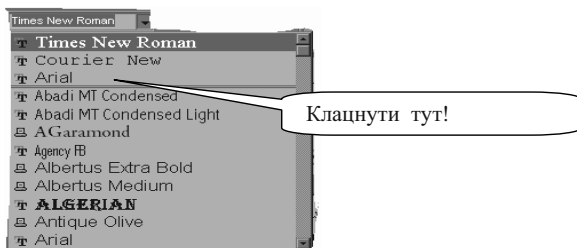


Рисунок 3.1 – Список кнопки установки типу шрифту, що випадає, панелі інструментів **Стандартная**

2) відформатувати абзац тексту: установити ліве поле 1 см, праве поле 1 см, новий рядок – 1.25 см, установити полуторний міжрядковий інтервал, текст вирівняти по ширині;

**Виконання:** виділити текст, що залишився, як один абзац – покажчик миші поставити в поле виділення навпроти будь-якого рядка абзацу і виконати подвійне клацання. Потім меню **Формат** → команда **Абзац** → установити параметри, як зазначено на рисунку 3.2.

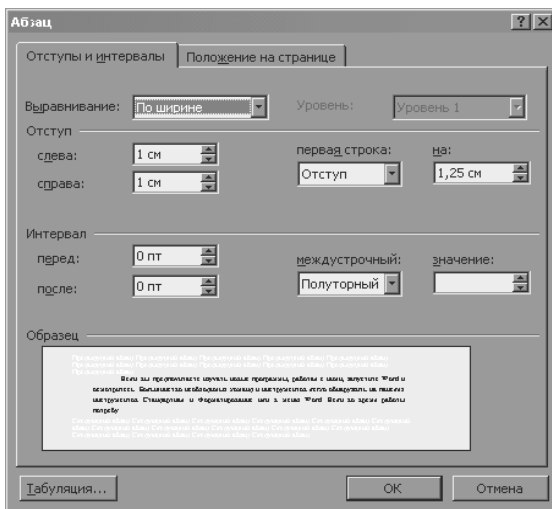


Рисунок 3.2 – Діалогове вікно команди **Абзац**

3) перший рядок тексту зробити курсивом, другий – підкресленим, третій – жирним підкресленим курсивом;

Виконання: виділити перший рядок і клацнути по кнопці **К** , виділити другий рядок і клацнути по кнопці **Ч** , виділити третій рядок і послідовно клацнути по кнопках **Ж К Ч** .

4) зняти з третього рядка жирність;

Виконання: виділити третій рядок абзацу і віджати кнопку **Ж** ;

5) слова тексту, що відповідають назвам, виділити напівжирним шрифтом;

Виконання: виділити слово Word подвійним клацанням і клацнути по кнопці **Ж** , аналогічні дії виконати зі словами Стандартна, Форматування і Знайти.

У результаті повинен утворитися текст наступного виду:

## **Як почати роботу, не читаючи посібників.**

*Якщо ви віддаєте перевагу вивченню нових програм, працюючи з ними, запустіть Word і озирніться. Більшість необхідних команд і інструментів легко знайти на панелях інструментів Стандартная й Форматирование або в меню Word. Якщо під час роботи буде потрібна допомога, викликайте помічника, уведіть питання і натисніть кнопку **Найти**.*

6) текст зберегти – м. **Файл** → к. **Сохранить** або кнопка **Сохранить**.

### **3.1.2 Графічні об'єкти в текстах**

Текстовий документ виглядає більш наочним і привабливішим, якщо в ньому присутні відповідні ілюстрації. Мова йде насамперед про розміщення на сторінці й налаштуванні параметрів відображення різноманітних графічних об'єктів. У Microsoft Office застосовуються графічні об'єкти різноманітних типів, що мають багато загального, але є і деякі розходження. Це наступні об'єкти:

**Малюнок**. Цей термін застосовується для позначення графічних об'єктів двох типів: по-перше, зображень (переважно растрових), що імпортовані з файлів у графічному форматі, і, по-друге, зображень, створених засобами малювання в самому Word.

**Картинка.** Звичайно так називають графічний об'єкт, що входить до складу колекції готових графічних об'єктів, що поставляються разом із Microsoft Office – **Microsoft Clip Gallery**.

**Фігурний текст.** Графічний об'єкт, створений додатком *WordArt*.

**Діаграма.** Графічний об'єкт, що будується по числовим даним таблиць. До складу Microsoft Office входить декілька різноманітних засобів побудови діаграм.

**Напис.** Спеціальний засіб, що дозволяє маніпулювати з будь-яким об'єктом, як із графічним.

Зазначені графічні об'єкти можуть створюватися в самому редакторі *Word* або розміщатися в тексті за допомогою допоміжних додатків, що викликаються через меню **Вставка** → **команда Об'єкт**.

Варто мати на увазі, що *Word* орієнтується на роботу в режимі WYSIWYG (What You See Is What You Get), тобто: те, що ви бачите на екрані в процесі роботи, відповідає тому, що ви одержите у виді твердої копії на папері в якості остаточного результату. При роботі з *Word* це дійсно так, але з однією обмовкою: відповідність підтримується в межах можливостей апаратури (наприклад, кольоровий або монохромний принтер).

Одним із найпоширеніших прийомів побудови складних зображень, наприклад, рекламних матеріалів, є накладення декількох шарів зображення один на одній. Кожна сторінка документа *Word* уже складається з трьох шарів: основний шар, або шар тексту, шар під текстом і шар над текстом. У шар під текстом природно помістити те, що звичайно називають фоном, у шар тексту поміщається основний текст, а в шар над текстом звичайно поміщають ті елементи, що потрібно винести на передній план. Крім того, у кожному шарі об'єкти можуть перекривати один одного і можна управляти відносним порядком відображення об'єктів на екрані. Таким чином, у *Word* є всі необхідні засоби для побудови багат шарових зображень.



**Вставка готових картинок.** Найбільше простий засіб проілюструвати текст – уставити готову картинку з колекції. Необхідно поставити курсор на місце вставки малюнка, потім:

меню **Вставка** → команда **Рисунок** → пункт **Картинки** → вибрати потрібну категорію малюнків, а в ній малюнок → кнопка **Вставити**.

Малюнок у тексті можна переміщати, змінювати його розміри. Для цього малюнок треба виділити (клацнути по ньому лівою кнопкою миші). Для переміщення покажчик миші треба помістити усередину малюнка і тягти останній у потрібне місце. Для зміни розміру варто помістити покажчик миші на один із маркерів рамки виділення (одержати ↔) і тягти в потрібний бік.

Можна використовувати власні малюнки для вставки в текст. Малювання виконується в графічних редакторах, наприклад, у *Paint*, *Corel Draw*, *3D Studio* і ін. Наприклад, створений у графічному редакторі *Paint* малюнок зберігається у файлі *Мій малюнок* у папці **Мої документи**. Щоб уставити цей малюнок у текст, необхідно зазначити курсором місце вставки, а потім виконати наступне:

меню **Вставка** → команда **Рисунок** → пункт **Из файла** → зазначити у діалоговому вікні, що відкрилося, папку **Мої документи** й ім'я файла *Мій малюнок* із малюнком → **ОК**.



Якщо потрібно вставити тільки фрагмент малюнка, то малюнок варто завантажити в графічний редактор *Paint*, виділити цікавлячий фрагмент за допомогою одного з інструментів  і скопіювати його в буфер обміну (меню **Правка** → команда **Копировать**). Закрити графічний редактор, зазначити в тексті місце вставки і клацнути по кнопці **Вставить** .

Приклад 3.3. Уставити в текст Приклада 3.2 готову картинку з колекції, малюнок із файлу з ім'ям *Мій малюнок* і фрагмент із цього файлу.

Виконання: 1) клацнути мишкою по порожньому рядку між заголовком і текстом, меню **Вставка** → команда **Рисунок** → пункт **Картинки** → вибрати потрібну категорію коміксів, а в ній клацнути по малюнку з комп'ютером → кнопка **Вставить**. Малюнок помістити по центру.

Зауваження – Якщо не вдається переміщати малюнок, то варто клацнути по малюнку правою кнопкою миші, у контекстному меню вибрати команду **Формат рисунка** → у діалоговому вікні вкладка **Положение** → клацнути по малюнку з написом **сверху и снизу** (у Office 2000 спочатку кнопка **Дополнительно**) → **ОК**. Після цього малюнок буде переміщатися.

2) Клацнути наприкінці тексту, натиснути декілька разів клавішу [Enter], (щоб мати можливість уводити надалі текст після малюнка) клацнути мишкою по першому рядку після тексту, а потім меню **Вставка** → команда **Рисунок** → пункт **Из файла** ( зазначити папку й ім'я файлу у діалоговому вікні, що відкрилося) → кнопка **Вставка**.

3) Завантажити графічний редактор *Paint*, відкрити в ньому файл *Мій малюнок*, клацнути по інструменту , поставити покажчик миші у верхній лівий кут прямокутника, який можна описати навколо шлюпочки, затиснути ліву кнопку миші і намалювати такий прямокутник, потім меню **Правка** → команда **Копировать**. Закрити графічний редактор. Перейти в текст і клацнути наприкінці тексту. Далі клацнути по кнопці **Вставить**  панелі інструментів **Стандартная**. Малюнок і фрагмент із нього поставити поруч.

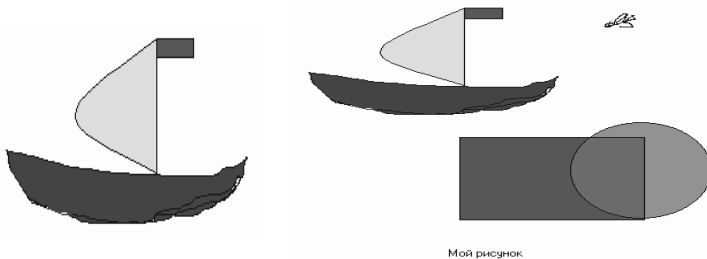
Повинен утворитися текст наступного виду:

## Як почати роботу, не читаючи посібників.




*Якщо ви віддасте перевагу вивченню нових програм, працюючи з ними, запустіть Word і озирніться. Більшість необхідних команд і інструментів легко знайти на панелях інструментів **Стандартная** й **Форматирование** або в меню **Word**. Якщо під час роботи буде потрібна допомога, викликайте помічника, уведіть питання і натисніть кнопку **Найти**.*





Мой рисунок

**Об'єкт WordArt.** Якщо користувач хоче представити текст у якомусь нестандартному (фігурному) вигляді (вигнутому, з різними розмірами і т.п.), то варто використовувати додаток *WordArt*. Щоб активізувати *WordArt*, необхідно уставити відповідний об'єкт:

м. **Вставка** → к. **Объект** → вибрати зі списку **Microsoft WordArt** → **ОК** або клацнути по кнопці  панелі **Рисование** → клацанням миші вибрати форму представлення (рисунок 3.3) текстового фрагмента → **ОК** → у наступне вікно внести сам текст, до якого слід застосувати стиль напису → **ОК**.

**Примітка** – Якщо текст набраний заздалегідь, то достатньо виділити потрібний фрагмент, щоб він з'явився у вікні

Вибраний в прикладі 3.4

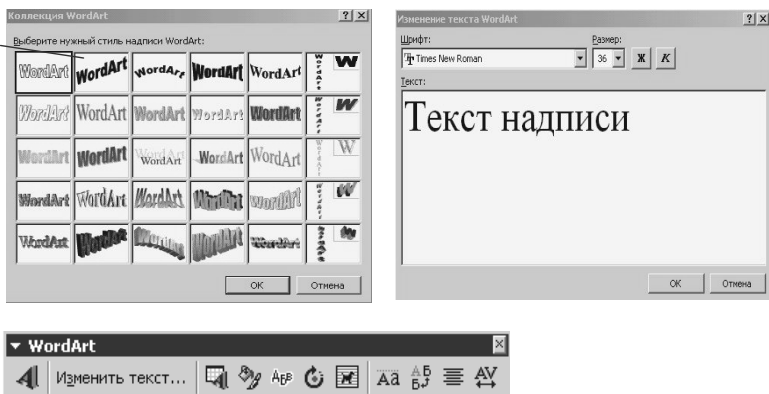




Рисунок 3.3 – Вікна програми та панель інструментів WordArt

Після появи фігурного тексту, на дисплеї з'явиться панель інструментів **WordArt**, що є основним засобом при роботі з даним графічним

об'єктом (для Office 2000 – малюнок 3.3). З її допомогою можна змінювати форму, розташування, вирівнювання, колір, відстань між буквами у фігурному тексті. Для того щоб повернутися до звичних інструментів *Word*, досить клацнути мишкою в будь-якому іншому місці документа. Панель знову з'явиться при виділенні фігурного тексту.

## Як почати роботу, не читаючи посібників.

Приклад 3.4. Заголовок тексту продублювати в об'ємному, що йде в перспективу, вигляді.

Виконання: Виділити заголовок тексту. Курсор залишити перед текстом. Меню **Вставка** → к. **Об'єкт** → вибрати **Microsoft WordArt** → **ОК** → у вікні **Коллекция WordArt** вибрати потрібний зразок (рисунок 3.3) → **ОК** → у вікні **Изменение текста WordArt** при необхідності відформатувати тип, вид, розмір шрифту → **ОК**. Отриманий графічний об'єкт можна переміщати та змінювати як звичайний малюнок. Крім того, можна продовжити форматування фігурного тексту, використовуючи інструменти панелі **Рисование**, наприклад, кнопки **Тень**  и **Объём** .

Малювання власними засобами Word. Панель інструментів **Рисование** містить у собі звичайні засоби, що є в будь-якому графічному редакторі. Якщо потрібно намалювати вручну дійсно складну картинку, краще це зробити в спеціалізованому графічному редакторі, наприклад, *Paint* або *CorelDRAW!* уставити, а потім уставити отриманий малюнок як об'єкт у документ *Word*. Але в простих випадках засобів малювання *Word* достатньо. При цьому засоби малювання *Word* добре узгоджені з "текстовою" частиною пакета, що не завжди вірно для графічних редакторів. Можна показати це на прикладі **винесень** (квадратики з пояснюючими написами на багатьох малюнках посібника).

Для початку потрібно створити самі винесення.

1. Клацніть на панелі інструментів **Рисование** по кнопці **Автофігури** → пункт **Выноски** → клацнути по потрібній формі виноски.

2. Потім перемістіть покажчик миші (який тепер має форму маленького хрестика) у ту точку, куди повинна вказувати стрілка винесення, і, утримуючи натиснутою ліву кнопку миші, протягніть рамку майбутнього винесення приблизно в те місце, де вона повинна бути. Тепер потрібно помістити курсор вводу в рамку винесення, увести текст, призначивши бажаний шрифт і розмір – усе як при звичайній роботі з текстом. Розмір і положення рамки й стрілки винесення можна змінювати: для цього треба виділити рамку винесення клацанням і маніпулювати відповідними маркерами за допомогою миші.

3. Установити формат винесень. Щоб зробити це відразу для усіх винесень, варто виділити їх по черзі клацанням миші при натиснутій клавіші **[Shift]**.

Меню **Формат** → команда **Автофігури...** → вкладка **Цвета и линии** → установити колір заливки, товщину ліній, стрілки; → вкладка **Положение** → клацнути по малюнку **перед текстом** → **ОК**

Установити параметри можна і для кожного винесення окремо, але, звичайно, зручніше встановити загальні параметри відразу для всіх графічних об'єктів. Винесення є окремим випадком написів. **Нанус** – це графічний об'єкт, що є прямокутним контейнером для будь-яких інших графічних і текстових об'єктів. У зв'язаних написах текст автоматично перетікає з одного напису в другий для його оптимального заповнення (див. довідку Windows).

Використання *математичних формул* у (друкарських) документах є скоріше правилом, ніж винятком. Тому розроблювачі *Word* приділили цьому питанню відповідну увагу і включили до складу пакета окремий додаток для набору математичних формул.

Наприклад, рівняння:  $(2 \cdot x^2 + 3) \cdot (1 - \sin(x)) = \ln(x)$   
цілком можна переписати і звичайними буквами:

$$(2x^2+3)(1-\sin(x)) = \ln(x)$$

Зауважимо, до речі, що показник ступеня тут піднятий за допомогою меню **Формат** → команда **Шрифт** → вкладка **Шрифт** → установити прапорець **верхній індекс** → **ОК**. Аналогічно можна записувати підрядкові (нижні) індекси

Більш складні формули в *Word* – це особливі об'єкти, що вводяться, редагуються і форматуються за допомогою додатка **Редактор формул (Microsoft Equation)**. Це робиться таким чином:

Меню **Вставка** → команда **Об'єкт** → вибрати **Microsoft Equation 3.0** → **ОК**

У результаті у вікні *Word* запускається додаток *Microsoft Equation* із своїм вікном для введення формули і своєю панеллю інструментів (рисунок 3.4).



Выпадающий список **Степени и индексы**


$$H = P_0 + \frac{P_1}{1+K} + \frac{P_2}{(1+K)^2} + \dots + \frac{P_T}{(1+K)^T} \qquad G = \int_1^{\infty} (\sin \beta^2 + \frac{r * n}{\sqrt{r+n}})$$

Рисунок 3.4 – Панель інструментів *Microsoft Equation 3.0* і формули, створені за її допомогою

Звичайні символи можна набирати на клавіатурі, інші вибираються за допомогою кнопок – списків, що випадають, панелі інструментів вікна **Формула**. Будь-яке клацання поза панеллю й вікном уведення формули закриває *Equation*. При наборі вищенаведеного рівняння використовувалася тільки кнопка індекси і ступені, за допомогою котрої перед набором ступеня 2 вказується, в якому виді повинен утворитися результат.

Подвійне клацання мишею по формулі в тексті викликає режим редагування й панель інструментів *Microsoft Equation* на екран, якщо формула спочатку була створена за допомогою даного об'єкта.

### 3.1.3. Оформлювальні засоби

**Межі й заливка.** Окремі фрагменти тексту можна виділяти **рамками** і змінювати їхній **колірний фон**. Для цього використовується меню **Формат** → команда **Границы и заливка** або відповідна кнопка панелі інструментів **Форматирование** для створення меж .

Приклад 3.5. Назву панелей інструментів у тексті нашого прикладу взяти в подвійну рамку і залити сірим кольором 35%.

Виконання: Виділити в тексті назву панелей інструментів потім: меню **Формат** → команда **Границы и заливка** → вставка **Граница** → вибрати в списку **Тип** подвійну лінію → вставка **Заливка** → вибрати сірий 35% → **ОК**.

Текст набуде вигляд, приведений нижче.

*Якщо ви віддасте перевагу вивченню нових програм, працюючи з ними, запустіть Word і озирніться. Більшість необхідних команд і інструментів легко знайти на панелях інструментів **Стандартная** й **Форматирование** або в меню **Word**. Якщо під час роботи буде потрібна допомога, викликайте помічника, уведіть питання і натисніть кнопку **Найти**.*

### Символи. Буквиця

Символи. У тексті, що набирається, можуть використовуватися *символи, котрих немає на клавіатурі*. Наприклад, символ № може бути вставлений за допомогою комбінації клавіш **[Shift+3]**.


Інші ж символи можуть бути відсутніми й у шрифті, який використовується. Їх можна вставити через меню **Вставка** → команда **Символ** → вкладка **Символнаприклад**, ти тип шрифту, наприклад **Symbol** → кнопка **Вставити** → кнопка **Закрити**. Саме так уставлявся символ →, потім його можна зберегти в буфері обміну або виділити й перетаскувати при натиснутій клавіші **[Ctrl]**.

Буквиця. Першу букву абзацу в тексті можна оформляти специфічним засобом. Для цього треба виділити цю букву, потім меню **Формат** → команда **Буквиця** → виділити один із варіантів параметра **Положение** → **ОК**.

## Колонки та таблиці

Дуже часто у документах, що створюються, використовуються **колонки**, табличний матеріал. Для такого розташування тексту можна скористатися:


- 1) табуляцією;
- 2) таблицями;
- 3) багато колонковим набором.

Табуляція дозволяє переміщати курсор у певну позицію екрана при натисканні на клавішу **Tab**. При відсутності будь-яких вказівок із боку користувача це буде пересування на задану по умовчання кількість сантиметрів вправо (м. **Формат** → к. **Табуляція** → поле **По умовчанию**). Для вказівки певного місця на аркуші можна скористатися лінійкою й кнопкою виду табуляції  на лівому краї горизонтальної лінійки. Клацаннями по кнопці можна встановити потрібний вид табуляції, а клацання по лінійці вкажуть, де повинні починатися колонки.

Більш точний результат можна одержати при використанні меню **Формат** → команда **Табуляція** → зазначити потрібні параметри → кнопка **Установить** → кнопка **ОК**



Приклад 3.6. Створити новий документ, у якому зробити шапку для ділового листа з назвами й адресами відправника й одержувача (користувач і його навчальний заклад). Текст відправника повинен розташовуватися до 8 см, а одержувача – після 8 см. .

### Виконання:

1. Для створення нового документа клацнути по кнопці  або меню **Файл** → команда **Создать** → вкладка **Общие** → вибрати **Обычный** або **Новый документ**, повинна бути включена селекторна кнопка **Документ** → **ОК**. З'явиться чистий лист. Для листа можна встановити параметри сторінки як у прикладі 3.1.
2. Клацнути мишкою по лінійці на діленні 8 см (повинен з'явитися табулятор).
3. Установити курсор на початок рядка і набрати частину тексту відправника (Студент 3 курсу ОФФ), потім натиснути клавішу [**Tab**] і набрати частину тексту одержувача. Аналогічно повторити для другого рядка. У результаті текст повинен бути поданий у такий спосіб:

Студент 3 курсу ОФФ  
ДонДУЕТ Іванов А.П.  
83129 р. Донецьк,  
вул.Шведова, д.34, кв.67

Донецький державні універ-  
ситет економіки і торгівлі  
83050, м.Донецьк, вул.Щорса, 31

Для створення таблиці використовується меню **Таблиця**, команда **Нарисовать таблицу**  або кнопка **Добавить** таблицю  на панелі інструментів **Стандартная**. При заповненні чарунок таблиці виконується автоматичний контроль ширини тексту й регулювання висоти чарунки. У таблицю можна добавляти нові рядки і стовпчики, видаляти непотрібні, об'єднувати й розділяти чарунки (меню **Таблиця**), очищати чарунку (клавіша **[Delete]**). Видалити частини таблиці або усю таблицю можна командою **Вырезать** або м. **Таблиця** → к. **Удалить**.

Приклад 3.7. Нижче створеного тексту за прикладом 5.6 додати таблицю такої форми:

Інформація про студентів .

П.І.Б.	Факультет	Курс	Домашня адреса


Назву граф зробити напівжирним шрифтом, відцентрувати, залити сірим кольором (35%). Заповнити три рядки таблиці


Після графі «Курс» уставити графу «Телефон», привести ширину граф у відповідність із шириною сторінки. Після назви граф уставити порожній рядок, об'єднати в ній чарунки і написати текст «Заочне відділення», тест відцентрувати.



Відсортувати рядки таблиці за алфавітом по графі П.І.Б.

Весь отриманий текст помістити в кінець документа, створеного в прикладах 3.1 – 3.5.

#### Виконання:


1) Клацнути нижче тексту, уведеного за прикладом 3.6 і ввести заголовок таблиці. Виконати центрівку назви – клацнути по кнопці , перейти на наступний рядок – клавіша **[Enter]**.


2) Клацнути по кнопці , виділити потрібну кількість рядків і стовпчиків, протягуючи над ними лівою кнопкою миші, і знов

клацнути. В отриману таблицю внести заголовки граф. Виділити рядок, в якому розміщений заголовок таблиці, центрувати в ньому текст , зробити його напівжирним , потім меню **Формат** → команда **Границы и заливка** → вкладка **Заливка** → вибрати колір → **ОК**.

Заповнити три рядки довільними даними.

3) Для вставки стовпчика варто підвести покажчик миші зверху до верхньої лінії графи «Домашня адреса» так, щоб утворилася стрілка ↓, і клацнути лівою кнопкою миші (графа буде виділена чорним кольором).

Кнопка **Добавить таблицу** змінить зовнішній вигляд на  і функцію на **Добавить столбцы**. Варто клацнути по цій кнопці й у таблиці з'явиться нова графа. Увести назву «Телефон». У Office 2000 програма контролює загальну ширину таблиці стосовно ширини сторінки тексту. У Office 97 таблиця може вийти за межі сторінки, і її необхідно відформатувати. У будь-якому випадку при клацанні по таблиці на лінійках з'являються межі граф і рядків, які можна пересувати мишкою в потрібну сторону. Або можна виділити потрібні стовпчики й у меню **Таблица** вибрати команди **Вировнять ширину столбцов** або **Ширина и высота ячейки** (для Office 97) і **Автоподбор** або **Свойства таблицы** в Office 2000).

Виділити як звичайний текст перший рядок таблиці з прізвищем, при цьому кнопка **Добавить таблицу** змінить зовнішній вигляд на  і функцію на **Добавить строки**. При клацанні по кнопці в таблицю добавляється новий рядок такої ж структури як і виділений. Не знімаючи з нього виділення зробити:

меню **Таблица** → команда **Объединить ячейки**.

Зняти виділення й усередині об'єднаної чарунки написати «Заочне відділення», потім центрувати текст.

Виділити чарунки предметної частини таблиці → м. **Таблица** → к. **Сортировка** → уточнити параметри сортування (рисунок 3.5) → **ОК**.



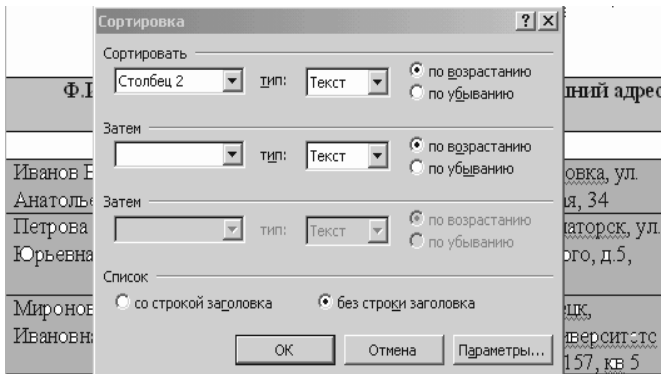


Рисунок 3.5 – Вікно параметрів сортування

У результаті всіх дій повинна вийти наступна таблиця (рисунок 3.6):

### Інформація про студентів.

П.І.Б.	Факультет	Курс	Телефон	Домашня адреса
Заочне відділення				
Іванов Валерій Анатолійович	ОФФ	1	-	м.Горлівка, вул. Зеленая, 34
Миронова Юлія Іванівна	ОФФ	1	-	м.Донецьк, вул.Університетська, д.157, кв 5
Петрова Світлана Юрійвна	ОФФ	1	3-16-29	м.Краматорськ, вул. Горького, д.5, кв.12

Рисунок 3.6 – Результат виконання приклада 3.7

Примітка – При необхідності перенесення частини таблиці на наступні сторінки для повторення заголовної частини варто виділити заголовки таблиці (бо він може бути значно складнішим в інших таблицях) → вибрати м. **Таблиця** → к. **Заголовок**. Словосполучення Продовження таблиці можна внести у верхній колонтитул, попередньо поставивши розриви на потрібних сторінках (див. далі).

Таблиця *Word* багато в чому нагадує електронні таблиці. Кожна чарунка такої таблиці має свою адресу, що складається з імені стовпця (англійські букви A, B, C, D, E, F і ін.) і номера рядка, на перетинанні

яких вона знаходиться (наприклад, A3, F10 і ін.). І також, як електронна, звичайна таблиця дозволяє виконувати різні математичні розрахунки над даними, що знаходяться в її чарунках. Для цього необхідно установити курсор у чарунку для розміщення результату, а потім:

П.м. **Таблиця** → к. **Формула** → у поле **Формула** увести формулу, по якій повинен виконуватися розрахунок. Це може бути самостійно створена формула або функція, обрана зі списку поля **Вставити функцію** (рисунок 3.7)

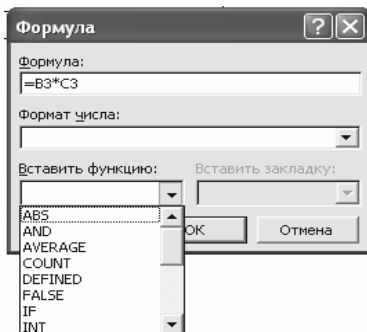


Рисунок 3.7 – Діалогове вікно для створення формул у таблицях *Word*

При самостійному створенні формул у них необхідно вказувати адреси чарунок з вхідними даними й знаки математичних операцій (+, -, \*, /). Усі формули повинні починатися зі знака « = ».

Найчастіше використовувані функції: **SUM** – сума, **AVERAGE** – середнє арифметичне, **COUNT** – кількість значень, **MIN** – мінімум, **MAX** – максимум і ін. Для вказівки послідовності чарунок із вхідними даними у функціях можна використовувати терміни: **ABOVE** (Вище), **BELOW** (Нижче), **LEFT** (Ліворуч), **RIGHT** (Праворуч). Наприклад: **=SUM(ABOVE)** – розрахунок суми значень всіх чарунок, розташованих вище чарунки для результату; **=SUM(LEFT)** – розрахунок суми значень всіх чарунок, розташованих ліворуч від чарунки для результату; **=AVERAGE(A5:A10)** – розрахунок середнього арифметичного чисел, що знаходяться в чарунках з **A5** по **A10**.

*Word* уставляє результати розрахунків у чарунки як **поле**. Це означає, що після внесення змін у чарунки з даними таблиці повторний розрахунок автоматично не виконується. Для відновлення результатів розрахунків необхідно виділити чарунку з результатом або усю таблицю і

натиснути клавішу [F9]. Для одночасного виведення на екран усіх створених формул необхідно виділити таблицю і натиснути одночасно клавіші [Alt + F9]. Повторне натискання цих клавіш скасовує режим виведення формул.

Для створення багато колонкового тексту необхідно виділити текст, який варто розбити на стовпчики, потім кнопка **Колонки** ■ на панелі інструментів **Стандартная** або: меню **Формат** → команда **Колонки** → указати кількість і ширину колонок → **ОК**.

Приклад 3.8. Текст «Як почати роботу, не читаючи посібників» представити у вигляді двох колонок з роздільником.

Виконання: Для розбивки на колонки варто виділити текст → м. **Формат** → к. **Колонки** → у полі **Число колонок** указати 2 → виставити прапорець **Разделитель** → **ОК**. Отриманий результат представлений на рисунку 3.8.

*Якщо ви віддаєте перевагу вивченню нових програм, працюючи з ними, запустіть Word і озірніться. Більшість необхідних команд і інструментів легко знайти на панелях інструментів **Стандартная***

*і **Форматирование** або в меню **Word**. Якщо під час роботи буде потрібна поміч, викличте помічника, уведіть запитання і натисніть кнопку **Найти***

Рисунок 3.8 – Приклад двох колонкового тексту

Якщо роздільник не потрібний, то можна скористатися інструментальною кнопкою **Столбцы** ■ і виділити мишкою потрібну кількість стовпчиків.

Примітка. Якщо текст невеличкий і не вдасться коректно розділити його на колонки, то попередньо варто клацнути на порожньому рядку до тексту, меню **Вставка** → команда **Разрыв** → установити селекторну кнопку **на текущей странице** → **ОК**. Аналогічні дії необхідно проробити і після тексту. Потім повторити операції по розбивці на колонки.

## Розділи

При оформленні документів, особливо складної структури, може виникнути потреба у використанні розділів. Розділи створюються, якщо змінюються:

- розміри друкованої сторінки;
- орієнтація друкованої сторінки;
- розміри полів або відступи колонтитулів;
- кількість стовпчиків тексту на сторінці;
- зміст колонтитулів друкованих сторінок;
- нумерація сторінок.

Новий розділ документа створюється:

м. **Вставка** → к. **Разрыв** → указати місце розриву → **ОК**.

Приклад 3.9. Перетворити отриманий у попередніх прикладах текст у такий спосіб (зовнішній вигляд на рисунку 3.9):

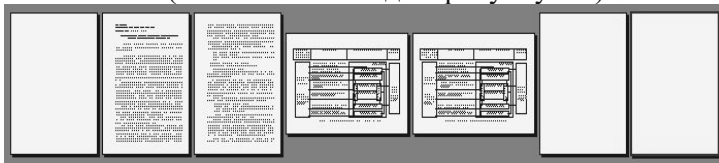


Рисунок 3.9 – Розташування розділів і тексту після виконання приклада 3.9

1) текстовий матеріал із малюнками залишити на листах книжкової орієнтації, перед ними вставити порожній лист;

2) усі таблиці розмістити в цьому ж тексті на листах альбомної орієнтації;

3) після листів альбомної орієнтації вставити два листи книжкової орієнтації.

Виконання:

1) Поставити курсор після текстового матеріалу → м. **Вставка** → к. **Разрыв** → указати селекторну кнопку **Со следующей страницы** → **ОК** → для нової сторінки установити параметри сторінки м. **Файл** → к. **Параметры страницы** → вкладка **Размер бумаги** → селекторна кнопка **Альбомная** → у полі **Применить** виставити **К текущему разделу** → **ОК**.

Порожній лист на початку тексту вставити самостійно.

- 2) Перемістити на альбомний лист усі необхідні таблиці.
- 3) Курсор поставити нижче таблиць і повторити операцію по вставці нового розділу з наступної сторінки (див. п. 1) приклада), змінити орієнтацію останнього листа на книжкову.

### 3.1.4 Засоби автоматизації вводу й заповнення документів

Текстовий редактор дозволяє робити ряд змін у тексті автоматично. Ці зміни діляться на три класи:

- Зміни, що робляться цілком автоматично, і не вимагають ніяких дій користувача. Такими є поява прописної букви на початку речення й виправлення орфографічної помилки.
- Зміни, що робляться по неявному «натяку» користувача. Такими є зміни стилю абзацу при повторному натисканні клавіші **[Enter]** і поява горизонтальної лінії замість трьох уведених дефісів.
- Зміни, що виконуються по явній указівці користувача. Таким є заміна букв «МО» на словосполучення «Microsoft Office» при натисненні клавіші **[F3]**.

Засоби *Word*, що виконують зміни цих трьох класів, називаються, відповідно, *Автозамена*, *Автоформат* (при вводі) і *Автотекст*. Конкретний набір змін, виконуваних цими засобами, визначається параметрами діалогового вікна **Автозамена**, яке відчиняється командою

меню **Сервіс** → команда **Автозамена...**

*Автозамена* використовується для виконання трьох основних функцій:

Автоматичне виправлення помилок при вводі тексту. Виправлення проводиться шляхом заміни слова в лівому стовпчику списку словом у правому стовпчику списку. У початковому стані список містить приблизно 250 елементів із російськими словами (і набагато більше елементів з англійськими словами). Для ефективного застосування цієї функції Автозаміни потрібно настроїти список, включивши в нього помилки, характерні для конкретного користувача.

Автоматична заміна при вводі визначених сполучень звичайних символів спеціальними символами. Наприклад, сполучення символів (r)

автоматично замінюється символом ®. У вхіднім списку утримується ледве більш дюжини таких замін.

Автоматична заміна визначених скорочень або позначень розгорнутими визначеннями, графікою і взагалі чим завгодно. Ця функція Автозаміни перетинається з основною функцією Автотексту.

Щоб поповнити список Автозаміни, потрібно ввести замінне буквосполучення в поле **Замінить**, слово або словосполучення, на яке воно замінюється, ввести в поле **На** і натиснути кнопку **Добавить**. Якщо перед відкриттям діалогового вікна **Автозаміна** виділити текст або інший об'єкт, то він відразу опиниться в полі **На**. Саме таким засобом у список Автозаміни включаються графічні об'єкти.

Добратися до цього опису можна за допомогою такої послідовності дій:

меню **Справка** → команда **Справка по Microsoft Word** → **Содержание** → папка **Форматирование** → папка **Использование автоформата** → вибрати **Автоматическое форматирование документа** → кнопка **Вывести** → читати текст у правому вікні.

У суті **Автотекст** виконує ту ж функцію, що й Автозаміна. Є список пар «ім'я – значення» і обидва засоби дозволяють замість імені підставляти його значення. Проте між цими засобами є деякі відмінності.

Автозаміна виконує підстановку автоматично, а Автотекст – при натисненні клавіші **[F3]** або **[Enter]**. Тому елементам Автотексту можна давати цілком довільні імена, у тому числі звичайні слова, без ризику небажаної підміни слів.

Автозаміна виконує підстановку під час уводу тексту, а Автотекст – ні. З цього слідує, що вставлений елемент Автотексту не піддається Автозаміні. Зокрема, якщо значення елемента Автотексту починається з малої літери й уставляється в початок речення, то автоматичне перетворення першої букви в прописну не відбудеться (що не зовсім зручно).

Елементи Автозаміни глобальні, а елементи Автотексту зберігаються у визначеному шаблоні, що дозволяє уникнути колізій між декількома користувачами одного примірника *Word*.

Вибір засобу залежить від конкретної задачі й смаків користувача.


У *Word* з'явився ще один корисний засіб прискорення уводу, тісно пов'язаний з Автотекстом, що називається **Автозавершение** (або

**Автозаповнення**). Суть складається в тому, що якщо цей засіб включений:

меню **Сервіс** → команда **Автозамена** → вкладка **Автотекст** → виставити прапорець **Автозаповнення для автотекста и дат**, – то після вводу перших чотирьох символів назви елемента Автотексту з'являється спливаюче підказування зі значенням елемента Автотексту, що дозволяє завершити ввід простим натисненням клавіші **[Enter]**. Крім того, Автозавершення пропонується ще і для назв місяців і днів тижня.

Ще один часто використовуваний засіб *Word* – **списки**. Списки *Word* бувають **маркіровані, нумеровані і багаторівневі**.

Фактично, **список** – це послідовність абзаців, відформатованих особливим способом, причому в початок кожного абзацу *Word* автоматично додає маркер для маркірованих списків або послідовний номер для нумерованих списків. Абзаци, що входять у список, називаються елементами списку. Як правило, елементи списку мають деякий відступ і виступ першого рядка.

Як завжди в Microsoft Office для простих і типових операцій передбачені найпростіші методи використання. Щоб створити нумерований або маркірований список, потрібно просто виділити абзаци списку і натиснути відповідну кнопку на панелі інструментів **Форматирование** . Крім того, можна скористатися командою меню для створення маркірованого списку:

Меню **Формат** → команда **Список** → вкладка **Маркированный список** → вибрати **Образец маркеров** → **ОК**

**Приклад 3.10.** Відкрити документ *Навчальний текст* і додати в третій розділ (листи книжкової орієнтації після альбомної орієнтації) наступне:

- 1) На початку даного параграфа надрукований текст із трьома абзацами у вигляді маркірованого списку, повторіть цей текст.
- 2) Увесь текст параграфа буває повторюваними назвами Автозаміна, Автотекст. Введіть у список функції Автозаміни відповідну пару: **замінити автоза на Автозаміна** й уведіть фрагмент тексту.
- 3) Збережіть текст із старим ім'ям.

## Стили абзаців і символів

Було б украй незручно вручну задавати форматування кожного символу і кожного абзацу у великому документі. По-перше, це досить утомливо, по-друге, чревате помилками. При ручному форматуванні легко пропустити якийсь фрагмент або випадково задати різні параметри для однорідного тексту. Якість форматування визначається тим, наскільки послідовно й одноманітно використовується форматування в документі.

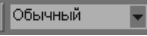
При правильній роботі, ручне форматування тексту в документі *Word* є рідкісним винятком. Як правило, форматування виконується за допомогою стилів.

**Стиль** – це сукупність параметрів форматування з унікальним ім'ям, призначених для багаторазового використання. Застосування стилю до деякого тексту означає, що для цього тексту встановлюються відразу всі параметри форматування, що входять у стиль. У *Word* розрізняються **стили символів** і **стили абзаців**.

**Стиль символів** може визначати параметри форматування, характерні для форматування символів: шрифт, його розмір і накреслення, ефекти (наприклад, підкреслення), інтервал між символами і т.д.

**Стиль абзацу** може визначати додатково до зазначених такі параметри форматування, як вирівнювання, відступи, позиції табуляції, інтервали між рядками і т.д.

Стиль абзацу може застосовуватися тільки до абзацу цілком, а стиль символів може застосовуватися до різних частин абзацу аж до окремих символів.

До будь-якого абзацу документа *Word* застосований деякий стиль. Ім'я цього стилю відображається в списку **Стиль**, що розкривається, на панелі інструментів **Форматирование** . За замовчуванням застосовується стиль "Обычный".

Щоб застосувати стиль до нового фрагмента тексту або до виділеної ділянки тексту, необхідно виконати наступні дії:

1. Меню **Формат** → команда **Стиль** → вибрати потрібний зі списку стилів → кнопка **Применить** (у цьому випадку доступний більш широкий список стилів).
2. Список стилів, що розкривається, на панелі інструментів **Форматирование** → вибрати потрібний зі списку стилів.



До того самого тексту може бути застосований стиль абзацу, стиль символів і ручне форматування, що задають різні значення для одного параметра форматування. У цьому випадку ручне форматування має перевагу перед стилем символів, а стиль символів має перевагу перед стилем абзацу.

Щоб довідатися, які параметри форматування і яким чином вони встановлені у визначеному місці документа, натисніть клавіші [**Shift + F1**] (при цьому покажчик миші прийме форму стрілки зі знаком питання) і клацніть лівою кнопкою миші в місці документа, що Вас цікавить. Щоб вийти з цього режиму, натисніть клавішу [**Esc**].

Приклад 3.11. У навчальному тексті, отриманому при виконанні попередніх прикладів, у кожному розділі внести заголовки глав і не менш двох параграфів. Усі заголовки відформатувати з використанням стилів **Заголовок 1** і **2** для наступного автоматичного формування змісту документа.

Кожному малюнкові в тексті дати назву з використанням автоматичної нумерації.

#### Виконання.

1. Внести для першого розділу наступні заголовки – **Глава 1. Як почати роботу, не читаючи посібників; 1.1. Уведення та редагування тексту; 1.2. Картинки.** Внести для другого розділу заголовки – **Глава 2. Колонки та таблиці; 2.1. Колонки; 2.2. Таблиці.** У третьому розділі дати заголовки – **Глава 3. Автоматизація введення та редагування тексту; 3.1. Автозаміна та автотекст; 3.2. Зміст та покажчики.**
2. Виділити назву глави – **Глава 1. Як почати роботу, не читаючи посібників** → у списку стилів, що випадає, панелі інструментів **Форматирование** вибрати **Заголовок 1**. Аналогічним образом відформатувати заголовки інших глав.
3. Виділити назву параграфа – **1.1. Уведення та редагування тексту** → у списку стилів, що розкривається, панелі інструментів **Форматирование** вибрати **Заголовок 2**. Аналогічним образом відформатувати заголовки інших параграфів.
4. Щоб вставити назву під уже створеним об'єктом або фрагментом тексту, треба його виділити (виділити малюнок із комп'ютером),

а потім м. **Вставка** → к. **Название** (рисунок 3.10) → указати постійну частину назви – **Рисунок** (або кнопка **Создать** для нестандартної назви) → указати місце положення назви (**под объектом**) → **OK** → додати текст назви малюнка.

Можна виконувати автоматичну нумерацію об'єктів, що вставляються, – рисунків, таблиць, формул:

м. **Вставка** → к. **Название** → кнопка **Автоназвание** → виставити прапорці напроти джерел створюваних рисунків, наприклад, **Microsoft Clip Gallery** → **OK**.

При уставці відповідних об'єктів автоматично буде з'являтися стандартна назва з поточним номером. Для прикладу уставте в текст готовий рисунок.

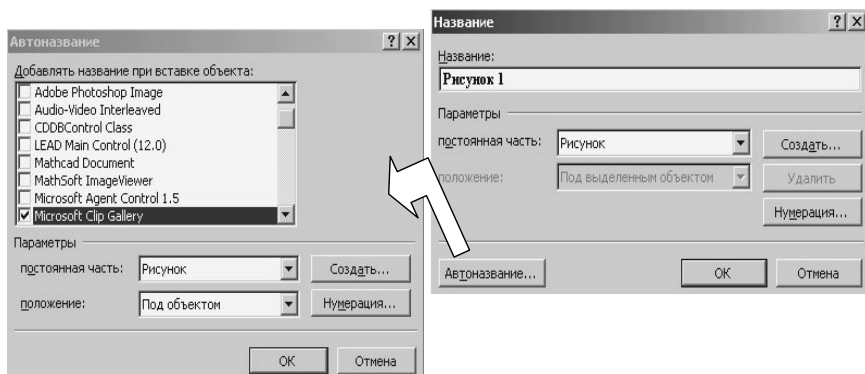


Рисунок 3.10 – Вікна для створення нумерованих назв до об'єктів

## Номера сторінок, колонтитули, виноски й примітки

При підготовці документів користувач достатньо часто зштовхується з багато сторінковим текстом. Проставити *номера сторінок* можна, використовуючи меню **Вставка** → команда **Номера страниц** → уточнити параметри діалогового вікна → **OK**.

Проставте в документі *Навчальний текст* номери сторінок угорі справа.




У верхнє й нижнє поля документа можна заносити текст, що повторюється, наприклад, про автора, назву документа і т.д. У цілому

**колонтитул** – це текст і/або малюнок (номер сторінки, дата друку документа, емблема організації, назва документа, ім'я файлу, прізвище автора і т.п.), що друкується насподі або вгорі кожної сторінки документа. У залежності від місця розташування (на верхньому або на нижньому полі сторінки) колонтитули бувають верхніми й нижніми. Для виконання цієї роботи виконується:

меню **Вид** → команда **Колонтитулы** → з'являється панель інструментів **Колонтитулы** (рисунок 3.10) і курсор встановлюється у верхній колонтитул → вводяться дані з клавіатури або за допомогою панелі інструментів → кнопка **Закрить** на панелі інструментів **Колонтитулы**.

Приклад 3.12. Відкрити файл *Навчальний текст* і вставити у верхній колонтитул найменування глави по центру на парних сторінках, своє прізвище, ім'я, по батькові на непарних сторінках, у нижній колонтитул – дату й час виконання приклада.

Виконання:

1. Меню **Файл** → під командами меню у списку файлів, що оброблялися останніми, вибрати клацанням миші «Навчальний текст».
2. Меню **Вид** → команда **Колонтитулы** → клацнути по кнопці **Параметры страницы** (рисунок 3.11) → виставити прапорець **Различать колонтитулы четных и нечетных страниц** → у полі **Начать раздел** виставити **С нечетной страницы** → **ОК**.
  - у верхньому колонтитулі на непарній сторінці набрати прізвище, ім'я і по батькові;
  - у верхньому колонтитулі на парній сторінці м. **Вставка** → к. **Перекрестная ссылка** (рисунок 3.12) → у списку **Тип ссылки** вибрати **Заголовок** → у списку **Для какого заголовка** вибрати **Як почати роботу, не читаючи посібників** → кнопка **Вставить** → кнопка **Отмена**.
    - переключитися у нижній колонтитул (кнопка ) → клацанням по кнопці  уставити поточну дату → увести 3 пробіли → увести час за допомогою кнопки  → кнопка **Закрить** панелі **Колонтитулы**.

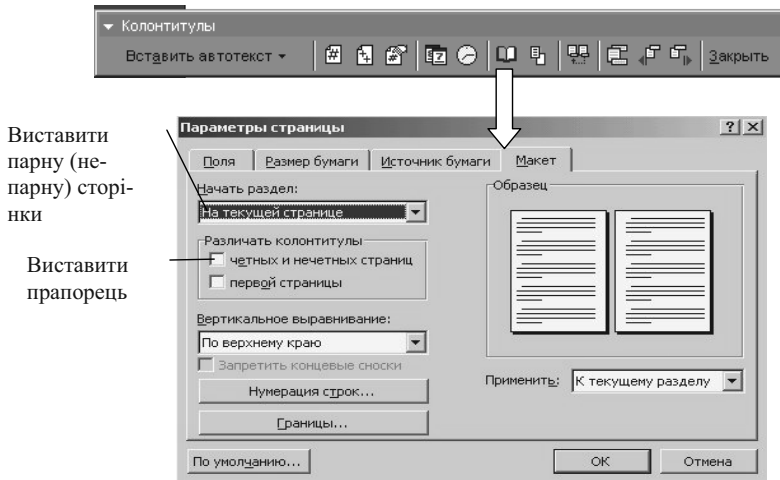


Рисунок 3.11 – Панель інструментів і вікно настройки виду колонтитулів

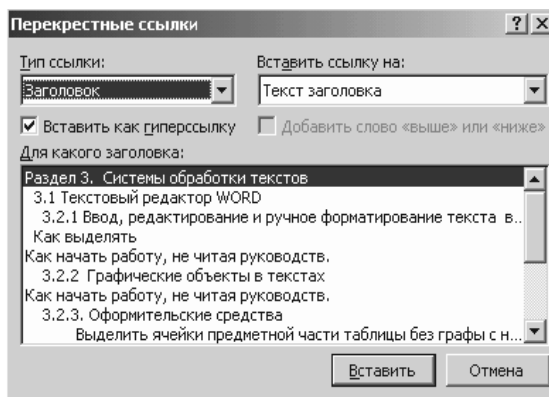


Рисунок 3.12 – Вікно настроювання автоматичного занесення заголовків у колонтитули

**Примітка 1.** При зміні назви або номера розділу в документі колонтитул буде автоматично оновлений при роздрукуванні документа. Для відновлення колонтитулу вручну треба його виділити і натиснути клавішу **F9**.

**Примітка 2.** Якщо в колонтитулі нового розділу повторюється назва попередньої глави, то при введенні нового колонтитулу варто простежити, щоб на панелі інструментів **Колонтитулы** кнопка **Как в предыдущем** була віджата!

### 3. Текст зберегти з колишнім ім'ям.

У тексті, який уводиться, іноді необхідно робити пояснення й уточнення. Для цього використовуються **примітки й виноски**.

**Примітку** можна написати просто текстом із словом примітка у певнім стилі (див. наступний параграф) або зробити її спливаючою при влученні покажчика миші на визначене слово. У цьому випадку увід примітки виконується в такий спосіб:

1. Виділіть текст або елемент, із яким потрібно зв'язати примітку, або перейдіть у кінець цього тексту.
2. Меню **Вставка** → команда **Примечание** → Уведіть текст примітки в область приміток насподі екрана → кнопка **Закрить** в області приміток.

Виділений елемент буде пофарбований.

Щоб видалити примітку, варто клацнути по виділеному слову правою кнопкою й у контекстному меню вибрати команду **Удалить примечание**.

У Office 2000 є можливість уставляти звукові примітки.

Для вставки звукових приміток на комп'ютері повинні бути встановлені звукова плата й мікрофон. Потім:

меню **Вставка** → команда **Примечание** → увести текст → кнопка **Добавить звуковой объект** в області приміток → записати звукову примітку через мікрофон → кнопка **Да** у вікні пропозиції **Обновить звуковой объект**.

**Приклад 3.13.** Наприкінці документа Навчальний *текст уставити* сховану примітку – аксіома Кана: якщо ніщо інше не допомагає, прочитайте, нарешті, інструкцію!

**Виконання:** Клацнути наприкінці тексту, потім меню **Вставка** → команда **Примечание** → у вікні, що з'явиться насподі екрана, набрати текст → кнопка **Закрить** у вікні примітки. Зберегти відредагований текст.

Пояснюючі **виноски** можуть бути звичайними й кінцевими. Звичайні виноски проставляються наприкінці сторінки, а кінцеві

наприкінці тексту. Виноски проставляються в такий спосіб: курсор повинен стояти після слова, що потребує пояснення; меню **Вставка** → команда **Сноска** → у діалоговому вікні (рисунок 3.13) вибрати потрібні селекторні кнопки → **ОК** → набрати в рядку, який відкривається насподі сторінки ( документа ), текст виноски.

Приклад 3.14. У документі *Навчальний текст* проставити звичайну виноску до слова *Word* («текстовий редактор») і кінцеву виноску до слова *Форматирование* («Панелі інструментів виставляються через меню Вид»).

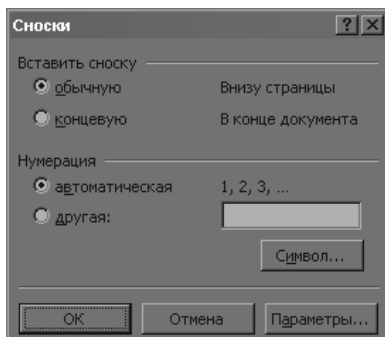


Рисунок 3.13 – Діалогове вікно **Сноска**

### Поле «Дата»


У багатьох документах для автоматизації введення даних можна використовувати поля (більш 60). Поле – це зарезервоване місце для даних, які *Word* автоматично вставляє в документ. Наприклад, можна вставляти дату і час створення документа, дату й час останнього зберігання документа, число сторінок у документі, число слів у документі і т.д. У документі – бланку можна вставляти дату, що буде автоматично обновлятися відповідно до поточної. Для цього дату потрібно вставити як поле:

Меню **Вставка** → команда **Дата и время** → вибрати потрібний формат дати → виставити прапорець **Обновлять автоматически** → **ОК**.

Приклад 3.15. У документі *Навчальний текст* біля часу створення документа (нижній колонтитул) уставте поле з датою, яка буде обновлятися автоматично.

### Виконання:

1) відкрити файл *Навчальний текст* – меню **Файл** → команда **Открыть** → зазначити папку **Мої документи** → клацнути по імені файлу *Навчальний текст* → кнопка **Открыть**;

2) меню **Вид** → команда **Колонтитул** → перейти в нижній колонтитул (  ) → зробити 5 пробілів після зазначеного часу → потім меню **Вставка** → команда **Дата и время** → вибрати потрібний формат дати → виставити прапорець **Обновлять автоматически** → **ОК**;

3) зберегти текст із попереднім ім'ям.

### **Автоматичне формування змісту, списків об'єктів тексту**

При створенні великих документів останні необхідно забезпечувати змістом, можлива вказівка списків ілюстрацій, таблиць, формул і інших об'єктів, використовуваних у тексті. Зазначені роботи можна виконати автоматично за умови використання відповідних стилів: заголовки документа повинні бути відформатовані стилями заголовків *Word*, назви малюнків, таблиць, формул і ін. необхідних об'єктів – стилями назв (див. пункт **Стилі абзаців і символів**).

Формування змісту і різних списків іде через:

м. **Вставка** → к. **Оглавление и указатели** (рисунок 3.14).

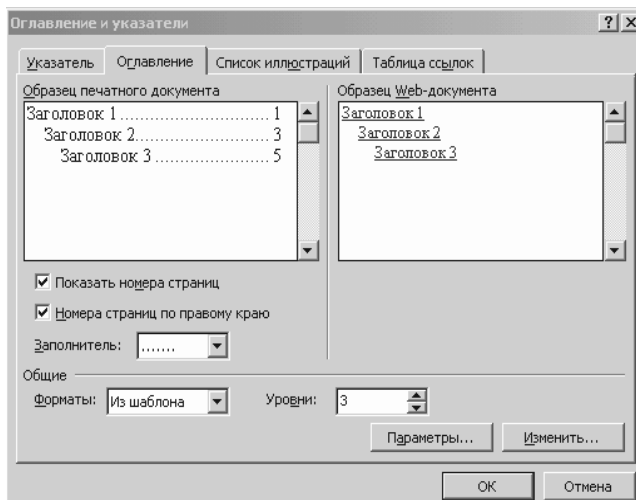


Рисунок 3.14 – Вікно для формування змісту і різних списків об'єктів, що використовуються у тексті.

Приклад 3.16. На першому чистому листі *Навчального тексту* автоматично створити зміст, а на останньому листі – список ілюстрацій.

#### Виконання.

1. Відкрити файл *Навчальний текст* і поставити курсор на перший лист тексту (за прикладом 3.11 лист повинен бути порожнім).
2. **М. Вставка** → к. **Оглавление и указатели** → вкладка **Оглавление** → вибрати формат змісту зі списку форматів → установити число рівнів заголовків (для нашого приклада – 2) → виставити прапорці **Показать номера страниц** і **Номера страниц по правому краю** → вибрати заповнювач → **ОК**.
3. Перейти на останній лист тексту.
4. **М. Вставка** → к. **Оглавление и указатели** → вкладка **Список иллюстраций** → із списку **Название**, що випадає, вибрати **Рисунок** → виставити прапорець **Полное название** → вибрати формат списку зі списку поля **Форматы** → виставити прапорці **Показать номера страниц** і **Номера страниц по правому краю** → вибрати заповнювач → **ОК**.

Аналогічно можна формувати списки таблиць, формул.

Примітка – Якщо зміст або списки включають не відповідні фрагменти тексту, значить вони відформатовані неправильним стилем. Клацання по такому фрагменту перенесе курсор у текст, де цей фрагмент розташований, → виділіть його → змініть стиль фрагменту тексту на **Обычный**.

Обновити зміст або список після внесених змін – виділити й натиснути клавішу **F9** або контекстне меню → к. **Обновить** → указати, що обновити – весь зміст або тільки номера сторінок.

Видаляється зміст або список об'єктів після виділення стандартними способами.

## Злиття

У роботі підприємств досить часто приходиться створювати документи з повторюваною й перемінною частинами. За допомогою злиття можна створювати документи на бланках (запрошення, рекламні листки і т.п.), поштові наклейки, конверти й каталоги.

Постійна частина тексту називається основним документом, перемінна частина – джерело даних. Як джерело даних можна використовувати наступні документи:



- таблицю *Word* рекомендується використовувати при наявності невеликого списку даних – 63 або менше полів даних. Якщо джерело даних *Word* містить більш 63 полів даних, варто перетворити його в текстовий файл із роздільниками;
- адресна книга *Microsoft Outlook*;
- таблиця або запит *Microsoft Access*;
- таблиця *Microsoft Excel*;
- інша база даних.

Робота зі злиття виконується через **М. Сервіс** → к. **Слияние...**

### Приклад 3.17.

1) Створити бланк запрошення на виставку наступної форми і зберегти його з ім'ям *Запрошення*:

#### **Шановн**

Запрошуємо Вас відвідати виставку  
промислового устаткування, що відбудеться  
**12-15 березня** у виставочному центрі  
**«Експодонбас».**

З повагою Оргкомітет виставки.

2) Створити список клієнтів фірми у вигляді наступної таблиці

Прізвище	Закінчення при звертанні	Фірма	Адреса
Іванов Іван Іванович	ий	ТОВ «Пастораль»	83050 р. Донецьк, вул. Щорса, д.31
Гачків Анатолій Олексійович	ий	НКМЗ	86115 р. Краматорськ, вул. Бєляєва, д.36
Реутова Антонина Михайлівна	а	МЧП «Реутова»	83114 р. Донецьк, ін. Полеглих комунарів, 17
Ковальов Сергій Олександрович	ий	АТЗТ «Спектр»	84075 р. Доброполя, вул. Зелена, 18
Лєскієвський Ігор Олегович	ий	ТОВ «Стильні вікна»	83004 р. Донецьк, вул. Артема, д.118, офіс 30
Тарасов Дмитро Вікторович	ий	АТЗТ «АртемСіль»	81001 р. Артемовськ, вул. Санаторна, 22
Корзиков Максим Миколайович	ий	МП «Наші меблі»	86017 р. Краматорськ, вул. Донецька, 7

та зберегти його з ім'ям *Список*.

Примітка – Таблиця в *Word* для злиття створюється без назви.

3) За допомогою інструмента **Слияние**, підставте в текст поле із закінченням у перший рядок і поле з прізвищем, ім'ям, по батькові в порожній рядок. П.І.Б. повинно бути виділене напівжирним, курсивом.

**Виконання:** 1) Виконати перший і другий пункт завдання → відкрити документ *Запрошення*.

2) **М. Сервис** → к. **Слияние** → кнопка **Создать** (рисунок 3.15) → вибрати

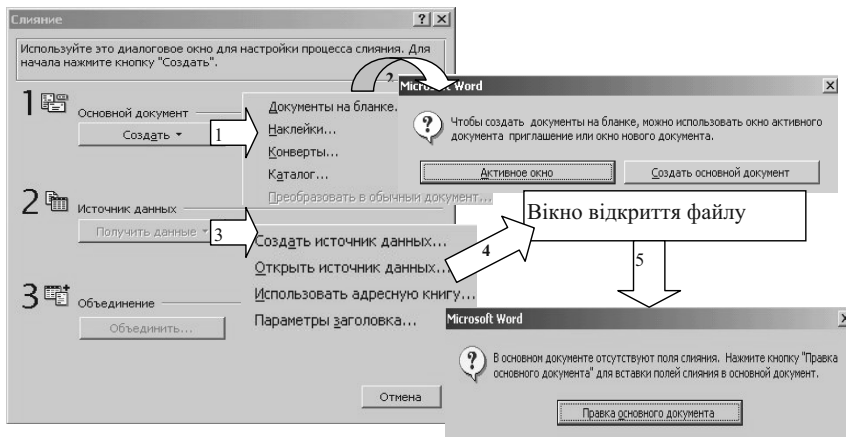


Рисунок 3.15 – Схема роботи користувача при визначенні основного документу та джерела даних

**Документ на бланке** → кнопка **Активное окно** → кнопка **Получить данные** → вибрати **Открыть источник данных** → у вікні відкриття документу указати файл *Список* → кнопка **Открыть** → кнопка **Правка основного документа**.

3) В вікно документа *Запрошення* буде додана панель інструментів **Слияние** (рисунок 3.16) → курсор поставити на місце вставки першого поля (закінчення слова **Шановн**) → відкрити список **Добавить поле слияния** →



Рисунок 3.16 – Панель інструментів **Слияние**

вибрати **Окончание\_при\_обращении** → перейти на місце вставки наступного поля (пустий рядок для П.І.Б) → відкрити список **Добавить поле** → вибрати **Прізвище** (перші два рядки запрошення набудуть наступний вигляд:

Шановн«Закінчення\_при\_звертанні»  
«Прізвище»

4) Виділити рядок «Прізвище» та установити напівжирний курсив звичайним способом.

5) Кнопка **Объединить** на панелі інструментів **Слияние** → у вікні **Слияние** (рисунок 3.17) у полі **Назначение** вказати **Новый документ** → кнопка **Объединить**.



Рисунок 3.17 – Вікно завдання параметрів злиття

У результаті виконаних дій буде сформований файл із готовими запрошеннями для всіх клієнтів фірми (рисунок 3.18).

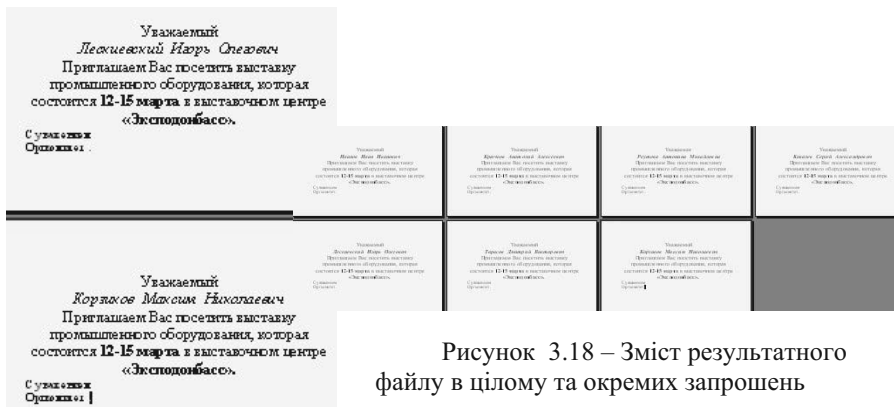


Рисунок 3.18 – Зміст результатного файлу в цілому та окремих запрошень

**Приклад 3.18.** Використовуючи інструмент **Слияние**, надрукувати адреси клієнтів фірми на конвертах для розсилання запрошення.

**Виконання:** **М. Сервис** → к. **Слияние** → кнопка **Создать** (рисунок 3.15) → вибрати **Конверты** → кнопка **Активное окно** → кнопка **Получить данные** → вибрати **Открыть источник данных** → у вікні відкриття документу указати файл *Список* → кнопка **Открыть** → кнопка **Правка основного документа** → у вікні **Параметры конверта** (рисунок 3.19) установити розміри конверта, відступи адрес від краю, напрямок конверта при друкуванні → **ОК** → У вікні **Адрес на**

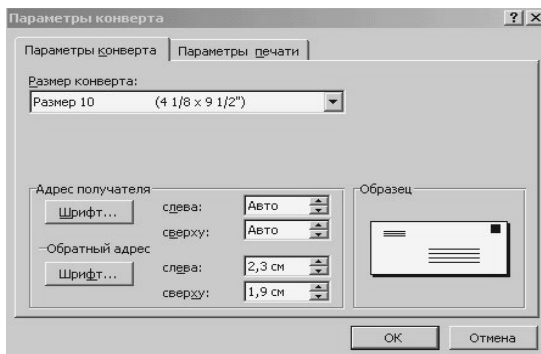


Рисунок 3.19 – Вікно установки параметрів конвертів

конверте (рисунок 3.20) за допомогою кнопки **Вставить поле слияния** вставити поля **Фирма** та **Адреса** → **ОК** → кнопка **Правка основного документа** → на формі конверта набрати адресу відправника → кнопка **Объединить** → у вікні **Слияние** у полі **Назначение** (рисунок 3.17) вибрати потрібний параметр **Новый документ, Принтер** або **Электронная почта** → кнопка **Объединить**.

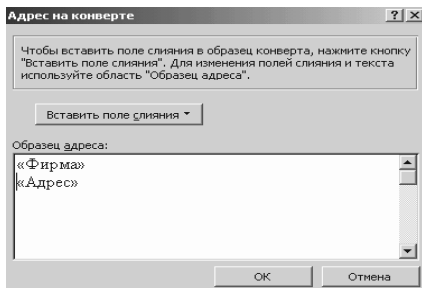


Рисунок 3.20 – Вікно формування адреси отримувача

Примітка – Підкреслений текст необов'язковий. Зворотна адреса може формуватися різними способами, прийнятими у фірмі – фірмові конверти, за допомогою штампа.

У результаті будуть надруковані (представлені у вигляді файлу) конверти з назвами конкретних фірм і їхніх адрес.

## Шаблони

**Шаблоном** називають зразок документів. Вибираючи шаблон для нового документа користувач указує редактору *Word*, яка інформація – текст, форматування, графіка – повинна з'явитися в документі. Чим більше інформації буде внесено в документ автоматично, тим менше залишається ручної роботи. При завантаженні *Word* автоматично завантажується шаблон **Обычный** – файл з ім'ям *Normal.dot*. У той же час мається можливість використовувати інші готові шаблони і створювати свої. Для використання готових шаблонів необхідно вибрати меню **Сервис** → команда **Шаблоны и надстройки** → у діалоговому вікні **Шаблоны и надстройки**, що відкриється (рисунок 3.21) кнопка **Добавить** → вибрати потрібний шаблон із папки **Шаблоны** (для перегляду

за допомогою кнопки  **Представлення** виберіть пункт **Перегляд**)  
→ **ОК** → **ОК**.

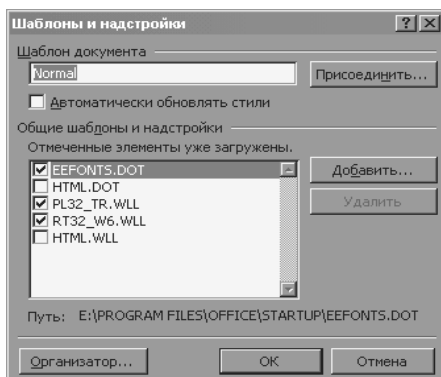


Рисунок 3.21 – Діалогове вікно **Шаблоны и настройки**

При бажанні користувач може створити свій шаблон. Найпростіший шлях – створити документ, який можна використовувати у якості шаблону: меню **Файл** → к. **Сохранить как** → у нижньому полі **Тип файла** розкрити список, що випадає, та вказати **Шаблон документа** → у полі **Имя файла** уведіть ім'я нового шаблону → **ОК**.

### **Закладки й гіперпосилання**

Розглянуті можливості *Word* передбачали формування визначеного документа, що містив би всі необхідні відомості. Однак додатки Microsoft Office дозволяють створювати електронні варіанти документів, які можна збирати із зовсім різнорідних компонентів (документів *Word*, таблиць *Excel*, фрагментів презентацій, малюнків, посилань на сторінки Internet ...). Ці компоненти можуть відкриватися як самостійно, так і з такого електронного документа. Зв'язувати різнорідні компоненти в єдиний документ можна за допомогою гіперзв'язків, засобу, який широко використовується в Internet. Прості способи створення гіперзв'язків розглянемо на прикладі *Word*.

**Гіперзв'язки** – це блоки підкресленого тексту (звичайно виділені блакитним кольором), клацаючи по яких, можна переходити: в інші частини файлу Word, в інші Word-файли на ПК або в локальній комп'ютерній мережі, в інші додатки Microsoft Office і Windows, на сторінки в зовнішній мережі або Internet.

Наприклад, в електронному варіанті даного посібника кожен розділ являє собою окремий файл, який можна викликати зі змісту, а можна відкривати й окремо. Крім того, для безпосереднього навчання на ПК у процесі читання посібника, можна викликати новий документ для набору тексту, лист графічного редактора для створення малюнка, тест для контролю і т.д. Як це можна зробити?

Щоб створити гіперзв'язок з іншим файлом, необхідно:

1. Виділити текст, який треба переформатувати як гіперзв'язок.
2. **М. Вставка** → к. **Гіперссылка** → у вікні **Добавление гиперссылки** (рисунок 3.22) → кнопка **Файл** → указати диск, папку та ім'я файла, який

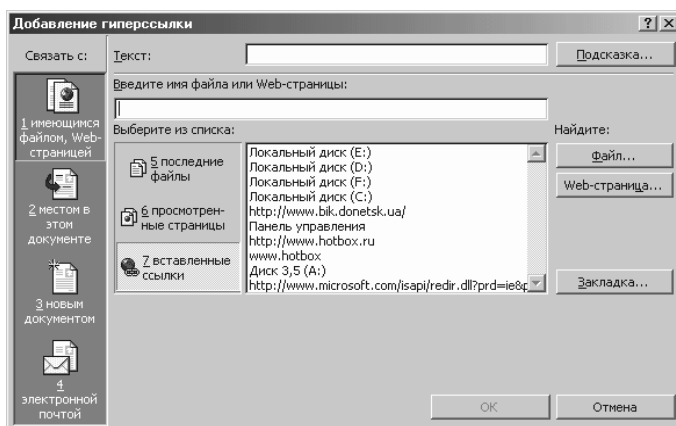


Рисунок 3.22 – Вікно **Добавление гиперссылки**

Щоб створити зв'язок з конкретним місцем якогось файлу, необхідно встановлювати зв'язки із закладками. *Закладки* – поименоване місце в документі або виділений фрагмент тексту, на який передбачається посилатися. Імена закладок повинні починатися з букви і замість пробілів використовувати символ підкреслення \_. Для створення закладки:

1. Виділити текст, що повинен асоціюватися із закладкою.
2. **М. Вставка** → к. **Закладка** → увести ім'я закладки → кнопка **Добавить**.

Для створення гіперзв'язку з конкретною закладкою виконують наступне:

1. Виділити текст, якій треба переформатувати як гіперзв'язок.
2. М. Вставка → к. **Гиперссылка** → у вікні **Добавление гиперссылки** → кнопка **Файл** → указати диск, папку та ім'я файлу, що зв'язується з текстом, → **ОК** → кнопка **Подсказка** → набрати текст підказки для користувача → кнопка **Закладка** в Office 2000 або кнопка **Обзор** праворуч від поля **Поименованное место файла** в Office 98 → виділити потрібну закладку → **ОК** → **ОК**.

Приклад 3.19. Створити новий файл з ім'ям *Текст із зв'язками* наступного змісту:

### Гіперпосилання

Створення тексту з посиланнями на інші документи з можливістю виклику їхнього змісту зв'язано з поняттями гіперпосилань і закладок. Щоб продемонструвати роботу *Word* у цьому напрямку, скористаємося текстом і малюнком, створеними в попередніх прикладах.

Щоб побачити текст, кляцніть тут.

Але якщо потрібен не весь текст, а тільки таблиця цього тексту, кляцніть по слову «таблиця».

Викликати і доробити файл із малюнком можна кляцанням по слову «рисунок».

#### Виконання:

1. Набрати наведений текст та зберегти його в папці **Мої документи** з ім'ям *Текст із зв'язками*.

2. Виділити у тексті словосполучення кляцніть тут → м. **Вставка** → к. **Гиперссылка** → кнопка **Файл** → указати папку **Мої документи** та ім'я файлу *Навчальний текст* → **ОК** → **ОК**.

3. Виділити у тексті слово рисунок → м. **Вставка** → к. **Гиперссылка** → кнопка **Файл** → указати папку **Мої документи** та ім'я файлу *Мій рисунок* → **ОК** → **ОК**.

4. Згорнути вікно **Текст со связями** в значок та відкрити документ *Навчальний текст* → виділити таблицю → м. **Вставка** → к. **Закладка** → увести ім'я закладки **Таблиця** → **ОК** → зберегти зміни в файлі *Навчальний текст* та закрити його → розгорнути вікно **Текст із зв'язками** → виділити слово таблиця → м. **Вставка** → к. **Гиперссылка** → кнопка **Файл**



→ указати папку **Мої документи** та ім'я файлу *Навчальний текст* → **ОК**  
→ кнопка **Закладка** → виділити закладку **Таблиця** → **ОК**.

У результаті виділені в кожному пункті слова будуть пофарбовані в блакитний колір і підкреслені. Показчик миші на цих словах перетворюється в руку з пальцем, що вказує. Подвійне клацання по них буде викликати відповідну інформацію. Повернутися у вхідний текст – закрити викликане вікно.

Для редагування гіперпосилання клацніть правою кнопкою миші по ньому → к. **Гиперссылка** → п. **Изменить гиперссылку** → відредагувати параметри → **ОК**.

Для видалення гіперпосилання клацніть правою кнопкою миші по ньому → к. **Гиперссылка** → п. **Удалить гиперссылку**.

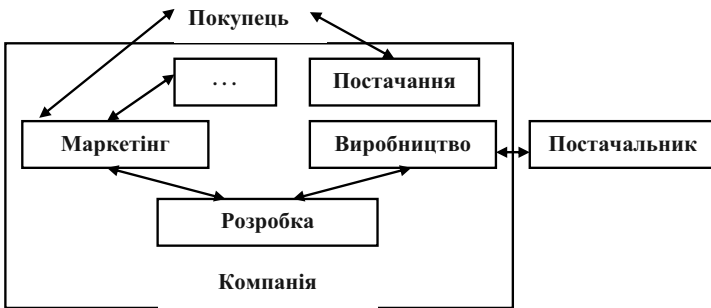
Показані в цьому параграфі прийоми роботи з закладками й гіперпосиланнями можна використовувати й в інших додатках Microsoft Office.

## Завдання для самостійного виконання

1. Запустити *Microsoft Word*.
2. Переглянути вміст рядка меню й склад команд кожного з пунктів.
3. Переглянути склад панелей інструментів для роботи в *Word*. Лишити на екрані панелі **Стандартная, Форматирование й Рисование**.
4. Прибрати й відновити елементи екрана:
  - а) масштабну лінійку
  - б) смуги прокручування
  - в) рядок стану
5. Установити режим перегляду документа **Разметка страницы**, масштаб – **По ширине страницы**.
6. Установити такі параметри сторінки: відступи – 2,5 см, розмір листа А4, орієнтація **Книжная**.
7. У вікні редагування ввести три абзаци довільного тексту із заголовком, у шрифті **Times New Roman (Cyr)**, розмір **14**, використовуючи Автотекст. Розміри абзацу:
  - лівий відступ – 2 см
  - перший рядок – 1,2 см
  - правий відступ – 1 см
8. Скопіювати текст у вікні редагування двічі.
9. Виділити заголовок і ключові слова тексту і замінити в них шрифт на **Полужирный, Курсив** або **Подчеркивание**, розмір 16. Для повторного

форматування використовувати функцію піктографічного меню **Формат по образцу**.

10. Установити міжрядкові інтервали: у першому абзаці – 1 інтервал, у другому абзаці – 1,5 інтервали й у третьому абзаці – 2 інтервали.
11. Установити відстань між абзацами в 6 пунктів до і після.
12. Вирівняти абзаци: перший – **По левому краю**; другий – **По центру**; третій – **По правому краю**; четвертий – **По ширине**. Порівняти й відзначити відмінності.
13. Виділити першу букву першого абзацу у вигляді буквиці.
14. Скасувати команди, що не задовольняють користувача (по власному вибору). Відновити команди (скасувати скасування).
15. Другий абзац узяти в рамку і виконати заливання.
16. Перетворити п'ятий абзац у дві колонки з роздільником.
17. Створити два довільних списки. Перший список пронумерувати засобами *Word*, у другому встановити маркери.
18. Увести схему створення багаторядкового тексту в чарунці:  
 меню **Формат** ⇒ команда **Ячейки** ⇒ вкладка **Вирівнювання** ⇒  
 установити прапорець **Переносить по словам** ⇒ **ОК**
19. Виконати малюнок засобами *Word*.



Модель, що показує, як різні функції забезпечують виконання процесу

20. Створити таблицю:

Можливі значення кон'юнктури інвестиційного ринку	Інвестиційний проект А		
	Обчислюваний дохід, €	Значення можливості, P <sub>i</sub>	Сума очікуваних прибутків (2*3)
Висока	600	0,25	150
Середня	500	0,50	250
Низька	200	0,25	50
У цілому	-	1,00	ε <sub>R</sub> =450

21. Створити колонтитული: у *верхній колонтитул* помістити дату й час створення документа; у *нижній колонтитул* помістити своє прізвище.
22. Перетворити заголовок у фігурний текст (об'єкт *WordArt*)<sup>22</sup>
23. Створити формулу:

$$y = \frac{\sum_{i=1}^n \frac{k_i}{(1+k)^i}}{\sqrt{\frac{\sum_{i=1}^n k_i * p}{c_i}}}$$

24. Помістити в документ малюнок із стандартної бібліотеки.
25. Помістити в документ малюнок попередньо створений за допомогою графічного редактора Paint.
26. Створити одну примітку, одну виноску звичайну й одну кінцеву.
27. Пронумерувати сторінки.
28. Зберегти документ.
29. Виконати попередній перегляд документа перед друком.
30. Надрукувати документ.

## Література

1. Бородкіна І.Л., Матвієнко О.В. Практичний курс з комп'ютерних технологій підготовки даних: Навчальний посібник. – К.: Центр навчальної літератури, 2004. – 448с.
2. Войтюшенко Н.М., Остапєць А.І. Комп'ютерна техніка і програмування. Частина 1. Основи підготовки користувача ПК: Навчальний посібник по базовій підготовці для студентів денної і заочної форм навчання. – Донецьк, ДонДует, 2001 – 150 с.
3. Інформатика: Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів/ За ред. О.І. Пушкаря. – К.: Видавничий центр “Академія”, 2003. – 704 с.
4. Камарда Б. Использование Microsoft Word 97: Пер. с англ. – К.; М.; СПб: Издательский дом “Вильямс”, 1998 – 800с.
5. Компьютеры, сети, Интернет. Энциклопедия./Новиков Ю и др. – СПб.: Питер, 2002. – 928 с.
6. Хэлворсон М., Янг М. Эффективная работа с Microsoft Office 2000 – СПб: Издательство «Питер», 2000. – 1232 с.

## 3.2 Системи оптичного розпізнавання текстів

Досить часто в практичній діяльності виникає необхідність мати електронний варіант документа, фрагмента друкованого тексту. Можна, звичайно, його набрати на клавіатурі, але є більш швидкий і зручний спосіб одержати той же результат – виконати розпізнавання тексту за допомогою сканера і відповідного програмного забезпечення.

Разом зі сканером звичайно, поставляється програмне забезпечення по обробці зображень і документів, за допомогою якого формується електронний образ (фотографія в пам'яті комп'ютера) останніх. Отриманий електронний образ можна зберігати, виводити на друк, передавати в деякі прикладні програми.

Але якщо скануються документи, із якими потрібно продовжувати роботу, коректувати, вилучати фрагменти і т.п., то зображення необхідно перетворити в текстовий файл. Для цього можна використовувати цілий ряд програм. Найбільш розповсюдженою програмою цього класу в нашій країні є *FineReader*. Основні прийоми розпізнавання й збереження документів розглянемо на прикладі версії 7.0.

*ABBY FineReader* – це система оптичного розпізнавання текстів (Optical Character Recognition, OCR). Вона призначена для того, щоб швидко перевести відскановані паперові документи, а також PDF-файли в зручний для подальшого використання електронний формат – файли *Word* або *Excel*, HTML-сторінку, презентацію *PowerPoint*, при цьому цілком зберігається оформлення документа.

*FineReader* дозволяє розпізнавати тексти, набрані практично будь-якими шрифтами. Особливістю програми *FineReader* є висока точність розпізнавання і мала чутливість до дефектів друку.

Для того щоб почати сканування, необхідно включити сканер відповідно до інструкції по користуванню конкретним апаратом і завантажити *FineReader*:

м. **Программы** → к. **ABBY FineReader 7.0** → п. **ABBY FineReader 7.0 Professional Edition**.

Буде завантажено первинне вікно програми (рисунок 3.23).

Панель інструментів Стандартная

Панель інструментів Изображение

Вікно Пакет

Вікно Крупний план

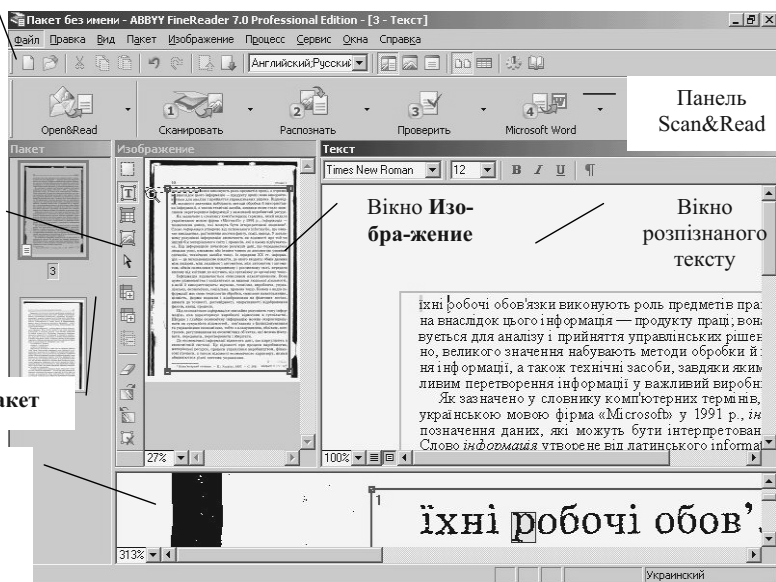


Рисунок 3.23 – Первинне вікно *ABBYY FineReader 7.0 Professional Edition* з результатом сканування й розпізнання

*FineReader*, як більшість серйозних типових програм загального призначення надає користувачеві можливість працювати під керівництвом майстра – спеціальної програми (*Scan&Read*), що за допомогою підказок керує роботою користувача. Однак майстри реалізують звичайно типовий процес автоматично з мінімальним утручанням користувача. Якщо ж користувач хоче контролювати процес розпізнання і, при необхідності, по ходу роботи вносити коректування, то варто одержати деякі навички усвідомленої роботи з даною програмою.

Інтерфейс програми простий і зручний, вона постачена гарною довідковою системою, тому буде розглянута мінімально необхідна послідовність дій для одержання з паперового документа відповідного текстового файлу в пам'яті ПК.

Після завантаження вікно програми може містити тільки меню, панелі інструментів і порожнє вікно **Пакет**. Інші вікна з'являються в процесі сканування. Настроювання усіх вікон здійснюється через меню **Вид**.

Перед початком сканування варто перевірити або установити необхідні параметри роботи програми: м. **Сервис** → к. **Опции** (рисунок

3.24 ) → вкладка **Сканирование/Открытие** → перевірити установку потрібного драйвера сканера → установити селекторну кнопку **Использовать интерфейс TWAIN-драйвера сканера** для забезпечення попереднього перегляду кожної сторінки, що сканується → при скануванні книг, журналів виставити прапорець **Делить разворот книги** → вкладка **Распознание** → список, що випадає, **Язык распознания** → установити мови для розпізнання тексту → кнопка **Закреть**.

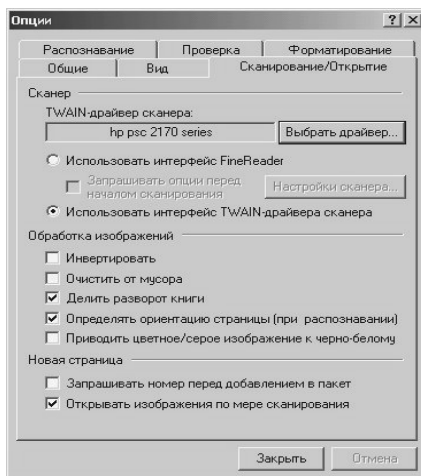


Рисунок 3.24 – Вікно для установки параметрів роботи програми

Примітка – 1. Установити мови розпізнання можна також через список, що випадає, панелі інструментів **Стандартная**.

2. При установці декількох мов треба клацнути по першій позиції списку (**Выбор нескольких языков**), установити прапорці потрібних мов і клацнути по кнопці **ОК**.

Далі необхідно покласти документ текстом униз у відповідності зі стрілкою на самому сканері.

Процес уведення документа в комп'ютер і перетворення його в текстовий файл складається з наступних етапів:

1. **Сканування** – сканер зчитує зображення і передає його в графічному вигляді комп'ютерові. При цьому отримане зображення неможливо відредагувати в жодному текстовому редакторі.

*FineReader* працює зі сканерами через TWAIN-інтерфейс. Це єдиний міжнародний стандарт, введений у 1992 році для уніфікації взаємо-

дії пристроїв для введення зображень у комп'ютер (наприклад, сканера) із зовнішніми додатками. У режимі **Использовать интерфейс TWAIN-драйвера сканера**, як уже відзначалося, доступна функція попереднього перегляду зображення (preview), що дозволяє точно задати розміри області, що сканується, підібрати яскравість, відразу контролювати результати цих змін. Вікно й діалог TWAIN-драйвера сканера в кожного сканера виглядає по-своєму. Вигляд цього вікна й зміст опцій описаний у документації, прикладеної до сканера. Для прикладу було використано багатофункціональний пристрій hp psc 2175 (принтер, сканер, копір).

Виконання сканування: у програмі *FineReader* на панелі інструментів **Scan&Read** розкрити список другої від лівого краю великої кнопки → к. **Сканировать изображение** → буде викликано діалогове вікно сканера (рисунок 3.25) → при необхідності скорегувати настройки сканера → клацнути по кнопці

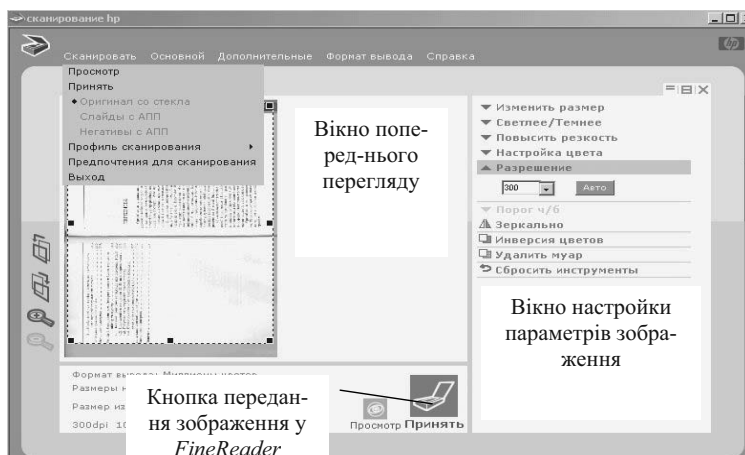


Рисунок 3.25 – Вікно сканера

**Принять** → у вікні *FineReader* з'явиться зображення сторінок(и) пакета у вікні **Пакет** і у вікні **Изображение**.

**Примітка** – При скануванні текстів рекомендується використовувати вирішення 300 dpi (розмір шрифту 10 і більш пунктів) і 400–600 dpi для текстів, набраних дрібним шрифтом (9 і менш пунктів).

2. **Сегментування** – виділення блоків тексту. Блоки – це ув'язнені в рамку ділянки зображення. Блоки виділяють для того, щоб указати

системі, які ділянки відсканованої сторінки треба розпізнавати й у якому порядку. Також по них відтворюється вихідне оформлення сторінки. Блоки різних типів (текст, картинка, таблиця, штрих-код ) мають різні кольори рамок.

Сегментування може виконуватися автоматично при клацанні по кнопці **Распознать** або вручну при частковому розпізнанні документа. Для ручного виділення блоків використовуються кнопки панелі інструментів **Изображение** (рисунок 3.23).

Примітка 1. Панелі інструментів можна поповнювати відсутніми кнопками стандартним для офісних програм способом: м. **Вид** → к. **Панели инструментов** → п. **Настройка** (рисунок 3.26) → установити панель інструментів → установити потрібну категорію → вибрати команду, наприклад, **Удалить блок**, і за допомогою клавіші >> перенести в перелік кнопок панелі → кнопка **Закреть**.

Примітка 2. Для правильного розпізнання ліній у таблиці викликати контекстне меню → п. **Анализ структуры таблицы** → при необхідності використовувати інструменти по роботі з таблицею панелі інструментів **Изображение** для редагування отриманого результату.

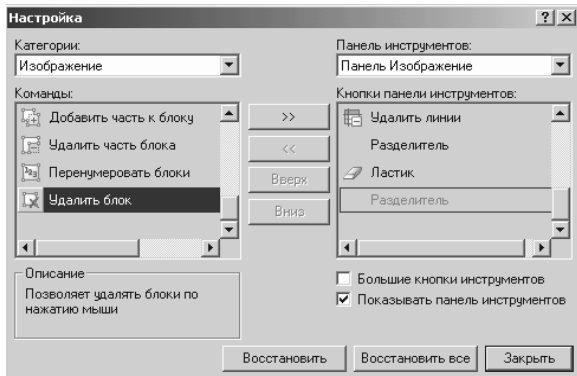


Рисунок 3.26 – Вікно настройки панелей інструментів

**3. Розпізнавання** – аналіз і розпізнавання кожного символу, виконується клацанням по кнопці **Распознать**. У результаті роботи у вікні *FineReader* з'явиться розпізнаний текст, який можна відредагувати і зберегти в найбільш зручному для користувача форматі.

**4. Збереження** – а) збереження пакета в обраному форматі (28 форматів) із метою подальшої роботи з ним;



б) збереження розпізнаних документів у вигляді тексту з зазначеним користувачем форматом (вибір з 20 форматів) для подальшого продовження роботи з текстом уже в обраній прикладній програмі.

5. **Передача сторінок тексту безпосередньо в прикладну програму** – список, що випадає, останньої кнопки праворуч панелі інструментів **Scan&Read** (рисунок 3.27) → к. **Передать все страницы в** → вибрати потрібну програму → *FineReader* сам відкриє вказану програму та завантажить у неї текст. Далі користувач може продовжувати працювати у обраній прикладній програмі.

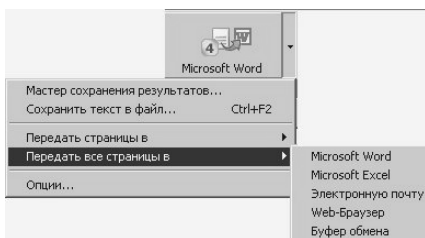


Рисунок 3.27 – Список, що випадає, кнопки **Сохранить**

6. **Друк зображення** – можна надрукувати одне зображення, відкрите у вікні **Изображение**, декілька зображень, виділених у вікні **Пакет**, або усе зображення. Для цього виконати: м. **Файл** → п. **Печать изображения** → у діалозі **Печать**, що відкриється, установити параметри друку (принтер, кількість сторінок, що друкується, кількість копій і т.д.).

Поглиблене знайомство з можливостями програми *FineReader* можна здійснити, використовуючи керівництво користувача: м. **Справка** → к. **Справка** (к. **Обучение на примерах**, к. **Техническая поддержка**).

## Контрольні питання

1. Які апаратні і програмні засоби необхідні для автоматизованого перетворення паперового документа в текстовий файл на ПК?
2. Етапи автоматизованого перетворення паперового документа в текстовий файл на ПК?
3. Що таке TWain-інтерфейс і для чого він потрібний?

4. Через який пункт меню набудовується вигляд вікон програми *FineReader*; параметри роботи?
5. Можливості *FineReader* по розпізнанню і подальшій обробці тексту?

## Література

1. Інформатика: Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів / за ред. О.І.Пушкаря. – К.: Видавничий центр “Академія”, 2003 – 704 с.
2. Справочная система пакета *FineReader*.

### 3.3 Системи електронного перекладу

Сучасний світ характеризується стрімким розвитком інформаційних зв'язків між підприємствами, країнами, окремими людьми, незважаючи на границі і просторове положення. У багатьох випадках це зв'язано зі спілкуванням на різних мовах, що вимагає в найкращому варіанті гарного знання іноземної мови або наявність перекладача. Сьогодні комп'ютер надає нам можливість використання електронних перекладачів. Але користувач повинен правильно оцінювати можливості останніх і необхідність установки тієї або іншої програми електронного перекладу.

Перекладом називається процес і результат створення на основі вхідного тексту на одній мові рівноцінного йому в комунікативному відношенні тексту на іншій мові. При цьому під комунікативною рівноцінністю розуміється як така якість тексту перекладу, що дозволяє йому виступати у якості повноправної заміни вхідного тексту (оригіналу) у сфері дії мови перекладу [1].

Рівноцінність нового тексту стосовно оригіналу забезпечується виконанням трьох основних вимог:

- текст перекладу повинен у можливо більш повному обсязі передавати зміст оригіналу, що насамперед означає неприпустимість довільного опущення або додавання інформації;
- текст перекладу повинен відповідати нормам мови перекладу, тому що їхнє порушення, щонайменше, створює перешкоди для сприйняття інформації, а іноді веде і до її перекручування;

- текст перекладу повинен бути приблизно порівнянний з оригіналом по своєму обсязі, чим забезпечується подібність стилістичного ефекту з погляду лаконічності або розгорнення вираження.

Узагалі, перші програми МП з'явилися в 50-х роках, усього кілька років потім після народження комп'ютера, але до широкого поширення ПК МП був скоріше цікавим об'єктом наукових досліджень, чим важливою сферою використання обчислювальної техніки, по двох причинах: дорожняча часу роботи комп'ютера і колективне користування його ресурсами. Остання обставина часто не дозволяла негайно звернутися до електронного помічника, зводячи нанівець найважливішу перевагу МП перед звичайним – його оперативність.

### **3.3.1 Загальна характеристика програм машинного перекладу**

Датою народження машинного перекладу прийнято вважати 1947 р., коли директор відділення природничих наук Рокфеллерівського фонду Уоррен Уивера написав лист до родоначальника кібернетики Норберта Вінера, у якому задачу перекладу порівнював із задачею дешифрування текстів.

Фірма ІВМ разом із Джорджтаунским університетом у США в 1954 р. показала першу систему, що базується на словнику з 250 слів і 6 синтаксичних правил, яка забезпечувала переклад 49 заздалегідь відібраних пропозицій. Цей експеримент поклав початок дослідницькому бумові: у наступні 10 років уряд і військові відомства США витратили на дослідження в області машинного перекладу близько 40 млн. доларів. Але жоден із дорогих проектів не привів до створення працюючої системи, і в 1967 році спеціально створена комісія Національної Академії наук США оголосила машинний переклад безперспективним і не заслуговуючим фінансуванням.

Тільки на початку 80-х років, коли ПК упевнено почали завойовувати світ, час їхньої роботи значно подешевів і доступ до них можна було одержати в будь-яку хвилину, відновилися дослідження в області машинного перекладу.

Поява сканерів і програм оптичного розпізнавання текстів, поширення Інтернет/інтранет обумовили реальний попит на програми машинного перекладу.

На початку 90-х з'явилися перші вітчизняні продукти, здатні виконувати прямий і зворотний переклад з російської на англійську, а та-

кож із російської на українські мови. А з популяризацією Internet одержали поширення програми для перекладу вмісту Web-сайтів, а також комп'ютерні перекладачі з Web-доступом.

В даний час використовуються дві основних технології перекладу за допомогою комп'ютерів.

Перша, часто називана Transfer, будується як комплекс трьох процесів. На першій стадії вхідний текст аналізується з розбивкою на групи іменників, прикметників, дієслів і прислівників, а також визначенням основи й закінчення кожного зі слів. Далі виробляється перетворення цієї структури в аналогічну структуру вхідної мови, після чого іде синтез кінцевої пропозиції за отриманою структурою.

Інша, рідше використовувана технологія Interlingua (проміжної мови), припускає створення якоїсь мови, що описує всі структури вхідної й вихідної мов. Тому процес перекладу тут проходить усього в два етапи: аналіз вхідної пропозиції у термінах проміжної мови і потім генерація відповідної пропозиції вихідною мовою. Найбільшу складність представляє розробка саме проміжної мови з описом у відповідних термінах природної мови. У зв'язку з цим технологія Interlingua застосовується менш часто, чим Transfer.

Для перекладу об'ємних текстів із великою кількістю повторів застосовується також інша технологія, що використовує концепцію Translation Memory. Робота подібної системи заснована на використанні баз відповідностей значенневих одиниць. Таким чином, перед початком роботи з такою системою потрібно створити цю базу відповідностей або скористатися готовою. При цьому не можна створити універсальну базу для усіх видів текстів, як правило, для кожної задачі створюється своя власна база відповідностей. У результаті, пропозиції або їхні частини, відсутні в базі, не будуть оброблятися взагалі, але зате ті, що є, будуть переводитися завжди правильно.

Саме з цієї причини застосування технології баз відповідностей виправдано лише для об'ємних текстів із численними повторами, наприклад, технічної документації, довідників і т.п.

Кращі на сьогодні програми машинного перекладу використовують комплекс різноманітних лінгвістичних, кібернетичних методів і підходів і їхня розробка являє собою складну, наукомістку задачу.

Сказавши про складності створення самих систем машинного перекладу, перейдемо до більш цікавого для користувача питання – якості перекладу. Тестування [5,6] і досвід користувачів показує, що

якість машинного перекладу ще далеко від ідеалу. З художнім текстом задовільно не справляється жодна з існуючих програм – перекладачів. Що стосується творчої частини, то в доступному для огляду майбутньому в змаганні комп'ютер – людина завжди перемаже "живий" перекладач. Однак для рішення проблем, обумовлених трудомісткістю процесу перекладу, комп'ютерні системи можуть виявитися гарною підмогою.

#### Достоїнства програм-перекладачів:

- наявність словників по спеціальностях, миттєвий пошук, розташування в кілька вікон, можливість одночасного огляду декількох варіантів перекладу;
- можливість створення власного словника користувача;
- підключення до текстового редактора Microsoft Office, що дозволяє, не відриваючись від оригіналу і його перекладу, вибрати підходящі відповідності для перекладу;
- можливість перекладу з різних мов;
- компактність, наявність практично необмеженого обсягу інформації у комп'ютері;
- швидкість пошуку – переклад багато сторінкового тексту може здійснюватися усього за кілька секунд. Це дозволяє швидко зрозуміти зміст тексту, крім того, якщо система правильно настроєна на тематику тексту, згодом буде потрібно мінімальне редакторське виправлення;
- надійний доступ до послуги з меншими витратами. Комп'ютерний перекладач завжди під рукою, у той час як використання послуг бюро перекладів або просто людини-перекладача вимагає витрат часу й грошей. За даними користувачів компанії ПРОМТ при перекладі від 50 сторінок тексту на місяць програма-перекладач ПРОМТ 98 окупається приблизно за місяць [4];
- конфіденційність. Електронні перекладачі часто використовуються в особистому листуванні, для перекладу різної закритої (наприклад, ділової) кореспонденції. Однак інформація подібного роду вимагає високої точності перекладу, що програмні засоби не зав-

жди можуть забезпечити, а точніше можливість цього залежить від складності структури вихідного тексту.

- переклад інформації в Інтернеті. В он лайні найбільше яскраво виявляються всі переваги систем машинного перекладу. Більш того, у більшості випадків переводити інформацію в Інтернеті, якщо Ви, звичайно, самі не знаєте декількох мов, можна тільки за допомогою програм-перекладачів. Саме ця потреба обумовила величезний ріст інтересу до систем машинного перекладу в усім світі. Тільки завдяки онлайн-системам перекладу з'явилася можливість переглядати іноземні сайти, не утруднюючись із їхнім перекладом. Крім того, тут діють усі перераховані вище достоїнства систем машинного перекладу: переклад текстів по будь-якій тематиці виробляється швидко й конфіденційно.

#### Недоліки програм-перекладачів:

- обмеженість огляду, обумовлена розмірами екрана, розкритий великий словник дозволяє побачити відразу набагато більшу кількість значень слова, особливо якщо воно має багато значень;
- словники-перекладачі не дотримують у повному обсязі правил граматики, стилістики й лексики, не враховують гру слів, художні прийоми;
- найчастіше текстові перекладачі вибирають одне зі значень багатозначного слова, що може не відповідати контекстові;
- при відсутності слова в словнику не переводять його;
- електронні словники-перекладачі часто видають кілька варіантів перекладу слова на іншу мову, не пояснюючи різниці в тих або інших словникових відповідностях, що утрудняє правильний вибір тієї або іншої відповідності в даному контексті.

Таким чином, хотілося б підкреслити, що програма-перекладач – це, насамперед, інструмент, що дозволяє вирішити проблеми перекладу або підвищити ефективність праці перекладача тільки в тому випадку, якщо він використовується грамотно.

До засобів автоматизації перекладу відносяться наступні типи програм – електронні словники й програми – перекладачі.

*Електронні словники* – засіб перекладу окремих слів і виражень. Використання їх дуже схоже на звичайні паперові словники, але непорівнянну швидкість пошуку, можливі обсяги й варіанти перекладу.

Найвідомішим і популярним електронним словником є **АВВУУ Lingvo**. Багатомовна версія Lingvo 10 – це 79 загальних, тематичних і країнознавчих словників; 5 мов (з англійської, німецької, французької, італійського, іспанського на російську); 10 мільйонів перекладів із докладними коментарями й прикладами використання; більш 100 безкоштовних додаткових словників й можливість створення власних словників; навчальний додаток для тих, хто вивчає мови, – Lingvo Tutor.

Приклади інших електронних словників.

**Контекст версія 6.0.** У Базовий комплект увійшли 4 англо-російсько-англійських словники: Загальної лексики – 460 000 термінів; Комп'ютерний – 110 000 термінів; Комерційний – 60 000 термінів; Труднощі англійської мови – 1 000 термінів.. Комплект містить основну лексику з численними прикладами вживання. Загальний обсяг складає більш 600 000 словникових статей. Мається тлумачний словник російської мови з показом наголосів. Інформація про шукане слово в словниках представлена детально: за бажанням користувача виводяться синоніми, помітки, приклади або тлумачення.

**Polyglossum II** включає медичні, банківські, технічний словники на 3 млн. слів; основні мови – російська, англійська, німецька.

**МультиЛекс 4.0** Англійський для професіоналів. Містить у собі англо-російський словник і російсько-англійський словник і 15 спеціалізованих словників із широким охопленням тематик. Обсяг лексики всіх сімнадцяти словників складає 3 000 000 англійських і 3 500 000 російських слів. Спеціалізовані словники охоплюють наступні теми: економіка, банківська справа, фінанси, юриспруденція, обчислювальна техніка, Інтернет, програмування, політехніка, фізика, будівництво, поліграфія, видавнича справа, нафта й газ, сонячна енергетика, медицина, соціологія. Мається убудована програма синтезу мови.

*Програми-перекладачі* – повно функціональні програми, що обробляють документ від його введення в комп'ютер до перекладу, редагування й збереження на іншій мові.

Приклади електронних програм текстових перекладачів:

Сімейство перекладачів компаній ПРОектМТ і Арсенал – **PROMT** (нащадки колись популярного електронного перекладача **Stylus**). Продукт поставляється в декількох варіантах комплектації: від «легкого» Express, що містить тільки програму *Prompt X*, до *Prompt XT Professional*, що комплектується максимальним числом різних модулів. У стандартний комплект постачання входять 17 спеціалізованих словників, також можна підключати свої власні. Можливі одноязычні й багатомовні версії, наприклад, *X-Translator* – переклад російсько-англійський і навпаки.

*Prompt XT* підтримує прямий і зворотний переклад з англійської, французької, німецької на російську, а також виконує однобічний переклад з іспанської й італійської на російську. Програма має базовий (загальний/генеральний) словник і спеціалізовані словники, що підключаються додатково.

Генеральний словник містить загальнозвичиване значення слова. Додаткові словники – спеціалізоване значення для визначеної тематики. При перекладі *PROMT* спочатку шукає слово в спеціалізованому словнику, причому саме в тім порядку, у якому вони підключені (зверху вниз). Якщо не знаходить слово в першому спеціалізованому словнику, то переходить до другого, якщо не знаходить у жодному з підключених спеціалізованих словників, то бере значення слова з генерального (загального) словника.

*Prompt XT* має багато корисних функцій, таких як озвучування тексту при наявності встановлених на комп'ютері засобів синтезу мови, сумісних із стандартом Text-To-Speech, і синхронний переклад, коли текст обробляється програмою відразу після введення чергового фрагмента пропозиції.

У процесі перекладу програма відбирає незнайомі слова і показує їх в окремому вікні. Далі їх можна перевести самостійно і занести в користувальницький словник. При необхідності, деякі слова можна позначити як зарезервовані (наприклад, торговельні марки компаній і т.д.).

До складу програми може входити модуль Prompt E для перекладу Web-сторінок у MS Internet Explorer.

Продукт *Prompt XT Office* у даний час є однозначним лідером по числу напрямків перекладу і функціональності. Тому його корисно й зручно використовувати для потреб керування, у науковій діяльності і



т.д. – скрізь, де є необхідність працювати з інформацією на декількох іноземних мовах.

**Magic Gooddy** – програма компанії ПРОектМТ із функцією підключення голосу й озвучування перекладу, зручний і до того ж забавний інтерфейс. Її можна визнати самим універсальним сімейним перекладачем. Вона цілком годиться для перекладу навчальних текстів, листів, посібників до програм, інтерактивної довідки, і багато чого іншого, аж до кулінарних рецептів.

Ті, кому вже приходилося зіштовхуватися з «Чарівником Гудді», наполегливо рекомендують цю програму дітям. *Magic Gooddy* є коштовним доповненням до навчальних програм, головним чином, завдяки можливості читати вголос довільні тексти.

Якість перекладу залежить від того, наскільки точно оригінал відповідає однієї з наявних тематик. Тексти "на стику" областей, а також розмовні переводяться гірше.

Програма машинного перекладу **Pragma 4.x** (розроблювач Trident Software ) прийшла на зміну більш ранній і досить популярній системі **L-Master 98**.

У цій версії програми автоматично визначається мова вхідного документа і переклад здійснюється у всіх напрямках між російською, українською, англійською, німецькою й латиською мовами.

Програмне забезпечення *Pragma* включає два незалежних додатки – Монітор і Коректор. Вони дозволяють переводити текстові документи з однієї мови на іншу у зазначених напрямках, а також створювати й коректувати користувальницькі словники.

Після інсталяції *Pragma Монітор* постійно запущений в операційній системі і його іконка [#] видна на панелі задач біля годинника.

Система машинного перекладу *Pragma* може бути поставлена з різними функціональними можливостями в залежності від обраного типу пакета.

У базовому пакеті переклад підтримується в MS *WordPad*, MS *Блокнот*, системний буфер обміну (Clipboard).

В Офісному пакеті переклад підтримується в MS Office версій 97 і більш пізніх, StarOffice, OpenOffice.

В Інтернет – пакеті переклад підтримується в Internet Explorer, Netscape Navigator, Outlook Express, MS Outlook, Eudora.

У пакеті *Експерт* переклад підтримується у всіх перерахованих вище додатках, коректор функціонує в повному обсязі (перегляд і коректування словників, альтернатив і непереведених слів).

Мережний пакет – переклад підтримується у всіх перерахованих вище додатках, коректор функціонує в повному обсязі; підтримка багатокористувальницького режиму з можливістю відновлення по мережі.

Система цікава наявністю української мови і можливістю перекладу з різних мов на різні ( 20 напрямків перекладу)!

Система автоматичного перекладу **Сократ Персональний 4.1** здійснює переклад усього в двох напрямках – із російської на англійську і назад. Це гранично проста система для ознайомлювального перекладу невеликих фрагментів тексту. Її козири – компактність і наявність убудованого електронного словника, вона невимоглива до кваліфікації користувача і максимально автоматизована.

*Сократ Персональний 4.1* – безсумнівний лідер у швидкості перекладу. Текст обсягом у кілька тисяч знаків оброблявся усього за кілька секунд. Якість перекладу цієї системи не завжди висока, але зате мало залежить від тематики тексту. Відповідно, ця програма знайде застосування при інтенсивній роботі з різноманітними іноземними текстами на ознайомлювальному рівні.

Найкраще піддаються перекладові з її допомогою тексти інтерактивної довідки. Їхній зміст удається зрозуміти у всіх випадках, а більшість пропозицій побудовані грамотно. Також гарні результати можна одержати при роботі з діловими листами – тут позначаються гарна збалансованість словника, великий запас економічної лексики, а також строгість і шаблонність ділового стилю мови, під яким, зокрема, і набувався програмний алгоритм "Сократа". Задовільними можна вважати результати роботи з документацією до програм і посібниками до побутових приладів, однак, зрозуміти їхній зміст, як правило, можна.

Слабким місцем програми є наукові тексти, особисте листування, розмовні теми й особливо керівництва до комп'ютерних ігор. У першому випадку позначається відсутність у складі "Сократа" словників спеціальних термінів. Інші ж являють собою природне утруднення для всіх систем машинного перекладу, викликане неформалізованою лексикою, складністю оборотів і стилістичною розмаїтістю зазначених текстів.

По кількості різних функцій дана програма досить бідна – із додаткових засобів у її склад входить лише електронний словник. У на-строюваннях програми не можна вказати спеціалізований словник під час перекладу, а також визначити зарезервовані слова. Хоча, за інфор-мацією розроблювача, Сократ Персональний 4.1 автоматично враховує тематику і стилістичні особливості тексту. З іншого боку, інтерфейс продукту надзвичайно простий і дозволяє приступити до роботи через лічені хвилини після установки програми.

У **Пролінг Офіс Стандарт** об'єднані такі широко відомі програми фірми ПроЛінг, як:

система перевірки правопису українських і російських текстів – **РУТА**,

програма перекладу текстів із російської мови на українську і на-впаки – **ПЛАЙ**

і російсько-українсько-російський електронний словник **УЛИС**.

За допомогою програми *РУТА* можна:

- перевірити правопис, тобто автоматично знайти й виправити помилки в словах;
- здійснити граматичний, пунктуаційний і стилістичний кон-троль у пропозиціях;
- розставити переноси в словах при форматуванні тексту;
- скористатися словником синонімів і близьких за значенням слів (тезаурусом);
- завантажити в Windows тексти, що набрані в будь-якій україн-ській кодовій таблиці DOS;
- зберегти український текст, що набраний у Windows, у коду-ванні DOS, затвердженому державним стандартом, або в будь-якому ін-шому;
- виправити помилки нестандартного кодування букв української мови, у тому числі при перенесенні текстів із DOS у Windows і навпаки.

Програма *ПЛАЙ* призначена для перекладу текстів із російської мови на українську і з української на російську.

Майстер установки ProLing Office вбудовує РУТУ і ПЛАЙ у додатки, із якими працює користувач. Функції РУТА і ПЛАЙ можуть бути задіяні при роботі в наступних додатках:

РУТА 5.0 працює з продуктами Microsoft – Word 2000,2002 і 2003; Excel 2000,2002 і 2003; Outlook 2000,2002 і 2003; Access 2000,2002 і 2003; PowerPoint 2000,2002 і 2003; Publisher 2000,2002 і 2003; FrontPage 2000,2002 і 2003;

З продуктами Adobe – PageMaker 6.0,6.5 і 7.0.

ПЛАЙ 5.0 працює з продуктами Microsoft -Word 2000,2002 і 2003.

У середовищі WINDOWS 95/98/NT і зараз активно працюють із системами РУТА і ПЛАЙ версії 4, основний словник якої містить більш 160000 основ і дозволяє розпізнавати більш 4,5 млн. словоформ української мови.

Надзвичайно корисною є можливість використання словника синонімів українських слів і постійних словосполучень. Словник містить понад 5 тисяч синонімічних рядів, приблизно 30 тисяч слів. Використовуючи словник можна швидко і коректно замінити слова і вираження в тексті синонімами (словами й вираженнями, близькими за значенням), що дозволяє уникнути повторів і монотонності, зробити текст більш виразним. Якщо вхідне слово багатозначне, то користувачеві пред'являється кілька синонімічних груп, при цьому слова, що входять у кожну з груп, цілком можуть бути замінені одне на одне в контексті. Список синонімів з'являється перед користувачем у тій же граматичній формі, у якій знаходиться вхідне слово, що дозволяє уникнути помилок при заміні слів у конкретному контексті.

З програмою РУТА поставляються додаткові утиліти:

– редактор користувацький словників (*Ruta User Dictionary Editor*);

– програма "підказування" у режимі on-line (*Ruta On-Line Tools*), що дозволяє при роботі з будь-яким застосуванням Windows для слів, що вводяться з клавіатури, перевірити правильність написання, розміщення переносів, або підібрати синоніми для такого слова;

– редактор розкладки клавіатури KeyCaps;

– простий перекладач буфера обміну Clipboard;

Використовуючи програму *Ruta User Dictionary Editor* можна переглянути й відредагувати створений користувачем словник. Після запуску на екрані з'явиться діалогове вікно, у якому можна побачити список словників користувача, які можна редагувати. Можна дода-

ти існуючий словник до цього списку або не використовувати кожної з підключених словників при подальшій роботі. Можна також створити новий словник і переглянути й відредагувати зміст будь-якого обраного зі списку в діалоговому вікні словника.

За допомогою програми *Ruta On-Line Tools* можна здійснювати спрощену перевірку правопису безпосередньо в процесі введення українського тексту в будь-якому додатку Windows.

Після запуску цієї програми на екрані з'явиться її діалогове вікно, що розташовується над усіма іншими вікнами. У ньому можна бачити слова, що вводяться з клавіатури.

У випадку помилкового набору тексту, у полі "Suggested word" з'являться запропоновані *РУТА* варіанти правильного написання помилкового слова. Обравши курсором один із запропонованих варіантів і, натиснувши кнопку "Сору", можна обраний варіант копіювати в Clipboard (Буфер Обміну) і його можна додати до тексту, що вводиться, властивим для поточного застосування методом (як правило, за допомогою натискання комбінації клавіш CTRL+V).

Таким чином, якщо користувач працює тільки з російською й українською мовами, йому безумовно, підходить перекладач відповідної версії фірми ПроЛінг.

### 3.3.2 Основні прийоми роботи з додатком ПроЛінг Офіс Стандарт (програми РУТА і ПЛАЙ)

Установка програми виконується Майстром установки (програма *Setup Proling Office*), у результаті роботи якої програми *РУТА* і *ПЛАЙ* убудовуються в *Word*.

Змінене меню *Word* показано на рисунку 3.28.

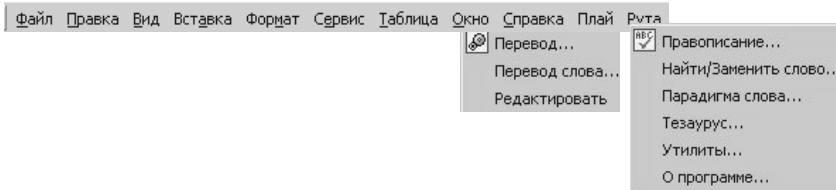


Рисунок 3.28 – Меню *Word* з підменю програм *ПЛАЙ* і *РУТА*

Настроювання всіх компонентів програми *Пролінг Офіс Стандарт* здійснюється через спеціальну програму:

м. **Программы** → к. **Proling Office** → п. **Proling Office Настройка** (рисунок 3.29) → настроїти параметри → **ОК**.

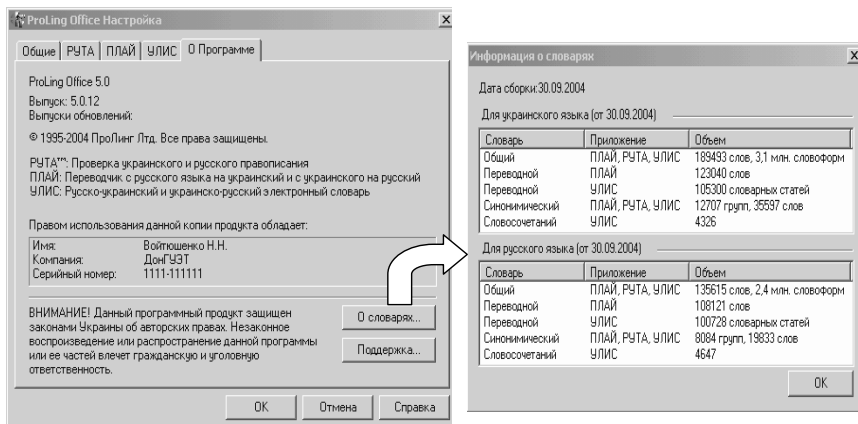


Рисунок 3.29 – Вікна настройки компонентів програми *Пролінг Офіс Стандарт*

Далі робота виконується гранично просто. Якщо необхідно перевести тільки одне слово, то слід його виділити, потім:

м. **Плай** → к. **Перевод слова...** → у вікні **Плай: Варианты перевода и синонимы** (рисунок 3.30) клацанням миші вибрати потрібний варіант → кнопка **Заменить**.

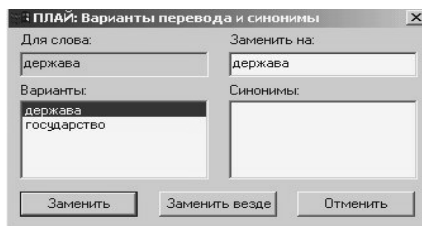


Рисунок 3.30 – Вікно **ПЛАЙ** для вибору варіантів перекладу

Якщо необхідно виконати переклад усього документа, то:

м. **Плай** → к. **Перевод...** → у вікні **Плай: Опции перевода** (рисунок 3.31)

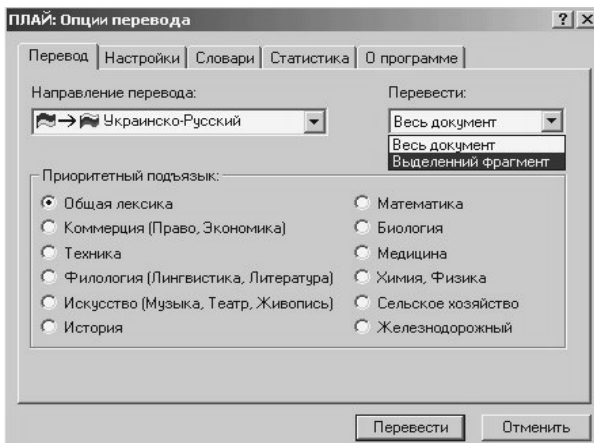


Рисунок 3.31 – Вікно настройки опцій перекладу

→ вкладка **Перевод** → перевірити напрямок перекладу, указати пріоритетну підмову; у списку **Перевести**, що випадає, вибрати **Весь документ** → вкладка **Настройки** → виставити прапорець **Включить режим редактирования ПЛАЙ после перевода** → кнопка **Перевести** → у вікні **Плай: Редактирование перевода** (рисунок 3.32) користувач бачить результати аналізу перекладу і може внести в нього свої поправки → перекладений текст з'явиться у новому вікні.

Примітка – У результаті аналізу користувач може побачити у вікні редагування наступне:

- Відсутність слова в словнику (слово виділяється **червоним** кольором, натисканням на праву клавішу миші Ви можете додати його до словника користувача).

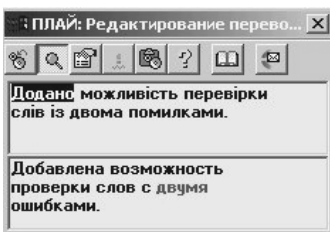


Рисунок 3.32 – Вікно редагування перекладу

- Наявність варіантів перекладу багатозначних слів (слово виділяється **зеленим** кольором, натисканням на праву клавішу миші Ви можете обрати відповідний варіант із запропонованого списку).

- Омонімічна форма слова, що припускає приналежність до декількох частин мови (слово виділяється **синім** кольором, натисканням на праву клавішу миші Ви можете уточнити її, обравши із запропонованого списку).

- Слово містить у собі латинські символи або символи, що не належать до обраної мови вхідного тексту (слово виділяється **блакитним** кольором).

- Зарезервоване слово (слово виділяється **бузковим** кольором).

Переклад фрагмента тексту виконується аналогічним образом. Фрагмент тексту для перекладу попередньо виділяється, а при завданні опцій на вкладці **Перевод** у полі **Перевести** слід указати **Выделенный фрагмент**.

У версії 4 програми *ПЛАЙ* мається можливість переводити фрагмент тексту із заміною на місці, що для невеликих фрагментів було достатньо зручно.

Робота з текстом може бути продовжена засобами програми *ПУТА*.

У випадку, якщо користувача не влаштовує перевірка правопису засобами *Microsoft Word*, стає корисною перевірка правопису (орфографії й граматики) за допомогою програми *ПУТА*.

**М. Рута** → к. **Правописание** → програма починає перевірку правопису в поточному документі. При виявленні орфографічної помилки система видає діалогове вікно (рисунок 3.33), де на вкладці **Орфографія** можна вибрати запропонований варіант заміни або виправити слово вручну (кнопка **Заменить**), замінити слово в усім документі (кнопка **Заменить все**), внести слово в словник користувача (кнопка **Добавить**), проігнорувати помилку, натиснувши кнопку **Продолжить**, проігнорувати такі ж помилки в усім документі (кнопка **Пропустить все**).

Якщо програма зустрічає граматичну помилку в реченні, то вікно перевірки правопису відкривається на закладці **Граматика** (рисунок 3.32) із короткою рекомендацією щодо виправлення знайденої помил-



ки. Натиснувши кнопку **Объяснения**, можна переглянути граматичне правило щодо виявленої помилкової ситуації. Функції інших кнопок такі ж, як і на попередній закладці.

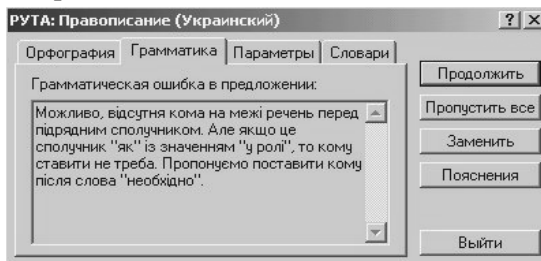


Рисунок 3.33 – Вікно перевірки правопису при наявності помилок

**Примітка** – Речення з граматичними помилками в тексті не виділяються хвилястою зеленою лінією, як це робиться при автоматичній перевірці граматики в англійському або російському текстах. Пошук і виправлення граматичних помилок здійснюється лише по команді перевірки правопису.

На закладці **Словари** можна оперативнo відкрити та відредагувати усі словники, які використовуються користувачем під час перевірки.

Користувач може одержати допомогу програми по напису слів у різних відмінках, часах за допомогою:

виділити слово → м. **Рута** → к. **Парадигма слова...** (рисунок 3.34).

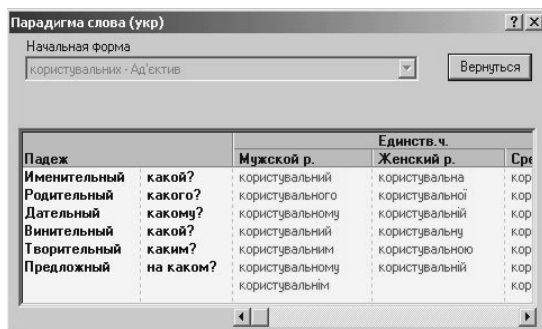


Рисунок 3.34 – Інформація про словоформи виділеного слова

За допомогою *РУТА Експрес* можна реалізовувати базові функції перевірки правопису безпосередньо в процесі введення українського тексту в будь-якій прикладній програмі *Windows* (крім *MS Word*). Крім того, для слів, що набираються, можна переглянути варіанти переносу і підібрати синоніми.

Запуск програми *РУТА Експрес*:

м. **Пуск** → к. **Програми** → п. **ProLing Office** → папка **Утилити** → програма **РУТА Експрес** → у вікні програми будуть відображені слова, що набираються (рисунок 3.35).

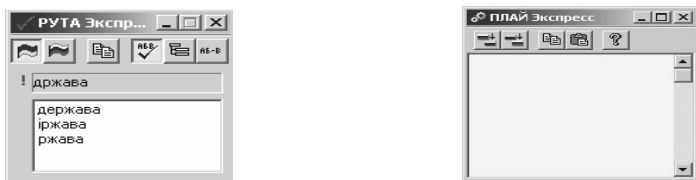


Рисунок 3.35 – Вікна програми фонові перевірки правопису *РУТА Експрес* та експрес-перекладача *ПЛАЙ Експрес*.

*ПЛАЙ Експрес* – це полегшений варіант україно-російського й російсько-українського перекладача, що поставляється у виді самостійного програмного модуля. Його можна запустити:

м. **Пуск** → к. **Програми** → п. **ProLing Office** → папка **Утилити** → програма **ПЛАЙ Експрес**. На екрані з'явиться вікно, яке зображено на рисунку 3.35.

Ця утиліта використовується для швидкого перекладу невеликих текстів із метою ознайомлення з їхнім змістом. Текст можна перевести, перетягнувши виділений фрагмент із текстового редактора в поле перекладача за допомогою миші або скопіювавши й уставивши потрібний текст через буфер обміну. Текст миттєво переводиться і через буфер обміну його можна перенести в потрібний документ.

### 3.3.3. Програма *Pragma 4.x*

При необхідності багатомовного перекладу (англійська, німецька, українська, російська, латиська) із включенням української мови можна рекомендувати програму – перекладач *Pragma 4.x*.

Програму можна придбати прямо через Інтернет або скачати демонстраційну версію, що працює тільки в On-Line (усі словники знаходяться на сервері). Монітор постійно висить у пам'яті і готовий до роботи. Через панель інструментів Pragma –монітора (рисунок 3.36) ви-кликається будь-який додаток, із яким може працювати перекладач.



Рисунок 3.36 – Панель інструментів Pragma – монітора

Функція швидкого перекладу використовується для перекладу тексту в текстових вікнах різних додатків. Переведений текст відображується у невеличкому вікні (рисунок 3.37).

Є кілька варіантів запуску вікна **Быстрый перевод** через кнопку *Pragma-Монітора* в статус-рядку:

1. Викликати спливаючого меню *Монітора*, клацнувши правою клавішею миші, а потім вибрати к. **Быстрый перевод** з меню.

2. Виділити текстовий блок у будь-якому вікні і клацнути по кнопці *Монітора* лівою кнопкою миші.

Другий варіант не тільки запустить вікно **Быстрый перевод**, але і виконає переклад виділеного тексту.

Якщо вікно **Быстрый перевод** відкрите, будь-яке копіювання тексту в буфер через меню додатка або шляхом натискання комбінації клавіш **Ctrl-C** буде приводити до активізації швидкого перекладу.

Програма-перекладач автоматично намагається визначити мову вхідного тексту, а напрямок перекладу необхідно вказувати за допомогою кнопки **Параметры** (рисунок 3.37). У відповідному вікні (рисунок 3.38) уточнюється напрямок перекладу і тематика.

Дві натиснуті кнопки з фіксацією – виділений текст перекладається і зберігається у буфері

Копіювати у буфер.  
Кнопка з фіксацією

Переклад виділеного тексту або змісту вікна. Режим включений по замовчанню.

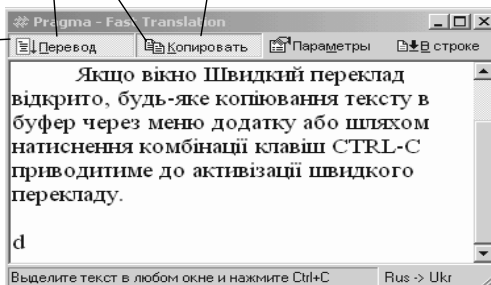


Рисунок 3.37 – Вікно **Быстрый перевод**

Для відключення функції **Быстрый перевод** досить мінімізувати або закрити вікно.

*Pragma 4.x* вбудовується в деякі офісні й Інтернет додатки. При цьому в правій частині меню додатка відображається додаткове меню **Pragma**.

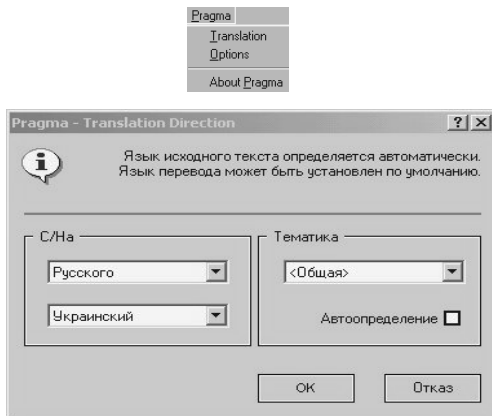


Рисунок 3.38 – Вікно установки параметрів перекладу

Процес перекладу динамічно відображається в допоміжному вікні внизу додатка. Переклад може бути перерваний натисканням клавіші **Esc**. По закінченні перекладу видається діалогова панель вибору варіанта збереження результуючого документа (рисунок 3.39).

При роботі системи *Pragma 4.x* у програмі *MS Word* мається режим розмітки переведеного тексту. Зеленим кольором відзначаються слова, що мають кілька варіантів перекладаємих значень. Червоним – слова, не знайдені в перекладному словнику. Розмітка тексту зручна при наступному редагуванні переведеного документа. При цьому документ може бути збережений без утрати розмітки. Для швидкого видалення розмітки документа необхідно натиснути клавішу **Esc**.

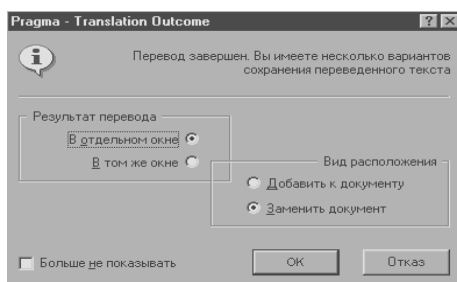


Рисунок 3.39 – Вікно вибору варіанта збереження результуючого документа

Режим розмітки включений за замовчуванням і може бути відключений у панелі **Параметры**.

Виклик довідки системи здійснюється при клацанні правою кнопкою миші по значку **Pragma – монітора** на **Панелі задач** (рисунок 3.40).

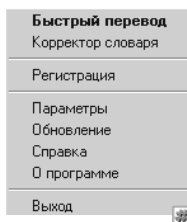


Рисунок 3.40 – Контекстне меню значка **Pragma – монітора** на **Панелі задач**

## Література

1. Алешков М.А. Программы-переводчики. Осваиваем сами. – СПб.: Наука и Техника, 2005. – 144 с.
2. Войтюшенко Н.М., Остапець А.І. Комп'ютерна техніка і програмування. Частина 1. Основи підготовки користувача ПК: Навчальний посібник по базовій підготовці для студентів денної і заочної форм навчання. – Донецьк, ДонДУ-ЕТ, 2001 – 150 с.
3. Информатика: Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів / за ред. О.І.Пушкаря. – К.: Видавничий центр “Академія”, 2003 – 704 с.
4. Компьютеры, сети, Интернет. Энциклопедия./Новиков Ю и др. – СПб.: Питер, 2002. – 928 с.
5. Перепичка И.И. Курсовая работа на тему «Машинный перевод». – Донецк. <http://www.meo.ru/catalog/312/62518/html>
6. Рябинин Д. Софт-обзор: программы-переводчики для Pocket PC <http://www.hpc.ru/lib/arts/1890/>
7. Хэлворсон М., Янг М. Эффективная работа с Microsoft Office 2000. – СПб: Издательство «Питер», 2000. – 1232с.

## Розділ 4. Табличний процесор Excel

### 4.1 Основи роботи з листом

#### 4.1.1 Загальна характеристика й область застосування програм обробки електронних таблиць. Програма Excel

Програми, призначені для обробки даних, поданих у табличному виді, називаються табличними процесорами. Усі програми даного типу при завантаженні видають на екран порожню стандартну таблицю, яку користувач може заповнювати даними, виконувати над ними різноманітні операції, змінювати форму таблиці, подавати результати обчислень у графічному виді.

Історія розвитку табличних процесорів нараховує трохи більш десяти років, але в наявності величезний прогрес у цій області розробки програмного забезпечення. Прикладами широко поширених програм даного типу є: *LOTUS 1-2-3*, *SUPERCALC 4,5*, *QVADRO*, *IMPROV*, *EXCEL*. Остання виконана під операційну систему *WINDOWS* і претендує на звання самої потужної і продуктивної програми обробки електронних таблиць на сьогодні. Саме з цією програмою ми і познайомимося.

*EXCEL* можна використовувати як для рішення простих задач обліку, так і для упорядкування різноманітних бланків, ділової графіки, і навіть повного балансу фірми. Наприклад, на підприємстві за допомогою *EXCEL* можна полегшити рішення таких задач, як обробка замовлень і планування виробництва, розрахунок податків і зарплати, облік кадрів і витрат, керування збутом і майном, а також багатьох інших.

Проте застосування програми не обмежується тільки сферою ділового життя. Завдяки потужним математичним і інженерним функціям за допомогою *EXCEL* можна вирішити безліч задач у галузі природних і технічних наук.

Основою програми є обчислювальний модуль, за допомогою якого відбувається обробка даних (текстових або числових) у таблицях. Для створення презентаційної графіки використовується модуль діаграм, що

дозволяє на основі результатів обчислень одержати діаграми різноманітних типів. За допомогою модуля бази даних реалізований доступ до зовнішніх баз даних. Можливості обробки й упорядкування даних значно розширені завдяки використанню Майстра зведених таблиць. Модуль програмування дозволяє користувачу не тільки автоматизувати рішення складних задач, але і створити власну оболонку програми.

Завантаження програми EXCEL:

кнопка **Пуск** на панелі задач → пункт **Програми** → підпункт **Microsoft EXCEL**.

Завершити роботу з програмою можна за допомогою одного з таких варіантів:

1. варіант – клацнути лівою кнопкою миші по кнопці системного меню вікна – пункт **Закричь**;
2. варіант – меню **Файл** робочого вікна **EXCEL** → команда **Выход**.

#### 4.1.2 Вікно EXCEL і його елементи

Після завантаження табличного процесора на екрані відображається головне вікно програми з чистою робочою книгою.

**Робоча книга** містить від 3 до 16 робочих листів у залежності від початкових установок, але є можливість уставляти додаткові листи (до 256).

**Робочий лист** (таблиця) складається з 256 стовпчиків і 16384 рядків. Для різних форм представлення інформації використовуються різноманітні листи.

**Діаграми** опрацьовуються й зберігаються в листах діаграм, хоча можна розташовувати їх і на робочому листі.

Крім робочої книги вікно програми (рисунок 4.1) містить велику кількість різноманітних елементів. Стандартні елементи вікна Windows були розглянуті раніше.



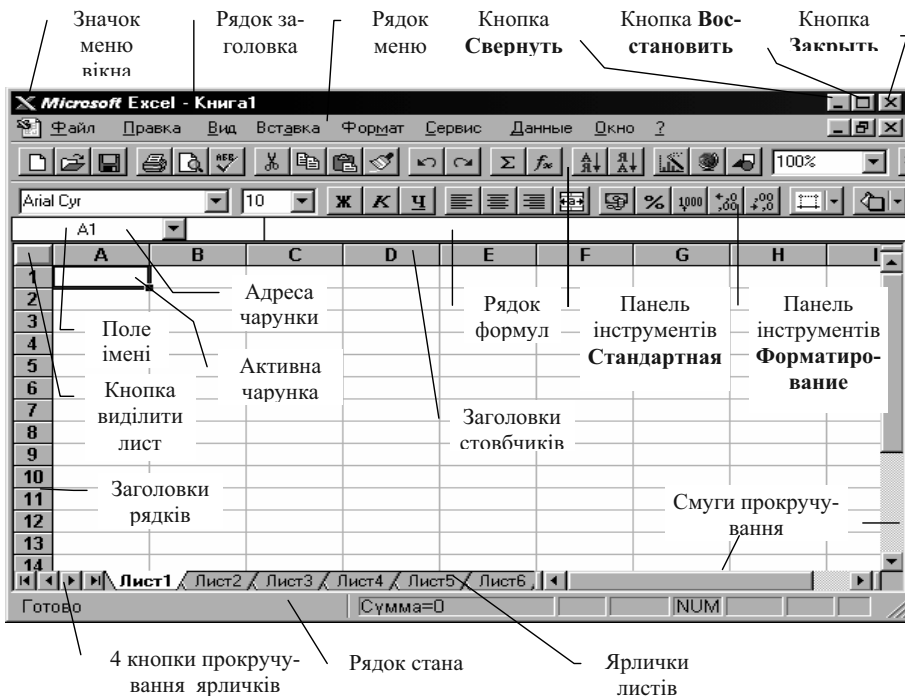


Рисунок 4.1 – Головне вікно програми EXCEL

Під рядком меню розташовані два рядки панелей інструментів, які призначені для швидкої активізації виконання визначених команд меню й функцій програми. Настроювання панелей інструментів здійснюється через меню **Вид** → команда **Панели инструментов**.

Під рядками панелей розташований **рядок формул**, у якому відбиваються дані, що вводяться в активну чарунку, а також виконується редагування вмісту поточної чарунки (клацнути мишкою по тексту і виконувати редагування).

У нижній частині вікна знаходиться **рядок стана**, у якому відбиваються різноманітні текстові повідомлення, графічний процес завантаження робочої книги або компонентів програми і та ін.

### 4.1.3 Заповнення таблиці робочого листа EXCEL

Після запуску програми покажчик чарунки (курсор) знаходиться в чарунці A1, A1 є *активною*.

Усі чарунки таблиці мають посилання або адреси (латинські букви – адреси стовпчиків, цифри – адреси рядків).

*Активна чарунка* обрамлена рамкою. В активну чарунку вводяться дані – текст або цифри. Причому, якщо довжина текстової інформації перевищує довжину чарунки, то вона вводиться цілком, але видно лише частину до першої зайнятої справа чарунки.

Пересуватися по таблиці можна за допомогою клавіш із стрілками, смуг прокручування або клацанням миші активізувати потрібну чарунку.

Занесені дані можна копіювати, переміщати в інше місце таблиці, видаляти як по одній чарунці, так і діапазоном чарунок.

*Діапазон чарунок* указується початковою чарункою і кінцевою чарункою, розділеними двокрапкою (:). Як виконувати названі операції показано в таблиці 4.1

Таблиця 4.1. – Виконання операцій з діапазоном чарунок

<b>Що робити</b>	<b>Як робити</b>
1	2
Виділення чарунок	Клацнути лівою кнопкою миші по першій чарунці діапазону (товстий світлий плюс), натиснути ліву кнопку і тягти до останньої чарунки діапазону. Виділення показане чорним кольором
Виділення стовпчиків (рядків)	Клацнути по заголовку крайнього стовпчика (рядка) лівою кнопкою, затиснути кнопку і тягти до кінця діапазону
Виділення всього листа	Клацнути по кнопці <b>Выделить лист</b>
Зняти виділення	Клацнути в будь-якому місці таблиці
Копіювання значень чарунок у суміжну область	Виділити діапазон чарунок, що копіюється. Покажчик миші поставити на чорний квадратик (маркер заповнення) у правому нижньому куті рамки виділення (повинен утворитися «+»), натиснути ліву кнопку миші і тягти в потрібні чарунки (праворуч, ліворуч, нагору, вниз). Зняти виділення

## Продовження табл. 4.1

1	2
Копіювання значень чарунок у несуміжну область	Виділити діапазон чарунок, що копіюється. Установити покажчик миші на нижній межі виділеного інтервалу чарунок. Покажчик набуває вид стрілки. Утримуючи натиснутими клавішу <b>[Ctrl]</b> і ліву клавішу миші, перемістити покажчик у потрібне місце таблиці. Зняти виділення. <b>Перевірте формули, чи те ви хотіли одержати.</b> <b>АБО</b> Виділити діапазон чарунок, що копіюється, → меню <b>Правка</b> → команда <b>Копировать</b> ; виділити діапазон чарунок для копії й меню <b>Правка</b> → команда <b>Вставить</b> .
Переміщення значень чарунок	Виділити діапазон чарунок, що переміщується. Установити покажчик миші на нижній межі виділеного інтервалу чарунок. Покажчик набуває вид стрілки. Нажати ліву клавішу миші і перемістити покажчик у потрібне місце таблиці. Зняти виділення.
Видалення значень чарунки (очищення чарунок)	Виділити діапазон чарунок, що очищується, і натиснути клавішу <b>[Delete]</b> . Зняти виділення.
Видалення стовпчиків (рядків)	Виділити стовпчики (рядки) → меню <b>Правка</b> → команда <b>Удалить</b> .
Вставка стовпчиків (рядків)	Виділити стовпчики (рядки), куди необхідно помістити нові → меню <b>Вставка</b> → команда <b>Столбец (Строка)</b> або клацнути по кнопці <b>Вставить таблицу</b> панелі інструментів <b>Стандартная</b> .

У чарунки, в яких повинні утримуватися результати обчислень, вводяться відповідні формули.

**Формули** являють собою числа, посилання на чарунки, функції, пов'язані між собою знаками операцій (+, -, \*, /). **Усі формули завжди починаються зі знака (=)** і **набираються в латинському алфавіті**.

Наприклад:  $=D7 * F7 + C7 / E7$  ↵.

При правильному внесенні формул у чарунці з'являться нулі або #ДЕЛ/0 (при відсутності даних) або результат. Поява символів #ЗНАЧ або #ИМЯ говорить про нечислові значення у використовуваних чарунках або про помилки в наборі формул.

При занесенні розрахункових формул рекомендується посилання на чарунки вказувати клацанням миші по цих чарунках. Це дозволить уникнути переключень з одного алфавіту в інший і забезпечить правильний запис посилань.

Якщо формули по своїй схемі однакові, то можна не повторювати їхнє введення, а виконати копіювання з автоматичним настроюванням на нові чарунки. Для цього необхідно активізувати потрібну чарунку або виділити інтервал чарунок, а потім виконати копіювання, як зазначено в таблиці 4.1.

При підрахунку підсумкових сум можна використовувати кнопку **Автосума**  $\Sigma$  на панелі інструментів **Стандартная**. ПК записує функцію, у якій користувач повинен уточнити діапазон (при необхідності зазначити його виділенням або ввести з клавіатури), потім натиснути клавішу **[Enter]**.

**EXCEL** дозволяє вводити в суміжні чарунки готові списки (назви місяців, днів тижня й ін.), переглянути які можна виконавши:

Меню Сервис → команда Параметры → вкладка Списки

Для введення списку необхідно активізувати чарунку для розміщення першого (не обов'язково за списком) елемента списку й увести в неї його назву, а потім установити покажчик миші на маркер заповнення й перетягнути його на необхідну для списку кількість чарунок у необхідному напрямку.

Для створення нового списку необхідно виконати:

Меню Сервис → команда Параметры → вкладка Списки → у вікно **Элементы списка** ввести новий список → кнопка **Добавить** → **Ок** (рисунок 4.2)

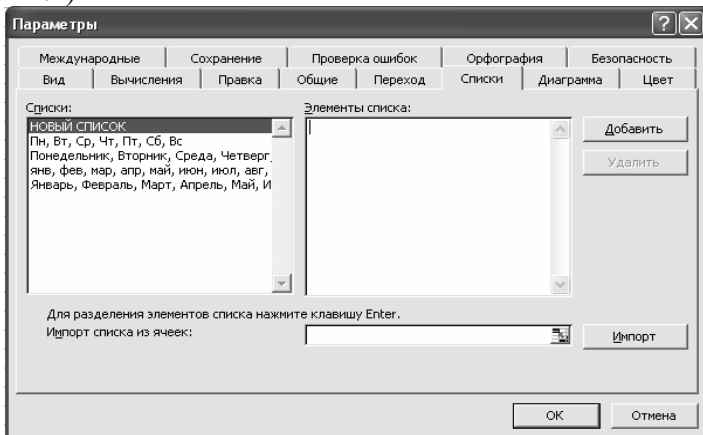


Рисунок 4.2 – Діалогове вікно створення списків

Таблиця *EXCEL* має однакові стандартні чарунки, що частіше усього не відповідає потребам користувача. При необхідності її можна відформатувати.

Форматування таблиці можна виконувати як до заповнення останньої, так і після заповнення. Останній варіант краще, тому що більш наочний. Для цього можна:

- використовувати в меню **Формат** пункт **Автоформат**, у якому вибирається зі списку потрібний формат. Наприклад, Стандарт 1, у якому автоматично розпізнаються й виділяються клітини з результатами, змінюється ширина стовпчиків із текстовими даними;
- виконувати форматування самостійно.

Засоби вибору ширини стовпчиків показані в таблиці 4.2

Таблиця 4.2 – Засоби зміни ширини стовпчиків

Що зробити?	Як зробити?
Автоматичний добір ширини по максималь-но заповненій чарунці	Виділити потрібні стовпчики → меню <b>Формат</b> → пункт <b>Столбцы</b> → команда <b>Автоподбор ширини</b>
Завдання ширини по відомому розміру в см	Виділити потрібні стовпчики → меню <b>Формат</b> → пункт <b>Столбцы</b> → команда <b>Ширина</b> , задати розмір у см
Зміна ширини безпосередньо в робочому листі	Покажчик миші поставити на розділювальну лінію між стовпчиками в рядку заголовків стовпчиків (утвориться чорний хрест), затиснути ліву кнопку миші і тягти в потрібну сторону, відпустити кнопку

Засоби зміни ширини рядків аналогічні, проте цього мало, щоб в чарунці утворювався багаторядковий текст, що часто зустрічається в заголовній частині документів. Для одержання багаторядкового тексту в чарунці необхідно:

- 1) текст повинний не поміщатися по ширині в чарунку;
- 2) меню **Формат** → команда **Ячейки** → вкладка **Вирівнювание**.
- 3) виставити прапорець **Переносить по словам**, тут же можна зазначити засіб вирівнювання тексту в чарунці *по горизонталі й вертикалі* → **ОК**.

Використовуючи на панелях інструментів відповідні кнопки, можна змінювати шрифти і їхнє написання, присвоювати чарункам із числовими даними грошовий стиль.

Для надання завершеного виду документу необхідно виділити заголовочну і підсумкову частину, в предметній частині виділити рядки й стовпчики, тобто зробити рамки. У *EXCEL* це виконується за допомогою панелі інструментів рамки (межі), виставленої через меню **Вид** або:

меню **Формат** → команда **Ячейки** → вкладка **Границя**. Попередньо варто виділити частину таблиці (усю таблицю), а потім клацнути по відповідній кнопці або йти через меню.

При бажанні, можна виділити заголовні рядки й стовпчики за допомогою кольору. Для цього виділяють потрібну частину таблиці та:

меню **Формат** → команда **Ячейки** → вкладка **Вид** → вибрати колір і (або) узор → **ОК**.

На закінчення оформлювальної роботи можна прибрати сітку з екрана і з друкарського документа:

меню **Сервіс** → команда **Параметри** → вкладка **Вид** → прибрати прапорець **Сетка** → **ОК**.

Для зберігання робочої книги в меню **Файл** треба вибрати команду **Сохранить** або натиснути кнопку **Сохранить** і задати ім'я файлу.

Звичайно при роботі з табличним процесором спочатку створюють форму документа з формулами і зберігають її для багатократного використання. А потім відкривають її, заповнюють і при необхідності зберігають з іншим ім'ям:

меню **Файл** → команда **Сохранить как**.

Після заповнення форми частина таблиці, що обчислювалася, може бути заповнена символами ###. Це пов'язано з недостатньою шириною чарунки. Як збільшити ширину чарунки зазначено в таблиці 4.2

У випадку сумнівів у правильності обчислень, тобто правильності занесених формул, можна скористатися режимом виводу формул:

меню **Сервис** → команда **Параметры** → вкладка **Вид** → установити прапорець **Формулы** → **ОК**. На екрані з'явиться ваша таблиця з усіма внесеними формулами.

Примітка – У режимі формул таблиці видаються на екран у розтягнутому виді, їх можна форматувати для видачі на друк.

*EXCEL* дає можливість візуально простежити **зв'язки** у формулах. Для цього варто скористатися:

меню **Сервис** → команда **Зависимости** і вибираються необхідні підпункти.

#### 4.1.4 Створення діаграм

Для створення діаграм у табличному процесорі *EXCEL* використовується *Мастер диаграмм*, який можна викликати одним із таких засобів:

- 1) меню **Вставка** → команда **Диаграмма**;
- 2) клацнути по кнопці **Мастера диаграмм**.

**Майстер діаграм** послідовно видає 4 вікна, відповідаючи на питання яких, користувач може швидко підібрати й побудувати наочну графічну інтерпретацію своїх даних. Перехід до наступного (попереднього) вікна Майстра діаграм – кнопки **Далее** (**Назад**).

**В першому вікні** клацанням вибирається *тип* діаграми з запропонованого набору й *вид* із зазначеного типу (рисунок 4.3).

Майстру діаграм необхідно зазначити *діапазони даних*, по яких варто виконати побудову діаграм. Для цього потрібно виділити відповідні інтервали чарунок. Якщо необхідні декілька не суміжних інтервалів, то при виділенні варто утримувати натиснутою клавішу **[Ctrl]**.

Зазначену операцію можна виконати до виклику Майстра діаграм або при роботі **в другому вікні** Майстра. Діапазони даних можна також набрати вручну (рисунок 4.4).

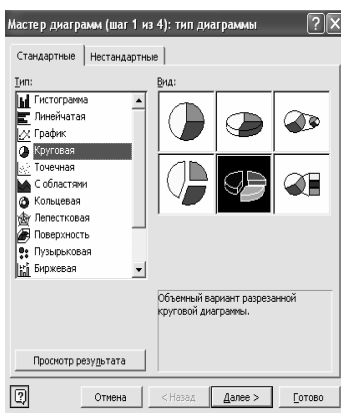


Рисунок 4.3 – Вікно **Мастера диаграмм** (крок 1)

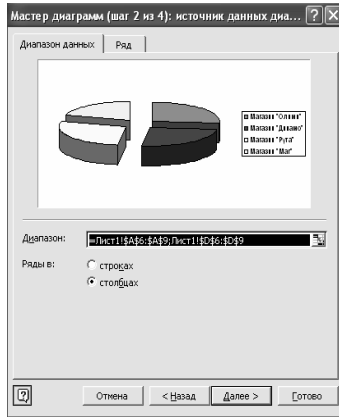


Рисунок 4.4 – Вікно **Мастера диаграмм** (крок 2)

**У третьому вікні** користувач одержує зразок діаграми і може змінити її вигляд за рахунок рядків і стовпчиків, що виділяються на легенду (на позначення). Якщо діаграма Вас не влаштовує, то ви можете повернутися в будь-яке попереднє вікно Майстра і змінити свій вибір. Тут же вносяться підписи – назва діаграми, осей (рисунок 4.5).

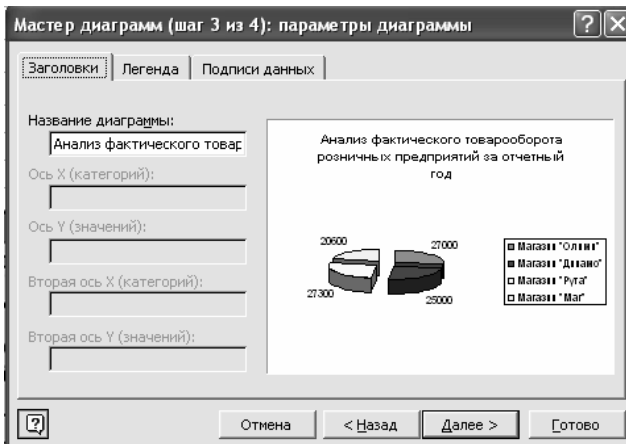


Рисунок 4.5 – Вікно **Мастера диаграмм** (крок 3)

**Четверте вікно** – вибір місця розташування діаграми – робочий лист або лист діаграм (рисунок 4.6).



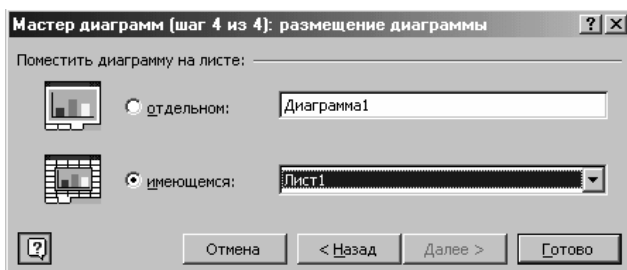



Рисунок 4.6 – Вікно **Мастера диаграмм** (крок 4)

При виборі розміщення на **наявному** листі діаграма з'являється в центрі екрана й її можна рухати й змінювати розміри. Для цього варто виділити діаграму, клацнувши в області діаграми, після чого навколо неї з'явиться чотирикутник із восьми чорними квадратами по периметру. При розміщенні покажчика миші на такий квадратик він набуває форми двуспрямованої стрілки, далі слідує, утримуючи натиснутою ліву кнопку миші, переміщати її в потрібному напрямку, збільшуючи або зменшуючи діаграму. Якщо ж покажчик миші просто помістити усередину великого квадрата, то біля покажчика з'явиться великий хрест із стрілочок. У цьому випадку переміщення миші при натиснутій лівій кнопці змінює місце положення діаграми на листі.

*Зняти виділення* діаграми – клацнути за межами контуру діаграми, що обрмовує, і останній зникне.

Редагування діаграми виконується шляхом виклику контекстного меню при клацанні по частини малюнка, редагується.

#### 4.1.5 Друк робочого листа

Для підготовки до друку робочого листа варто включити принтер і виконати попередній перегляд друку за допомогою відповідної кнопки  або через:

меню **Файл** → команда **Предварительный просмотр**

При цьому можна змінити розміри документа при його розміщенні на друкованому аркуші (збільшити або зменшити), виконати вирівнювання відносно полів, вибрати орієнтацію паперу, створити колонтитули й ін. – кнопки **Поля** й **Страница...** у вікні перегляду.

Якщо таблиця готова до друку, то:

меню **Файл** → команда **Печать**, або кнопка **Печать** у вікні **Прогляд печатки**. На принтері буде роздрукована таблиця на одному або декількох аркушах.

**Приклад 4.1.** – Створення, формування й заповнення документа.

1. На **Листі 1** створити форму документа «Виконання плану товарообігу роздрібними підприємствами» (рисунок 4.7).

	A	B	C	D	E	F	G	H
1	<b>Виконання плану товарообігу роздрібними підприємствами</b>							
2								
3	Найменування підприємств	Фактично за минулий рік	Звітний рік			Відхилення		У% до минулого року
4			план	факт	% відхилення	від плану	від минулого року	
5	1	2	3	4	5	6	7	8
6	Магазин "Олимп"	25000	26000	27000	*	*	*	*
7	Магазин "Динамо"	24000	24500	25000	*	*	*	*
8	Магазин "Руга"	27000	27000	27300	*	*	*	*
9	Магазин "Mag"	17000	20000	20600	*	*	*	*
10	Усього:	*	*	*	*	*	*	*

Рисунок 4.7 – Форма документа «Виконання плану товарообігу роздрібними підприємствами»

В чарунках із символом “\*” необхідно розмістити розрахункові формули:

$$\text{гр.5} = (\text{гр.4} * 100) / \text{гр.3}$$

$$\text{гр.6} = \text{гр.4} - \text{гр.3}$$

$$\text{гр.7} = \text{гр.4} - \text{гр.2}$$

$$\text{гр.8} = (\text{гр.4} * 100) / \text{гр.2}$$

2. Віформатувати створену таблицю аналогічно наведеної у завданні таблиці.
3. Зберегти порожню форму документа під ім'ям **Форма 1**.
4. Створити список, що містить найменування підприємств.
5. Виконати автозаповнення елементами створеного списку чарунок графі ”**Найменування підприємств**”.
6. Заповнити документ конкретними вихідними даними.
7. Підготувати документ до друку в режимі попереднього перегляду:

- а). змінити орієнтацію паперу, масштаб для розміщення, виконати центрування на сторінці;
- б). створити нижній колонтитул, помістивши в ньому дату створення документа й своє прізвище.
8. Роздрукувати документ.
9. Скопіювати документ на **Лист 2**. Перейменувати **Лист 2**: привласнити йому ім'я **Режим формул**.
10. Установити *режим виводу формул*.
11. Зменшити ширину стовпців. Роздрукувати документ із формулами. Повернутися на **Лист 1**.
12. Створити на поточному листі *кругову діаграму порівняння фактичного товарообігу за звітний рік*.
13. Створити на окремому листі *гістограму порівняння даних фактичного товарообігу за минулий і звітний роки*.
14. Віформатувати створені діаграми за своїм розсудом (цвіт, шрифт, фон й т.д.)

Алгоритм формування документа представлений у Таблиці 4.3, а створення діаграм – у Таблиці 4.4.

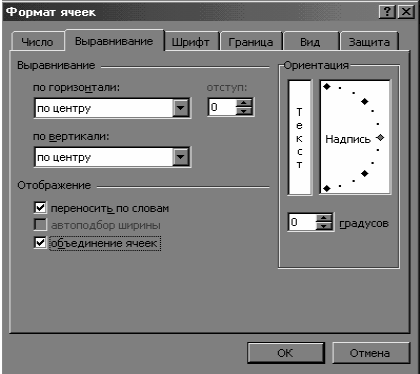

Таблиця 4.3. – Алгоритм формування документа

№ п/п	Актив-на чарунка	Операція	Спосіб виконання
1	2	3	4
1.	Весь лист	Установити шрифт для всіх чарунок таблиці	1). Виділити весь лист, клацнувши на кнопці <b>Виділити лист</b> . 2). Виконати: п. м. <b>Формат</b> → к. Ячейки → вкладка <b>Шрифт</b> . Вибрати шрифт <b>Times New Roman Cyr</b> , розмір шрифту <b>12</b> пунктів
2.	A1	Увести назву документа	Набрати на клавіатурі текст: Виконання плану товарообігу роздрібно-ми підприємствами і нажати клавішу <b>[Enter]</b> (Надалі – ↵)
3.	A3	Увести текст	Найменування підприємств ↵
4.	B3	Увести текст	Фактично за минулий рік ↵
5.	З3	Увести текст	Звітний рік ↵
6.	F3	Увести текст	Відхилення ↵



## Продовження таблиці 4.3.

1	2	3	4
7.	H3	Увести текст	В % до минулого року ↵
8.	34	Увести текст	план ↵
9.	D4	Увести текст	факт ↵
10.	E4	Увести текст	% виконання ↵
11.	F4	Увести текст	від плану ↵
12.	G4	Увести текст	від минулого року ↵
13.	A5	Увести текст	1 ↵
14.	B5	Увести текст	2 ↵
15.	35	Увести текст	3 ↵
16.	D5	Увести текст	4 ↵
17.	E5	Увести текст	5 ↵
18.	F5	Увести текст	6 ↵
19.	G5	Увести текст	7 ↵
20.	H5	Увести текст	8 ↵
21.	A10	Увести текст	Разом: ↵
22.	A3:A4	<p>1). Виділити чарунки з A3 по A4</p> <p>2). Віформатувати уміст чарунок: об'єднати чарунки й розмістити текст назви графи документа по центру чарунок і в декілька рядків.</p>	<p>1). Клацнути лівою кнопкою миші в чарунці A3 і, утримуючи її натиснутою, поширити виділення на комірку A4.</p> <p>2). Виконати: п.м. <b>Формат</b> → к. <b>Ячейки</b> → вкладка <b>Вирівнювання</b>. Вибрати: а). у полі <b>Вирівнювання по горизонталі – по центру</b> б). у полі <b>Вирівнювання по вертикалі – по центру</b> в). установити прапорець – <i>переносить по словам</i> г). установити прапорець – <i>об'єднання ячеек</i></p>

Продовження таблиці 4.3.

1	2	3	4
			 <p>д). клацнути по кнопці <b>Ok</b></p>
23.	B3:B4	Аналогічно пункту 22	Аналогічно пункту 22
24.	C3:E3	1). Виділити чарунки з C3 по E3 2). Об'єднати чарунки й вирівняти назву підзаголовка граф 3, 4 і 5 документа по центру.	1). Спосіб виділення див. у пункті 22.  2). В панелі інструментів <b>Формати-</b> <b>рование</b> клацнути по кнопці:  – <i>об'єднить и поместить в центре.</i>
25.	F3:G3	Аналогічно пункту 24.	Аналогічно пункту 24.
26.	H3:H4	Аналогічно пункту 22.	Аналогічно пункту 22.
27.	C4:G4	1). Виділити чарунки з C4 по G4 2). Віформатувати уміст чарунок: розмістити текст назв граф документа по центру чарунок і в кількя рядків.	1). Способ выделения см. в пункте 22. 2). Виконати: п.м. <b>Формат</b> → к. <b>Ячейки</b> → вкладка <b>Выравнивание</b> . Вибрати: а). у полі <b>Выравнивание по горизонтали</b> – <i>по центру</i> б). у полі <b>Выравнивание по вертикали</b> – <i>по центру</i> в). установити прапорець – <i>переносить по словам</i> г). клацнути по кнопці <b>Ok</b>

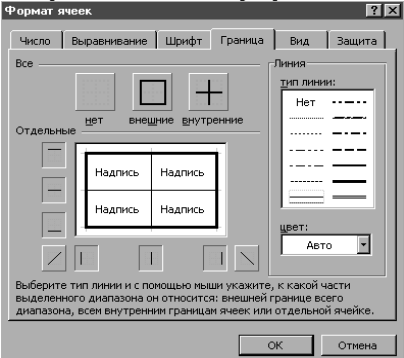
Продовження таблиці 4.3.

1	2	3	4
28.	A5:H5	1). Виділити чарунки з A5 по H5 2). Вирівняти номери граф документа по центру чарунк.	1). Способ выделения см. в пункте 22. 2). В панелі інструментів <b>Форматирование</b> клацнути по кнопці:  – <i>По центру</i> .
29.		Змінити ширину стовпців і висоту рядків для додавання необхідного вигляду шапці документа.	Установити покажчик миші на границю між маркерами стовпців (у рядку заголовків стовпців). Покажчик повинен прийняти форму хрестика зі стрілками. Нажати ліву кнопку миші й, не відпускаючи її, змінити ширину стовпців. Аналогічним образом змінити висоту рядків.
30.	A1:H1	1). Виділити чарунки з A1 по H1 2). Вирівняти назву документа по центру 3). Збільшити розмір шрифту	1). Способ выделения см. в пункте 22. 2). В панелі інструментів <b>Форматирование</b> клацнути по кнопці:  – <i>объединить и поместить в центре</i> . 3). Виконати: п.м. <b>Формат</b> → к. <b>Ячейки</b> – вкладка <b>Шрифт</b> . Вибрати розмір шрифту <b>14</b>
31.	E6	Ввести формулу для розрахунку по першому підприємству відсотка виконання плану товарообігу	=D6*100/C6.␣
32.	E6	Скопіювати формулу з чарунки E6 в чарунки з E7 по E10	Установити покажчик миші на маркер заповнення в чарунці E6 (маленький квадратик у нижньому правому куті чарунки). Покажчик прийме форму хрестика. Утримуючи натиснутою кнопку миші, виділити чарунки з E6 по E10, перетаскуючи маркер.
33.	F6	Ввести формулу для розрахунку	=D6-C6.␣

Продовження таблиці 4.3.

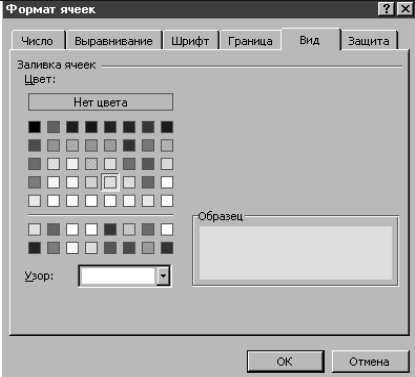
1	2	3	4
		по першому підприємству відхилення товарообігу від плану.	
34.	F6	Скопіювати формулу з чарунки F6 в чарунки з F7 по F10	Спосіб виконання дивися в пункті 32.
35.	G6	Ввести формулу для розрахунку по першому підприємству відхилення товарообігу від минулого року.	=D6-B6.␣
36.	G6	Скопіювати формулу з чарунки G6 в чарунки з G7 по G10	Спосіб виконання дивися в пункті 32.
37.	H6	Ввести формулу для розрахунку по першому підприємству виконання плану товарообігу у відсотках до минулого року.	=D6*100/B6.␣
38.	H6	Скопіювати формулу з чарунки H6 в чарунки з H7 по H10	Спосіб виконання дивися в пункті 32.
39.	B10	Увести формулу для автоматичного підсумовування даних по графі	В панелі інструментів <b>Стандартная</b> клацнути по кнопці $\Sigma$ – <i>Автосумма</i> . Клацнути в чарунці B6 і поширити пунктирну рамку, що бігає навколо неї,

Продовження таблиці 4.3.


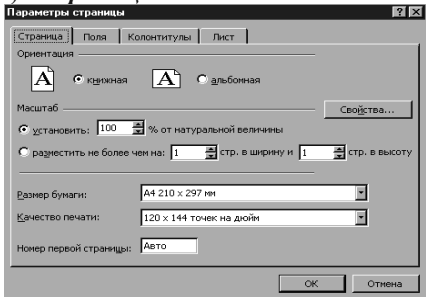
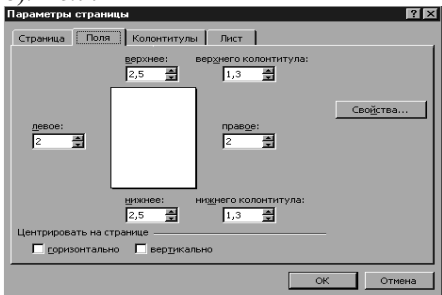
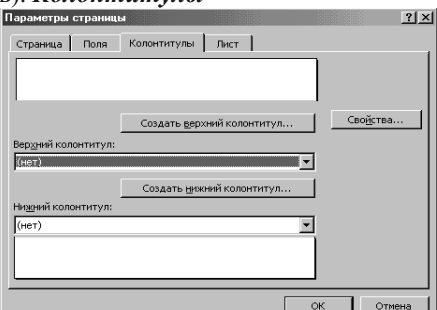
1	2	3	4
		<p><b>Фактично за минулий рік,</b> використовуючи математичну функцію “Автосумма”</p>	<p>на чарунки з B6 по B9. В чарунці B10 повинна з'явитися формула =СУММ(B6:B9). Для завершення її введення нажати клавішу ↵</p>
40.	B10	Скопіювати формулу з чарунки B10 в чарунки C10 і D10	Установити покажчик миші на маркер заповнення в чарунці B10 і втримуючи натиснутою кнопку миші, виділити чарунки з B10 по D10 перетаскуючи маркер.
41.	A3:H10	<p>1). Виділити табличну частину документа. 2). Створити рамку навколо таблиці (лінії товще) і сітку усередині неї (лінії тонше).</p>	<p>1). Клацнути лівою кнопкою миші в чарунці A3, потім натиснути клавішу [Shift] і, не відпускаючи її, клацнути в чарунці H10. 2). Виконати: п.м. <b>Формат</b> → к. <b>Ячейки</b> → вкладка <b>Граница</b> а). клацанням миші в полі <b>Тип лінії</b> вибрати товщину лінії для рамки, а потім клацнути по кнопці <b>Внешние</b> б). клацанням миші в полі <b>Тип лінії</b> вибрати товщину лінії для сітки, а потім клацнути по кнопці <b>Внутренние</b>.</p>  <p>в). Клацнути по кнопці <b>Ok</b>.</p>
42.	A3:H5	<p>1). Виділити чарунки з A3 по H5 2). Змінити колір в чарун-</p>	<p>1). Спосіб виділення див. У пункті 41. 2). Виконати: п. М. <b>Формат</b> → к. <b>Ячейки</b> – вкладка <b>Вид</b> → клацанням миші вибрати колір заливки.</p>



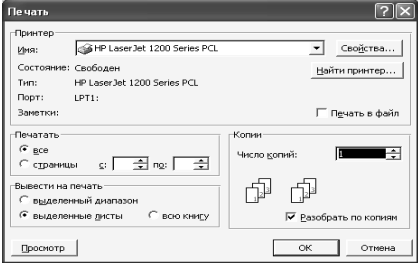
Продовження таблиці 4.3.

1	2	3	4
		ках таблиці, відведених під шапку документа.	 <p>3). Клацнути по кнопці <b>Ок</b>.</p>
43.	A10:H10	<p>1). Виділити чарунки з A10 по H10</p> <p>2). Змінити колір в чарунках таблиці, відведених під підсумкові дані</p>	<p>1). Спосіб виділення див. У пункті 41.</p> <p>2). Виконати: п. М. <b>Формат</b> → к. <b>Ячейки</b> → вкладка <b>Вид</b> → клацанням миші вибрати колір заливки.</p> <p>3). Клацнути по кнопці <b>Ок</b>.</p>
44.		Зберегти робочу книгу з порожньою формою документа під ім'ям <b>Форма 1</b>	<p>Виконати: п. М. <b>Файл</b> → к. <b>Сохранить</b> → ввести ім'я файла <b>Форма 1</b> → клацнути по кнопці <b>Сохранить</b></p>
45.	A6	Створити список, що містить найменування підприємств.	<p>Виконати: п.м. <b>Сервис</b> → к. <b>Параметры</b> → вкладка <b>Списки</b> → у вікно <b>Элементы списка</b> ввести список, що містить найменування підприємств → кнопка <b>Добавить</b> → <b>Ок</b></p>
46.	A6	Виконати автозаповнення елементами створеного списку чарунк з A6 по A9	<p>1). Ввести в чарунку A6 найменування одного з підприємств створеного списку.</p> <p>2). Установити покажчик миші на маркер заповнення в чарунці A6 і перетягнути його на чарунки з A6 по A9.</p>
47.		Заповнити документ конкретними даними	Заповнити всі чарунки документа конкретними даними з використанням при наборі дробових чисел десяткової коми на цифровій клавіатурі.


Продовження таблиці 4.3.

1	2	3	4
48.		<p>Підготувати документ до друку в режимі попереднього перегляду: змінити орієнтацію паперу, масштаб для розміщення, виконати центрування на сторінці; створити нижній колонтитул, помістивши в ньому дату створення документа й своє прізвище.</p>	<p>Виконати: п. м.Файл → к.Предварительный просмотр або кнопка  – Предварительный просмотр в панелі інструментів Стандартная.</p> <p>Для установки необхідних параметрів скористатися у вікні попереднього перегляду кнопкою <b>Страница</b>, а потім вкладками:</p> <p>а). <b>Страница</b></p>  <p>б). <b>Поля</b></p>  <p>в). <b>Колонтитулы</b></p> 

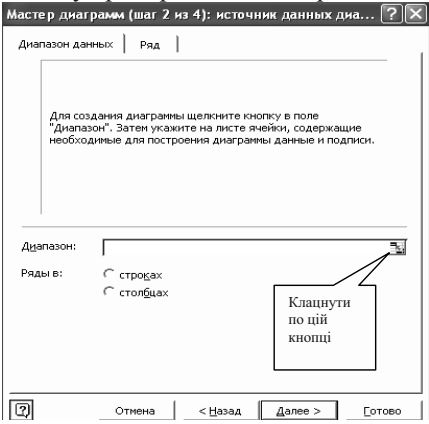
Продовження таблиці 4.3.

1	2	3	4
			Клацнути по кнопці <b>Ок</b> , а потім по кнопці <b>Закричь</b> для виходу з режиму попереднього перегляду.
49.		Роздрукувати документ, попередньо включивши принтер і вставивши в нього папір.	<p>Виконати: п.м. <b>Файл</b> → к. <b>Печать</b> → перемикач <b>Все</b> → кнопка <b>Ок</b>.</p>  <p>Роздрукований документ повинен бути аналогічний документу представленою на рисунку 4.8</p>
50.		Скопіювати документ на <b>Лист 2</b>	<p>1). Клацнути лівою кнопкою миші в чарунці A1, потім натиснути клавішу <b>[Shift]</b> і, не відпускаючи її, клацнути в чарунці H10. 2). Виконати: п.м. <b>Правка</b> → к. <b>Копировать</b> → клацнути мишкою по ярлику <b>Лист 2</b> → на <b>Листе 2</b> клацнути мишкою в чарунці A1 → п.м. <b>Правка</b> → к. <b>Вставить</b> → <b>Enter</b></p>
51.		Переименувати <b>Лист 2</b> : присвоїти йому ім'я <b>Режим формул</b> .	Виконати: п.м. <b>Формат</b> → к. <b>Лист</b> → к. <b>Переименовать</b> → ввести нове ім'я листа <b>Режим формул</b> (старе буде видалено)
52.		Установити режим виводу формул в чарунках документа	Виконати: п. м. <b>Сервис</b> → к. <b>Параметры</b> → вкл. <b>Вид</b> → установити прапорець <b>Формулы</b> → <b>Ок</b>
53.		Роздрукувати документ із формулами.	Як у пунктах 29, 48, 49. Роздрукований документ повинен бути аналогічний документу представленою на рисунку 4.9

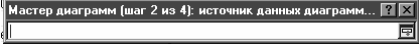
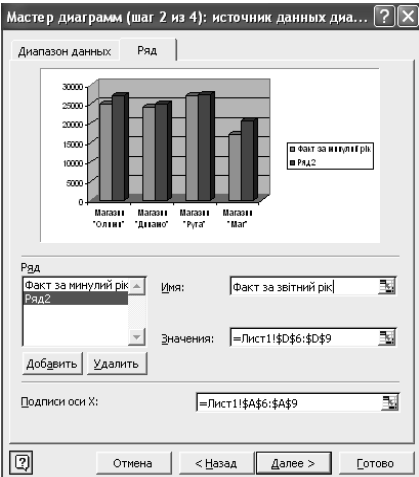
Таблиця 4.4. – Побудова діаграм

№ п/п	Операція	Спосіб виконання
1	2	3
<b>Побудова діаграми «Аналіз фактичного товарообігу роздрібних підприємств за звітний рік»</b>		
1.	Виділити діапазон даних, необхідних <b>Майстрові діаграм</b> для побудови діаграми (графи <b>Найменування підприємств</b> і <b>Факт</b> ). Брати тільки заповнені рядки.	1). Виділити чарунки з А6 по А9 2). Нажати клавішу <b>[Ctrl]</b> і, утримуючи її натиснутою, виділити чарунки з D6 по D9.
2.	Визвати <b>Майстер діаграм</b> .	Клацнути по інструменту  – <b>Майстер діаграм</b> в панелі інструментів <b>Стандартная</b> або виконати команду: п.м. <b>Вставка</b> → к. <b>Діаграма</b> .
3.	У вікні: <b>Майстер діаграм (шаг1 из 4): тип діаграммы</b> вибрати: 1). Тип діаграми 2). Вид діаграми	Вкладка <b>Стандартные</b> → <b>Тип</b> – <b>Круговая</b> → <b>Вид</b> – <b>объемный</b> варіант розрізаної кругової діаграми (середня в нижньому ряді) → клацнути по кнопці <b>Далее</b> (див. рисунок 4.3)
4.	У вікні: <b>Майстер діаграм (шаг2 из 4): источник данных диаграммы</b> необхідно виділити діапазон даних для діаграми. Тому що це було зроблено в пункті 1, перейти до наступного кроку.	Клацнути по кнопці <b>Далее</b> (див. рисунок 4.4)
5.	У вікні: <b>Майстер діаграм (шаг3 из 4): параметры диаграммы</b> : 1). Увести назву діаграми.  2). Вибрати місце для розміщення легенди 3). Установити режим виводу на діаграму числових значень вихідних даних	Виконати: 1). Вкладка <b>Заголовки</b> → поле <b>Название диаграммы</b> → надрукувати назву діаграми: <b>Аналіз фактичного товарообігу роздрібних підприємств за звітний рік</b> 2). Вкладка <b>Легенда</b> → установити перемикач <b>Справа</b> 3). Вкладка <b>Подписи данных</b> → установити перемикач <b>Значение</b> → клацнути по кнопці <b>Далее</b> (див. рисунок 4.5)

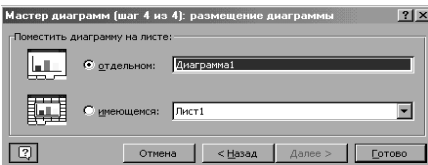
Продовження таблиці 4.4.

1	2	3
6.	У вікні: <b>Мастер діаграм (шаг 3 из 4): размещение диаграммы</b> вибрати спосіб розміщення діаграми на одному листі з документом	Установити перемикач <b>на наявному листі</b> → клацнути по кнопці <b>Готово</b>
7.	На робочому листі визначити: 1). Місце розташування діаграми.  2). Розмір діаграми.	1). Клацанням лівої кнопки миші виділити діаграму (навколо неї повинна з'явитися рамка з 8-ю чорними квадратами) і перемістити на листі так, щоб її верхній лівий кут розмістився в чарунці A12. 2). Змінити розмір діаграми, перетаскуючи за допомогою миші за чорні квадратики її границі. Створена діаграма представлена на рисунку 4.10
<b>Побудова гистограми «Порівняння даних фактичного товарообігу за минулий і звітний роки»</b>		
8.	Визвати майстер діаграм	Див. пункт 2
9.	<b>Шаг 1 мастера диаграмм:</b> вибрати тип і вид діаграми	Вкладка <b>Стандартные</b> → <b>Тун</b> – Гистограма → <b>Вид</b> – Объемный вариант обычной гистограммы → кнопка <b>Далее</b>
10.	<b>Шаг 2 мастера диаграмм:</b> 1). Указати діапазони чарунок для побудови гистограми.	1). Чарунки можна виділяти до кліку майстра діаграм, як у п.1 при побудові попередньої діаграми або на 2-му кроці роботи Майстра: 

Продовження таблиці 4.4.

1	2	3
	<p>2). Увести текст легенди.</p>	<p>Клацнути по зазначеній у винесенні кнопці перетворення вікна. Вікно майстра трансформується в один рядок:</p>  <p>а). виділити чарунки A6:A9          б). натиснути клавішу [Ctrl] і, утримуючи її натиснутою, виділити чарунки B6:B9 і D6:D9</p> <p>Клацнути по кнопці перетворення вікна в трансформованому вікні майстра діаграм і відновити його розміри.</p> <p>2). Клацнути по вкладці <b>Ряд</b> → у полі <b>Ряд</b> клацнути (виділити) по тексту <b>Ряд1</b> → клацнути в порожньому полі <b>Имя</b> і написати в ньому <b>Факт за минулий рік</b> → у полі <b>Ряд</b> клацнути (виділити) по тексту <b>Ряд2</b> → Клацнути в порожньому полі <b>Имя</b> і написати в ньому <b>Факт за звітний рік</b> ;</p>  <p>4). Клацнути по кнопці <b>Далее</b>.</p>

Продовження таблиці 4.4.

1	2	3
11.	<b>Шаг 3 мастера диаграмм:</b> Увести назву діаграми	Вкладка <b>Заголовки</b> → поле <b>Название диаграммы</b> → надрукувати назву діаграми <b>Порівняння даних фактичного товаро-обігу за минулий і звітний роки</b>
12.	<b>Шаг 4 мастера диаграмм:</b> Вибрати спосіб розміщення діаграми на окремому листі.	 <p>Створена діаграма представлена на рисунку 4.11</p>

Виконання плану товарообігу роздрібними підприємствами							
Найменування підприємств	Фактично за минулий рік	Звітний рік			Відхилення		У % до минулого року
		план	факт	% відхилення	від плану	від минулого року	
1	2	3	4	5	6	7	8
Магазин "Олімп"	25000	26000	27000	103,8461538	1000	2000	108
Магазин "Динамо"	24000	24500	25000	102,0408163	500	1000	104,1666667
Магазин "Рута"	27000	27000	27300	101,1111111	300	300	101,1111111
Магазин "Маг"	17000	20000	20600	103	600	3600	121,1764706
Усього:	93000	97500	99900	102,4615385	2400	6900	107,4193548

28.03.2005 Кононов В.А.

Рисунок 4.8 – Документ «Виконання плану товарообігу роздрібними підприємствами» (підготовлений до друку)

Виконання плану товарообігу роздрібними підприємствами							
Найменування підприємств	Фактично за минулий рік	Звітний рік			Відхилення		
		план	факт	% відхилення	від плану	від минулого року	
1	2	3	4	5	6	7	
Магазин "Олімп"	25000	26000	27000	=D6*100/C6	=D6-C6	=D6-B6	
Магазин "Динамо"	24000	24500	25000	=D7*100/C7	=D7-C7	=D7-B7	
Магазин "Рута"	27000	27000	27300	=D8*100/C8	=D8-C8	=D8-B8	
Магазин "Маг"	17000	20000	20600	=D9*100/C9	=D9-C9	=D9-B9	
Усього:	=СУММ(B6:B9)	=СУММ(C6:C9)	=СУММ(D6:D9)	=D10*100/C10	=СУММ(F6:F9)	=СУММ(G6:G9)	

Рисунок 4.9 – Документ «Виконання плану товарообігу роздрібними підприємствами» (у режимі виводу формул)



Рисунок 4.10 – Діаграма «Аналіз фактичного товарообігу роздрібних підприємств за звітний рік»

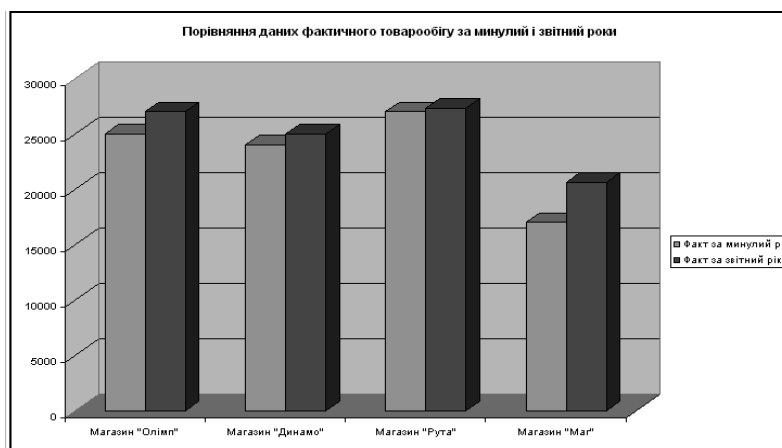


Рисунок 4.11 – Діаграма «Порівняння даних фактичного товарообігу за минулий і звітний роки»

Розглянуті прийоми побудови таблиць і діаграм є основою для знайомства з багатьма можливостями *Excel* по обробці даних, деякі з них, які не потребують підготовки в області фахових дисциплін будуть розглянуті в параграфах 4.2 – 4.4.



## 4.2 Використання формул

Формули – це душа і серце робочого листа табличного процесора, їхнє використання є відмінною рисою електронних таблиць від таблиць, утворюваних іншими додатками.

Загальні правила по створенню найпростіших формул були зазначені в параграфі 4.1. У даному параграфі будуть розглянуті більш серйозні питання їхньої побудови, без котрих неможливо вирішити практичні задачі.

### 4.2.1 Типи посилань у формулах

Посилання (адреса) є ідентифікатором чарунки або групи чарунок у книзі. Значення формули залежить від умісту чарунок, на які вказують посилання, і воно змінюється при зміні вмісту цих чарунок.

У попередньому прикладі були використані відносні посилання – наприклад, A1 або D2:E8. Цей тип посилань ґрунтується на положенні чарунки щодо чарунки, у яку вводиться формула, – «на два рядки вище» і «на три стовпчики правіше». Такий тип посилань широко використовується в *Excel* при звертанні до чарунок на тому ж листі, на інші листи в тій же або інших книгах. Щоб посилатися на чарунки іншого листа тієї ж книги, необхідно при наборі формули (введення формули починається зі знака рівності) клацнути по ярлику потрібного листа і виділити чарунку, потім ставити потрібний знак операції і таким само образом посилатися на будь-яку чарунку потрібного листа книги. Приклад такої формули: =Лист2!C9 + Лист3!E9.

Аналогічним образом можна посилатися і на чарунки розташовані в іншій книзі. Такі посилання називаються зовнішніми. Наприклад, щоб у чарунку A1 листа Лист2 книги *Книга1* увести посилання на чарунку A2 листа Лист3 книги *Книга2* варто виконати такі дії:

1. Відчинити *Книга1* – відкривається при завантаженні *Excel*.
2. Створити нову книгу *Книга2* – кнопка **Создать** на панелі інструментів **Стандартная**.

3. Виділити чарунку A1 листа Лист1 книги *Книга1*:  
Для Microsoft Office 97 – м. **Окно** → у нижній частині меню клацнути по назві *Книга1* → клацнути по ярличку **Лист2** → клацнути по чарунці **A1**.

Для Microsoft Office 2000 – клацнути по значку *Книга1* на панелі задач → клацнути по ярличку **Лист2** → клацнути по чарунці **A1** → **[Enter]**.

4. Увести знак рівності, а потім викликати книгу *Книга2* аналогічно діям п. 3 → клацнути по ярличку **Лист3** → клацнути по чарунці **A2** → **[Enter]**.

У рядку формул перед натисканням **[Enter]** буде таке:  $=[\text{Книга2}]\text{Лист3}\$A\$2$

Як видно з запису, частина посилання, що вказує книгу, замкнута у квадратні скобки, а саме посилання на чарунку записане в абсолютному виді. **Абсолютне посилання** використовується для указівки фіксованого положення чарунки в таблиці. У нашому прикладі чарунка знаходиться в стовпчику A рядка 2. При копіюванні абсолютного посилання її зміст не змінюється, тобто залишиться  $\$A\$2$ .

При необхідності можна використовувати **змішані посилання** – комбінацію абсолютного й відносного посилань. Наприклад,  $=\$A2$  – фіксований стовпчик A і відносний номер рядка, тобто при копіюванні він буде змінюватися.  $=A\$2$  – координата стовпчика відносна, а рядка – абсолютна.

Абсолютні й змішані посилання особливо корисні при копіюванні формули з одного місця листа в інше.

За допомогою натискання на клавішу **[F4]** можна швидко змінити тип посилання. Однократне натискання відносно посилання перетворить в абсолютне, а два наступні натискання дають можливість одержати змішані посилання.

**Приклад 4.2.** Скласти таблицю розрахунку щомісячних відсотків на вкладені суми в банк на декілька місяців, якщо відомі суми вкладення і річні банківські ставки.

Рішення подане на рисунку 4.12.

	A	B	C
1	<b>Розрахунок щомісячних відсотків</b>		
2	Сума внеску	1000	2000
3			
4	Річн. % ставка	сума %	сума %
5	0,07	$=\$B\$2*A5/12$	$=\$C\$2*A5/12$
6	0,075	$=\$B\$2*A6/12$	$=\$C\$2*A6/12$
7	0,08	$=\$B\$2*A7/12$	$=\$C\$2*A7/12$

Рисунок 4.12 – Абсолютні й відносні посилання у формулах

## 4.2.2 Загальна характеристика умонтованих функцій Excel

Функція – це заздалегідь визначена формула, що оперує з одним або декількома значеннями і повертає значення. Одна з функцій уже використовувалася в прикладі 6.1 по створенню приймального акта – СУММ. Її запис виконувався за допомогою кнопки **Автосуммирование** на панелі інструментів **Стандартная**. Але це далеко не все.

Пакет Microsoft *Excel* має потужні можливості по використанню умонтованих функцій. *Excel* має більш 300 умонтованих функцій, що виконують широкий спектр різноманітних обчислень. Всі функції, об'єднані в такі категорії:

- 1) фінансові – функції, що виконують фінансові обчислення;
- 2) дата і час – функції, що дозволяють представляти дату й виді чисел і виконувати операції над ними;
- 3) математичні й тригонометричні – включає математичні, логарифмічні і тригонометричні функції;
- 4) інженерні – функції для роботи з комплексними числами, для перетворення між 10-; 16-; 8-; і 2-вою системами числення, різноманітні форми функції Бесселя;
- 5) статистичні – функції для аналізу статистичних даних;
- 6) перегляду й посилання – функції дозволяють “переглядати” інформацію, що зберігається в списку або таблиці, а також обробляти посилання;
- 7) текстові – перетворюють текстові значення в числа, числові значення в рядки символів, дозволяють виконувати різноманітні операції над рядками символів;
- 8) логічні – більшість функцій використовують логічні вирази для визначення істинності заданої умови;
- 9) визначені користувачем.

Для одержання більш детальної інформації варто клацнути по кнопці **Справка**. Для *Excel 7.0* додаткову інформацію про функції можна одержати, виконавши дії:

- 1) у меню **Справка** вибрати команду **Вызов справки**;

2) клацнути по вкладці **Содержание**, а потім по значку книги **Справочник**;

3) клацнути по значку книги **Функции листа**, потім по значку книги для категорії функцій, які ви хочете подивитися;

4) двічі клацнути по значку **?** для функції, що вас цікавить.

Для Office 2000 у змісті довідки варто вибрати пункт **Создание и проверка формул у книгах**.

Функції складаються з 2-х частин: імені функції й одного або декількох аргументів. Функція завжди починається зі знака « = » , а аргументи містяться у круглих скобках. У якості аргументів можуть використовуватися константи, посилання на чарунки, діапазони чарунок, функції, імена діапазонів, арифметичні вирази, розділені «;» .

Перелік функцій значно поповнюється при підключенні надбудови **Пакет анализа**, що доступний при кожному запуску *Excel*, якщо виконана повна установка пакета меню **Сервис**, команда **Анализ данных**. Якщо такої команди немає, необхідно:

1. Вибрати меню **Сервис**, команду **Надстройки**;

2. Вибрати **Пакет анализа -VBA** у списку надбудов і **ОК**.

Якщо пакет аналізу відсутній у списку надбудов, то його можна встановити двічі клацнути по значку **Установка Microsoft Excel**.

Для виклику категорій і самих функцій можна користуватися **Мастером функций** – кнопка зі значком **fx** на панелі інструментів або пункт меню **Вставка** команда **Функции**, або набір із клавіатури. При виборі мишею функції у вікні майстра функцій дається опис її призначення (рисунок 4.13)

Вибір категорій виконується клацанням у лівій частині вікна, а праворуч при цьому показуються упорядковані за алфавітом функції цієї категорії. Виділення будь-якої функції викликає появу в нижній частині вікна її стислої характеристики. При клацанні по кнопці **ОК** викликається друге вікно **Мастера функций** (рисунок 4.14), яке допомагає користувачу заповнити аргументи.

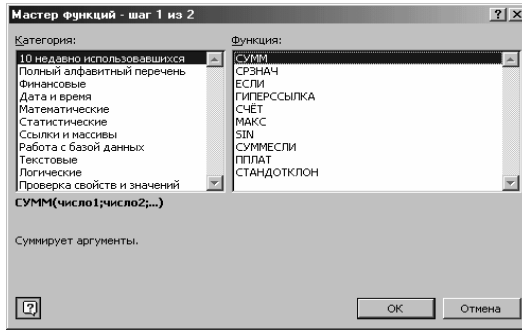


Рисунок 4.13 – Вікно Мастера функций – крок 1

Після знака дорівнює праворуч нижче аргументів можна побачити результат обчислень, не покидаючи вікна.

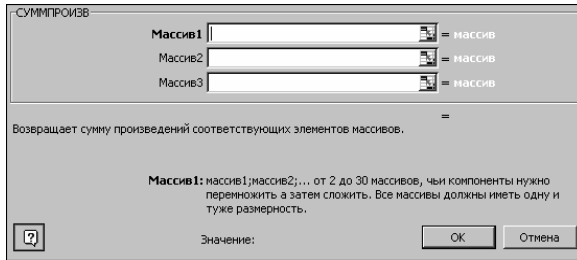


Рисунок 4.14 – Вікно Мастера функций – крок 2

Якщо формула містить декілька функцій, то другу і наступні функції можна викликати після знака операції, використовуючи лівий список, що випадає, у рядку формул (рисунок 4.15).

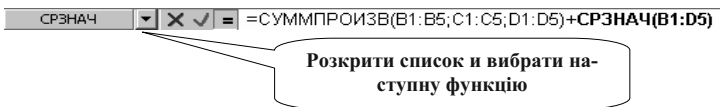


Рисунок 4.15 – Список для включення додаткових функцій у формулу

Приклад 4.3. Створити форму документа «Акт інвентаризації» (рисунок 4.16).

Примітка – При обчисленні відхилення використовувати логічну функцію **ЯКЩО**:

**ЯКЩО** (логічне\_вираження;значення\_якщо\_істина;значення\_если\_неправда)

При записі логічних виражень використовуються операції відносин –  $A5-U7>C5$ ,  $D11\leq 20$ ,  $B7=D7$  і т.д. Якщо логічне вираження приймає значення «істина» (виконується), те в чарунку з формулою заноситься значення, що стоїть відразу після нього, у противному випадку (не виконується – значення «неправда») записується «значення\_якщо\_неправда».

	A	B	C	D	E	F	G	H	I	J	K
1	<b>Акт інвентаризації</b>										
2	від										
3	по коморі фірми "Здорове харчування"										
4	Код продукту	Найменування продукту	Один вим.	Сорт	Ціна	Кількість		Сумма		Відхилення	
фактич.						по обліковим даним	фактич.	по обліковим даним	+	-	
5											
6	1	A			5	10	10	50	50		
7	2	B			5	10	9	50	45		5
8	3	C			5	9	10	45	50	5	
9								0	0		
10								0	0		
11		Разом						145	145	5	5




Рисунок 4.16 – Форма документа «Акт інвентаризації»

Алгоритм формування документа представлень у Таблиці 4.5.




Таблиця 4.5. – Алгоритм формування акта інвентаризації

№ п/п	Активна чарунка	Операція	Спосіб виконання
1	2	3	4
1.	A1	Увести назву документа	Набрати на клавіатурі текст: Акт інвентаризації і натиснути клавішу [↵]
3.	A2	Увести текст	від ↵
4.	A3	Увести текст	по коморі фірми "Здорове харчування" ↵
5.	A4	Увести текст	Код продукту ↵
6.	У4	Увести текст	Найменування продукту ↵
7.	З4	Увести текст	Один. вим. ↵
8.	D4	Увести текст	Сорт ↵
9.	E4	Увести текст	Ціна ↵
10.	F4	Увести текст	Кількість ↵
11.	F5	Увести текст	Фактич.
12.	G5	Увести текст	По обліковим даним ↵
13.	H4	Увести текст	Сума ↵

Продовження таблиці 4.5.

1	2	3	4
14.	H5	Увести текст	Фактич.↵
15.	I5	Увести текст	По обліковим даним ↵
16.	J4	Увести текст	Відхилення ↵
17.	J5	Увести текст	+ ↵
18.	K5	Увести текст	- ↵
19.	A4:E5	1). Виділити чарунки з A4 по E5 2) Відформатувати шапку документа в зазначених чарунках (див. форму документа)	Клацнути лівою кнопкою миші в чарунці A4 і, утримуючи її натиснутою, виділити чарунки з A4 по E5. Виконати команду: м. <b>Формат</b> → к. <b>Ячейки</b> → вкладка <b>Вирівнювання</b> → виставити прапорець <b>Переносить по словам</b> → у полі <b>Вирівнювання по горизонталі</b> установити <b>по центру</b> → у полі <b>Вирівнювання по вертикалі</b> установити <b>по центру</b> → <b>Ок</b>
20.	F4:G4	Об'єднати чарунки	Виділити діапазон F4:G4 → м. <b>Формат</b> → к. <b>Ячейки</b> → виставити прапорець <b>Об'єднати ячейки</b> → <b>ОК</b>
21.	H4:I4, а потім J4:K4	Об'єднати чарунки	Виконати дії аналогічні п.20
22.	F5:K5	Виділити чарунки з F5 по K5 и виконати форматування	Аналогічно п.19
23.	A4:A5	Об'єднати чарунки	Виділити діапазон A4:A5 → м. <b>Формат</b> → к. <b>Ячейки</b> → виставити прапорець <b>Об'єднати ячейки</b> → <b>ОК</b>
24.	B4:B5 ; C4:C5; D4:D5; E4:E5	Об'єднати чарунки в кожному діапазоні	Для кожного діапазону виконати дії аналогічні п. 23
25.	A11	Увести текст	Разом: ↵ → кнопка <b>вирівнять по правому краю</b> 
26.	A4:K11	Намалювати рамку	Виділити A4:K11 → за допомогою списку кнопки <b>граници</b> вибрати рамку з перетинанням усередині 
27.	A4:K5	Виділити жирною рамкою	Аналогічно п. 26, рамка – 
28.	A11:K11	Виділити жирною рамкою	Аналогічно п. 27

Продовження таблиці 4.5.

1	2	3	4
29.	H6	Занесення формули	Виділити чарунку H6 і ввести формулу =E6*F6 → перейти в I6
30.	I6	Занесення формули	Виділити чарунку I6 і ввести формулу =E6*G6 ↵
31.	J6	Занесення формули	<p>Виділити чарунку J6 → клацнути по кнопці  → виділити категорію <b>Логические</b> → виділити функцію <b>Если</b> → <b>ОК</b> → у полі <b>Логическое выражение</b> ввести I6-H6&gt;=0 → клацнути по полю <b>Значение_если_ложь</b> → ввести “ “ → клацнути по полю <b>Значение_если_истина</b> → в рядку формул у лівому полі вибрати ще раз функцію ЕСЛИ → у новому вікні у полі <b>Логическое выражение</b> ввести I6-H6&gt;0 → клацнути по полю <b>Значение_если_истина</b> → ввести I6-H6 → клацнути по полю <b>Значение_если_ложь</b> → ввести " " → <b>ОК</b> .</p> <p>В чарунці J6 повинна вийти формула: =ЕСЛИ(I6-H6&gt;=0;ЕСЛИ(I6-H6&gt;0;I6-H6;"");"")</p>
32.	K6	Занесення формули	<p>Виділити чарунку K6 → клацнути по кнопці  → виділити категорію <b>Логические</b> → виділити функцію <b>Если</b> → <b>ОК</b> → у полі <b>Логическое выражение</b> ввести I6-H6&lt;0 → клацнути по полю <b>Значение_если_истина</b> → ввести H6-I6 → клацнути по полю <b>Значение_если_ложь</b> → ввести “ “</p> <p>В чарунці K6 повинна вийти формула: =ЕСЛИ(I6-H6&lt;0;H6-I6;" ")</p>
33.	H6:K6	Копіювання формул в інші рядки	Виділити діапазон H6:K6 → покажчик миші поставити на маркер заповнення (приклад 4.1) → затиснути ліву кнопку миші і тягти до рядка «Разом:» → відпустити кнопку → зняти виділення.
34.	H11	Занесення суми по стовпцю	<p>Виділити чарунку H11 → клацнути по кнопці <b>Автосуммирование</b>  → перевірити діапазон → ↵</p> <p>В чарунці повинна вийти формула</p>



Продовження таблиці 4.5.

1	2	3	4
			<b>=СУММ(Н6:Н10)</b>
35.	Н11	Копіювання формули в чарунки Н11 – К11.	Виділити чарунку Н11 → покажчик миші поставити на маркер заповнення (приклад 4.1) → затиснути ліву кнопку миші і тягнути до К11 → кнопку відпустити → зняти виділення.
36.		Зберегти форму документа з ім'ям Акт інвентаризації.	М. <b>Файл</b> → к. <b>Сохранить</b> → дати ім'я <i>Акт інвентаризації</i> → указати потрібну папку → <b>ОК</b>

### 4.2.3 Формули з масивами

При створенні бланків документів у прикладах 4.1 (документ «*Виконання плану товарообігу роздрібними підприємствами*») і 4.3 («*Акт інвентаризації*») у стовпцях суми й відхилень використовувалися однотипні формули, що одержували шляхом копіювання з першої створеної. У випадках, коли потрібно маніпулювати деякою безліччю значень або одержувати безліч результатів по однотипних формулах, можна використовувати **формулу масиву**. Масив у даному випадку – дані розташовані або одержувані у визначеному діапазоні чарунок. Наприклад, у прийомному акті є діапазон цін, діапазон кількості й одержують діапазон сум.

Правила введення формул масиву:

1. Перед введенням формули масиву потрібно виділити чарунку або діапазон, що буде містити результати. Якщо повертається масив результатів, то діапазон їх повинен бути такого ж розміру й форми, як діапазони з вхідними даними.

2. Увести формулу, у якій аргументами є діапазони з вхідними даними. Діапазони вхідних даних повинні бути однієї розмірності. Якщо в якості одного з аргументів використовується константа, то *Excel* може розширити цей аргумент, щоб він відповідав іншому по розміру й формі.

Приклад.

Уведена формула  $\{=A1:B3*10\}$  буде еквівалентна формулі  $\{=A1:B3*\{10;10;10;10;10;10\}\}$ .

3. Натиснути клавіші **[Ctrl]+[Shift]+[Enter]** для фіксації уведення формули масиву – вона міститься у фігурні дужки. Не вводьте фігурні дужки самі, інакше Excel розцінить уведення як текстове значення.

4. У діапазоні масиву формул не можна змінювати, очищати, переміщати, вставляти й видаляти окремі чарунки. Вони розглядаються як єдине ціле.

5. Для зміни або очищення масиву формул варто виділити весь масив і активізувати рядок формул. (Фігурні дужки навколо формули зникнуть.) Тепер можна змінити або очистити формулу, а потім знову натиснути **[Ctrl]+[Shift]+[Enter]** .

6. Масив формул можна копіювати й переміщати аналогічно звичайному діапазону чарунок. Можна копіювати окремі чарунки діапазону масиву і вставляти їх в інші області робочого листа. Можна призначити різні формати окремим чарункам у масиві.

**Приклад 4.4.** – Відкрити документ «Виконання плану товарообігу роздрібними підприємствами» (приклад 4.1), скопіювати форму документа на **Лист3**, очистити графи 5, 6, 7, 8 і заповнити їх за допомогою формули масиву.

Алгоритм рішення приклада представлений у Таблиці 4.6.

Таблиця 4.6. – Алгоритм рішення приклада 4.4

№ п/п	Активна чарунка	Операція	Спосіб виконання
1	2	3	4
1.	A1:H10	Копіювання форми документа	Виділити чарунки A1: H10 → м. <b>Правка</b> → к. <b>Копировать</b> → клацнути по ярличкові Лист3 → м. <b>Правка</b> → к. <b>Вставить</b> . При необхідності відформатувати ширину граф.
2.	E6:E9	Очистити графу від формул	Виділити зазначений діапазон і натиснути клавішу <b>Delete</b> .
3.	E6:E9	Уведення масиву формул	Виділення не знімати. Увести формулу: =D6:D9*100/C6:C9 → натиснути <b>[Ctrl]+[Shift]+[Enter]</b>
4.	F6:F9	Очистити графу від формул	Виділити зазначений діапазон і натиснути клавішу <b>Delete</b> .
5.	F6:F9	Ввод массива формул	Виділення не знімати. Увести формулу: =D6:D9-C6:C9 → натиснути <b>[Ctrl]+[Shift]+[Enter]</b>

Продовження таблиці 4.6

1	2	3	4
6.	G6:G9	Очистити графу від формул	Виділити зазначений діапазон і натиснути клавішу <b>Delete</b> .
7.	G6:G9	Ввод массива формул	Виділення не знімати. Увести формулу: =D6:D9-B6:B9 → натиснути <b>[Ctrl]+[Shift]+[Enter]</b>
8.	H6:H9	Очистити графу від формул	Виділити зазначений діапазон і натиснути клавішу <b>Delete</b> .
9.	H6:H9	Ввод массива формул	Виділення не знімати. Увести формулу: =D6:D9*100/B6:B9 → натиснути <b>[Ctrl]+[Shift]+[Enter]</b> Результат у режимі формул показаний на рисунку 4.17.

**Приклад 4.5** – Відкрити файл із документом «*Акт інвентаризації*» (приклад 4.3), скопіювати форму документа на **Лист2**, очистити графі **Відхилення** і заповнити них за допомогою формул масиву.

Алгоритм виконання аналогічний прикладові 4.4. Фрагмент форми акта інвентаризації у режимі формул показаний на рисунку 4.18.

У розглянутих прикладах були використані одномірні масиви формул, але можна створювати двовимірні масиви, що містять кілька рядків і стовпців.

Виконання плану товаробігу роздрібними підприємствами							
Найменування підприємства	Фактично за минулий рік	Звітний рік			Відхилення		У % до минулого року
		план	факт	% відхилення	від плану	від минулого року	
1	2	3	4	5	6	7	8
Магазин "Стиль"	25000	26000	27000	=D6:D9*100/C6:C9	=D6:D9-C6:C9	=D6:D9-B6:B9	=D6:D9*100/B6:B9
Магазин "Дизайн"	24000	24500	25000	=D6:D9*100/C6:C9	=D6:D9-C6:C9	=D6:D9-B6:B9	=D6:D9*100/B6:B9
Магазин "Рука"	27000	27000	27300	=D6:D9*100/C6:C9	=D6:D9-C6:C9	=D6:D9-B6:B9	=D6:D9*100/B6:B9
Магазин "Мод"	17000	20000	20600	=D6:D9*100/C6:C9	=D6:D9-C6:C9	=D6:D9-B6:B9	=D6:D9*100/B6:B9
Усього:	=СУММ(B6:B9)	=СУММ(C6:C9)	=СУММ(D6:D9)	=D10*100/C10	=СУММ(F6:F9)	=СУММ(G6:G9)	=D10*100/B10

Рисунок 4.17 – Форма документа «Виконання плану товаробігу роздрібними підприємствами» із використанням масиву формулу режимі формул.

Відхилення	
+	-
=ЕСЛИ(Н6:Н11>=І6:І11;ЕСЛИ(Н6:Н11=І6:І11;" ";Н6:Н11-І6:І11);" ")	=ЕСЛИ(Н6:Н11<І6:І11;І6:І11-Н6:Н11;" ")
=ЕСЛИ(Н6:Н11>=І6:І11;ЕСЛИ(Н6:Н11=І6:І11;" ";Н6:Н11-І6:І11);" ")	=ЕСЛИ(Н6:Н11<І6:І11;І6:І11-Н6:Н11;" ")
=ЕСЛИ(Н6:Н11>=І6:І11;ЕСЛИ(Н6:Н11=І6:І11;" ";Н6:Н11-І6:І11);" ")	=ЕСЛИ(Н6:Н11<І6:І11;І6:І11-Н6:Н11;" ")
=ЕСЛИ(Н6:Н11>=І6:І11;ЕСЛИ(Н6:Н11=І6:І11;" ";Н6:Н11-І6:І11);" ")	=ЕСЛИ(Н6:Н11<І6:І11;І6:І11-Н6:Н11;" ")
=ЕСЛИ(Н6:Н11>=І6:І11;ЕСЛИ(Н6:Н11=І6:І11;" ";Н6:Н11-І6:І11);" ")	=ЕСЛИ(Н6:Н11<І6:І11;І6:І11-Н6:Н11;" ")
=ЕСЛИ(Н6:Н11>=І6:І11;ЕСЛИ(Н6:Н11=І6:І11;" ";Н6:Н11-І6:І11);" ")	=ЕСЛИ(Н6:Н11<І6:І11;І6:І11-Н6:Н11;" ")

Рисунок 4.18 – Форма акта інвентаризації із використанням масиву формул у режимі формул.

**Приклад 4.6.** Необхідно обчислити корінь квадратний 12 значень перемінних, розташованих у діапазоні А3:D5.

**Виконання.**

1. Виділити діапазон такого ж розміру і форми, як діапазон з вхідними даними – А7:D9.
2. Викликати майстер функцій → категорія **Математичні** → функція **КОРЕНЬ** → **ОК** → указати діапазон **А3:D5** → **ОК** → клацнути в рядку формул і натиснути **[Ctrl]+[Shift]+[Enter]** . Результат буде мати вигляд (рисунок 4.19):

	A	B	C	D
1				
2				
3	45	21	17	25
4	8	16	24	33
5	18	2	9	64
6				
7	6,708204	4,582576	4,123106	5
8	2,828427	4	4,898979	5,744563
9	4,242641	1,414214	3	8

	A	B	C	D
1				
2				
3	45	21	17	25
4	8	16	24	33
5	18	2	9	64
6				
7	=КОРЕНЬ(А3:D5)	=КОРЕНЬ(А3:D5)	=КОРЕНЬ(А3:D5)	=КОРЕНЬ(А3:D5)
8	=КОРЕНЬ(А3:D5)	=КОРЕНЬ(А3:D5)	=КОРЕНЬ(А3:D5)	=КОРЕНЬ(А3:D5)
9	=КОРЕНЬ(А3:D5)	=КОРЕНЬ(А3:D5)	=КОРЕНЬ(А3:D5)	=КОРЕНЬ(А3:D5)

Рисунок 4.19 – Приклад реалізації двовимірного масиву формул

### 4.2.4 Присвоєння імен чарункам і діапазнам

Не завжди зручно використовувати у формулах посилання на чарунки. При створенні великих таблиць або цілого ряду таблиць через досить короткий проміжок часу важко розібратися в структурі формул. Значно простіше це робити, якщо замість посилань використовуються імена. Чарункам і діапазнам чарунок можна призначати імена і потім

використовувати їх у формулах. На імена діапазонів можна посилатися в будь-якому місці будь-якої книги.

Найпростіший спосіб визначення імені – використання поля імені в рядку формул (рисунок 4.20):

клацнути по потрібній чарунці → клацнути по полю імені → Увести ім'я **Внесок\_2** → [Enter]. Замість посилання на чарунку тепер при її виділенні в полі імені буде виводитися **Внесок\_2**.

	А	В	С
1	Розрахунок щомісячних відсотків		
2	Сума внеску	1000	2000
3			

Рисунок 4.20 – Визначення імен за допомогою поля імені

Якщо ім'я визначене для діапазону осередків, то воно буде з'являтися в полі імені тільки при виділенні всього діапазону.

Правила присвоєння імен чарункам і діапазнам:

1. Ім'я повинне починатися з букви, або символів «/» або «\_».
2. В імені можуть використовуватися тільки букви, цифри й символи «/» і «\_».
3. Як імена можуть використовуватися одиночні букви за винятком **R** і **C**.

Як імена можна використовувати текст у сусідніх чарунках. Якщо цей текст знаходиться ліворуч або вище виділення, то *Excel* автоматично вставляє його в поле імені, в інших випадках користувач сам указує ім'я.

Приклад 4.7. – Привласнити ім'я діапазонаві чарунок із даними по річних процентних ставках.

Виконання. Виділити діапазон **A5:A7** → м. **Вставка** → к. **Имя** → п. **Присвоить** → у діалоговому вікні **Присвоение имени** з'явиться ім'я **Річн\_ставка** → **ОК**.

Приклад 4.8. – Привласнити імена діапазнам чарунок з обчисленими відсотками відповідно **Проц\_з\_1000** и **Проц\_з\_2000**.

Виконання. Виділити діапазон **B5:B7** → м. **Вставка** → к. **Имя** → п. **Присвоить** → у діалоговому вікні **Присвоение имени** стерти пропонуване ім'я **Сума** і написати **Проц\_з\_1000** → кнопка **Добавить** → клацнути у полі **Формула** унизу вікна **Присвоение имени** та виділити діапазон **C5:C7** → у полі імені написати **Проц\_з\_2000** → **ОК**.

Змінити імена або діапазони чарунок, яким привласнені імена можна через теж вікно **Присвоєння імени**. Імена, що виділені, можна видаляти за допомогою кнопки **Удалить**, замінити (видалити старі і ввести нові), або змінювати для них чарунки або діапазони в полі **Формула**.

**Зауваження** – при видаленні імені діапазону формули, що посілалися на це ім'я будуть повертати помилкове значення **#ИМЯ?**.

Для заміни посилань на чарунки на імена у вже існуючих формулах варто виконати:

м. **Вставка** → к. **Имя** → п. **Применить** → виділити при натиснутій клавіші **[Shift]** потрібні імена у списку імен → **ОК**.

При введенні нових формул на місці посилання на чарунки необхідно використовувати м. **Вставка** → к. **Имя** → п. **Вставить** → виділити потрібне ім'я → **ОК**.

#### 4.2.5 Робота з групою листів. Об'ємні формули

Досить часто в практиці доводиться узагальнювати дані однотипних документів. У цьому випадку форма документа та сама при різному змісті, наприклад, місячні звіти або бюджети відділень. Для створення таких документів *Excel* має засіб **групового редагування**. Можна згрупувати разом кілька листів у книзі і потім додати, відредагувати або відформатувати дані на всіх цих листах одночасно.

Виділення й угруповання аркушів можна виконати одним із перерахованих нижче методів.

1. Виділити ярличок першого листа в діапазоні суміжних листів, які потрібно згрупувати, і при натиснутій клавіші **[Shift]** клацнути по ярличку останнього листа в діапазоні.
2. Виділити ярличок першого листа і при натиснутій клавіші **[Ctrl]** послідовно клацнути по ярличках усіх листів, що потрібно включити в групу, незалежно від їхнього положення в книзі.
3. Команда **Выбрать все листы** в контекстному меню ярличків аркушів.

*Excel* позначає режим групового редагування, додаючи **[Группа]** у рядку заголовка книги.

**Зауваження** – Можна активізувати інші листі в групі, не виходячи з режиму групового редагування. Якщо клацнути по ярличку листа поза групою, *Excel* виходить з режиму групового редагування і видаляє

**[Група]** із рядка заголовка книги. Вихід із режиму групового редагування, якщо всі листи книги включені в групу – клацання по ярличку будь-якого неактивного листа.

Тепер, все, що буде вводиться на першому листі, як під копірку буде з'являтися на всіх листах групи. Будь-які зміни форматування, що зроблені на вкладках **Число, Выравнивание, Шрифт, Рамка, Вид** та **Защита** у вікні діалогу **Формат ячеек**, застосовуються одночасно до всіх членів групи. У режимі групового редагування команди **Строка, Столбец, Автоформат** та **Стиль** також застосуються до всіх листів у групі.

Усі команди з меню **Правка**, за винятком **Найти та Заменить**, застосовуються одночасно до всіх листів у групі.

Команди з меню **Вставка – Ячейки, Строка, Столбец** та **Функция** можна використовувати для одночасної вставки відповідних елементів в усі листи в групі.

У режимі групового редагування всі команди друку з меню **Файл** застосовуються до всіх листів групи.

Приклад 4.9. – Необхідно створити звіти для заповнення й обробки даних по безалкогольних напоях у розрізі матеріально-відповідальних осіб (5 чоловік) за тиждень за формою, представленою на рисунку 4.21:

	A	B	C	D	E	F	G	H	I	J	K	L
1	<b>Звіт про реалізації безалкогольних напоїв</b>											
2	за											
3	матеріально-відповідальна особа											
4												
5	Найменування товару	Залишок на початок	Прихід	Реалізація по датах							Залишок на кінець	Ціна
1				2	3	4	5	6	7			
6	Фанта	2	40	5	3	7	1	10	5	6	5	2,4
7	Кока-кола	2	40	5	5	6	5	10	5	6	0	3
8	Миринда	2	40	5	2	5	3	10	5	6	6	3,2
9	Байкал	2	40	5	3	4	7	10	5	6	2	2,5
10											0	
11											0	
12											0	
13											0	
14											0	
15	Итого:	8	160	20	13	22	16	40	20	24	13	

Рисунок 4.21 – Форма документа «Звіт про реалізації безалкогольних напоїв»

### Виконання:

1. Додати нові аркуші в книгу – клацнути правою кнопкою миші по ярличку останнього листа → **к. Добавить** → вкладка **Общие** → виділи-

ти значок **Лист** → **ОК** (з'явиться новий лист). Повторювати потрібну кількість разів.

2. Упорядкувати листи: перетягнути мишкою ярличок листа, що стоїть не один по одному, на відведене для нього місце.

3. Клацнути по ярличку **Лист1**, натиснути клавішу [**Shift**] і одночасно клацнути по ярличку **Лист5**.

4. На першому виділеному листі створити зазначену форму с формулами.

5. Зняти виділення з листів, клацнувши по ярличку **Лист6**.

6. У кожному листі заповнити прізвище матеріально-відповідальної особи.

Приклад 4.10. – Додати у сформовану книгу (приклад 4.9) звітні форми по двом новим реалізаторам.

Послідовність дій:

1. Виділити групу листів з 5 по 7.

2. На листі 5 виділити усю форму звіту, м. **Правка** → к. **Заполнить** → п. **По листам** → покажчик **Все** → **ОК**.

3. Зняти виділення з групи листів і відкоригувати прізвища на листах 6 і 7.

Примітка – Якщо звіт, що копіюється, був заповнений, → виділити групу нових листів (6 і 7) → виділити предметну частину листа 6 → натиснути клавішу [**Delete**].

При формуванні підсумкової таблиці по створених звітних формах можна використовувати *об'ємні формули*, що можуть підсумовувати дані відразу по декількох однойменних чарунках різних листів.

Приклад 4.11. – По звітах прикладів 4.9 – 4.10 створити підсумковий звіт за тиждень.

Виконання.

1. Додати в книгу ще один лист і перетягнути його в кінець книги.

2. Назвати новий лист **Тижневий звіт**: клацнути правою кнопкою миші по ярлику останнього листа → к. **Переименовать** → стерти назву **Лист8** и набрати **Тижневий звіт** → [**Enter**].

3. Скопіювати форму звіту: виділити на **Лист7** форму звіту (**A1:L15**) → інструментальна кнопка **Копировать** → перейти на **Тижневий звіт** → інструментальна кнопка **Вставить**. При необхідності відформатувати ширину граф.



4. На листі «Тижневий звіт» виділити чарунку **B7** → клацнути по кнопці **Автосуммирование** → клацнути по ярличку **Лист1** → при натиснутій клавіші **[Shift]** клацнути по ярличку **Лист7** → на листі **Лист1** виділити чарунку **B7** → **[Enter]**. Одержимо формулу =СУММ(Лист1:Лист7!B7)
5. Скопіювати формулу з чарунки **B7** у чарунки **C7:K7** (див. приклад 4.1).
6. Виділити чарунки **B7:K7** і скопіювати їх у чарунки **B8:K14**.

У результаті в кожному чарунку звітної форми буде занесена об'ємна формула, що буде підраховувати суму по відповідних днях тижня з усіх звітів реалізаторів.

Зауваження – Не можна створювати об'ємні масиви формул із використанням декількох листів книги.

При роботі з об'ємними формулами можна використовувати наступні функції: **СУММ**, **СЧЁТЗ**, **СРЗНАЧ**, **МАКС**, **МИН**, **ПРОИЗВЕД**, **СЧЁТ**, **СТАНДОТКЛОН**, **СТАНДОТКЛОНП**, **ДИСП**, **ДИСПР**.

### 4.3 Консолідація листів

При виконанні робіт із створення зведених звітів по однотипних детальних звітах, крім об'ємних формул, *Excel* дозволяє використовувати ще один досить простий і зручний прийом – консолідацію даних.

Команда **Консолідація** з меню **Данные** може об'єднати інформацію з вхідних листів (до 255 листів) в одному підсумковому листі. Вхідні листі можуть розташовуватися в тій же самій книзі, у якій знаходиться підсумковий лист, або в інших книгах.

Команду **Консолідація** можна використовувати декількома способами. Можна зв'язати консолідовані дані з вхідними даними, щоб наступні зміни у вхідних листах відображалися в підсумковому листі. Або можна просто консолідувати вхідні дані без створення зв'язків.

Консолідувати дані можна *по розташуванню* або *по категорії*.

- Якщо консолідувати дані *по розташуванню*, *Microsoft Excel* збирає інформацію з однаково розташованих чарунок кожного вхідного листа.
- При консолідації даних *по категорії* *Excel* використовує як основу для об'єднання листів заголовки стовпців або рядків.

Консолідація по категорії надає велику свободу для організації даних у вхідних листах. Наприклад, якщо графа **Січень** знаходиться в стовпці **B** на одному листі й у стовпці **D** на іншому листі, то при консолідації по категорії дані за січень усе рівно можна об'єднати.

Можна консолідувати листи, за допомогою будь-якої функції, приведеної у полі **Функція** зі списком, що розкривається, діалогового вікна **Консолідація**. За замовчуванням використовується функція **Сумма**, що підсумовує дані з кожного вхідного листа і поміщає результат у підсумковий лист.

Також можна використовувати кожен з наступних функцій: **Количество значений, Среднее, Максимум, Минимум, Произведение, Количество чисел, Несмещенное отклонение, Смещенное отклонение, Несмещенная дисперсия и Смещенная дисперсия**.

Можна консолідувати листи з відкритих у даний момент книг, або з тих, що знаходяться на диску. Книга, що містить лист, у який поміщаються консолідовані дані, повинна бути відкрита. Вхідні книги для консолідації можуть бути закриті, звичайно, за умови, що ви надали *Excel* правильні відомості про їхнє місцезнаходження, інакше він не зможе знайти усі файли цих книг. Ви повинні зберегти усі вихідні книги, перш ніж почати консолідацію.

Нижче даються три приклади консолідації: по розташуванню, по категорії і по категорії з використанням зв'язків із вихідними даними. У прикладах розглянута консолідація листів однієї книги, але точно так само можна консолідувати листи, що знаходяться в різних книгах.

### 4.3.1 Консолідація по розташуванню

При консолідації по розташуванню *Excel* застосовує підсумкову функцію (**Сумма, Среднее** або будь-яку іншу з перерахованих раніше) до чарунок з однаковими адресами в кожному вхідному листі. Це найпростіший спосіб консолідації, при якому консолідуємі дані у всіх вхідних листах повинні мати зовсім однакове розташування.

Приклад 4.12.- Виконати консолідацію звітів про реалізації безалкогольних напоїв за тиждень. Розташувати консолідовані звіти по розташуванню й категоріям на окремих листах.

#### Виконання:

1. Додати в книгу зі звітами 2 нових листи і присвоїти їх ярличкам імена **Консолід\_по\_розташ** і **Консолід\_по\_категор** (див. приклад 4.11).

2. Скопіювати в лист **Консолід\_по\_розташ** форму звіту по реалізації товару, очистити *кінцеву область* (область призначення), тобто блок чарунок, у котрий будуть поміщені консолідовані дані – **В7:К14**. Виділення не знімати!

3. Дано: Виберіть команду **Консолидация** в меню **Данные** (рисунок 4.22).

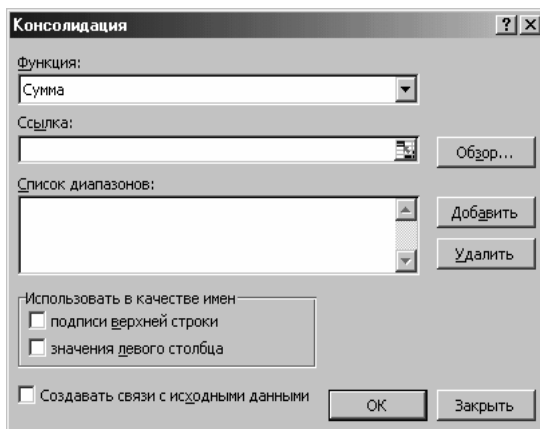


Рисунок 4.22 – Діалогове вікно консолідації

4. Для підрахунку значень з кожного вхідного листа треба в полі із списком **Функция**, що розкривається, в діалоговому вікні **Консолидация** вибрати функцію **Сумма**. Залишіть прапорці в секції **Использовать метки** неустановленими. Оскільки ми не збираємося створювати зв'язки з вхідними листами, прапорець **Создавать связи с исходными данными** також залиште неустановленими.

5. Уведіть посилання для кожного вхідного діапазону у полі **Ссылка** або виділіть ці діапазони за допомогою миші:

клацнути по полю **Ссылка** → клацнути по ярличку **Лист1** → виділити чарунки **В7:К14** (зменшити діалогове вікно можна клацанням по кнопці наприкінці поля **Ссылка**) → клацнути по кнопці **Добавить** (розгорнути діалогове вікно можна клацанням по тій же кнопці у кінці поля **Ссылка**) – зазначений діапазон чарунок повинен з'явитися у полі **Список диапазонов** → клацнути по ярличку **Лист2** → виділити чарунки **В7:К14** (при створенні форм документів у групі діапазон чарунок

буде установлений автоматично) → клацнути по кнопці **Добавить** → повторити зазначені дії для всіх звітів, що підсумовуються, → **ОК**.

Звичайно, використання миші є найбільш простим способом введення посилань, але якщо необхідно послатися на вхідні листі закритих у даний момент книг, то вам доведеться ввести ці посилання з клавіатури. (Можна використовувати кнопку **Обзор**, щоб визначити місцезнаходження файлу, а потім вручну ввести посилання на чарунку.) Посилання, що вводиться, повинне мати наступну форму:

{Ім'я файлу}Ім'я листа! Посилання

Якщо вхідному діапазону було призначено ім'я, можна використовувати це ім'я замість посилання.

**Зауваження.** Після виконання консолідації, посилання, введені у вікні діалогу **Консолидация**, запам'ятовуються при збереженні книги. Якщо при наступному відкритті книги необхідно оновити консолідовані значення, варто вибрати команду **Консолидация** і натиснути кнопку **ОК**.

**Примітка.** Консолідація виконується по блоку суміжних чарунк, тому графа «**Ціна**», як довідковий матеріал, винесена в кінець звіту, щоб не розривати діапазон, який консолідується.

### 4.3.2 Консолідація по категорії

Розглянутий приклад 4.12 припускав повну ідентичність звітів у частині розглянутих категорій, у даному випадку, реалізованих товарів, що в практичній діяльності зовсім не обов'язково. Список реалізованих товарів у кожної підзвітної особи може змінюватися. У цьому випадку необхідно використовувати консолідацію по категоріях.

**Приклад 4.13.** Внести у звіт про реалізації на **Лист1** дані по напої «Монастирський квас», на **Лист3** – дані по напоях «Монастирський квас» і «Содова», на **Лист5** – по напої «Буратіно». На листі **Консол\_по\_категор** створити консолідований звіт по всім реалізаторам за тиждень.

**Виконання.**

1. На лист **Консолід\_по\_категор** скопіювати заголовну частину таблиці звіту (**A1:L6**) із будь-якого листа.
2. Виділити чарунку **A7** як кінцеву область. При завданні однієї чарунки як кінцевої області команда **Консолидация** заповнює необхідну область нижче і праворуч від цієї чарунки.

3. М. **Данные** → к. **Консолидация** → заповнити вікно діалогу **Консолидация** → вибрати **Сумма** у полі **Функция** зі списком, що розкривається, → установити прапорець **В левом столбце** в секції **Использовать метки**.

4. Увести посилання для кожного вхідного діапазону у полі **Ссылка** або виділити ці діапазони за допомогою миші: клацнути по полю **Ссылка** → клацнути по ярличку **Лист1** → виділити чарунки **A7:K11** (у діапазон повинні бути включені чарунки з категоріями) → клацнути по кнопці **Добавить** → клацнути по ярличку **Лист2** → виділити чарунки **A7:K12** → клацнути по кнопці **Добавить** → повторити зазначені дії для всіх звітів, що підсумовуються, з виділенням даних про реалізовані товари → **ОК**.

5. Підсумковий лист тепер містить рядки, що відповідають унікальним елементам рядків вхідних листів (у даному випадку, для кожного товару). Якщо кілька листів мають однакові елементи рядків, підсумковий лист виконує обрану математичну операцію на відповідних значеннях у кожному стовпці.

6. Консолідований звіт можна відформатувати, додати підсумкові рядки з формулами, довідкову інформацію (наприклад, ціну).

Примітка – Категорії можуть розташовуватися або в лівому стовпці, або у верхньому рядку виділюваного діапазону, на що повинен указувати прапорець у секції **Использовать метки**. Тільки в одному стовпці або в одному рядку.

### 4.3.3 Створення зв'язків із вхідними листами

У попередніх прикладах виконувалася консолідація даних за допомогою функції **Сумма**. У результаті був отриманий ряд констант у підсумковому листі. Наступні зміни у вхідних листах не зроблять ніякого впливу на підсумковий лист, поки не повторити консолідацію (м. **Данные** → к. **Консолидация** → **ОК**).

Команду **Консолидация** можна також використовувати для встановлення постійного зв'язку між підсумковим і вхідним листами. Щоб створити цей зв'язок, варто установити у вікні діалогу **Консолидация** прапорець **Создавать связи с исходными данными** і виконати консолідацію так само, як і без зв'язку.

Коли виконується консолідація даних з установленим прапорцем **Создавать связи с исходными данными**, *Excel* створює структуру в підсумковому листі (рисунок 4.23).

Клацання по плюсі ліворуч від рядка звіту розкриває склад вхідних елементів, що формують підсумкові числа цього рядка (для рядка **11** це **A7:M11**). Можливо, прийдеться відформатувати деякі чарунки після виконання консолідації з установленням зв'язків, тому що в цьому випадку створюються додаткові рядки й стовпці.

#### 4.4 Використання зведених таблиць

У *Excel* мається чудовий засіб обробки і представлення даних, що набагато перевершує по гнучкості, різноманітності і зручності використання традиційних списків із проміжними підсумками – цей засіб називається зведені таблиці. Зведені таблиці не тільки дозволяють представити в набагато більш зручному й доступному для огляду виді інформацію з традиційних списків *Excel*, але також дають можливість використовувати для побудови зведених таблиць запити до зовнішніх баз даних.

1 2	A	B	C	D	E	F	G	H	I	J	K	L	M
1	<b>Звіт про реалізації безалкогольних напоїв</b>												
2	за												
3	матеріально-відповідальна особа												
4													
5	<b>Найменування</b>		<b>Залишок</b>	<b>Прихід</b>	<b>Реалізація по датах</b>							<b>Залишок</b>	<b>Ціна</b>
6	<b>товару</b>		<b>на початок</b>		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>на кінець</b>	
7		Звіт Про Реалізацію Напоїв	2	40	5	3	7	1	10	5	6	5	2,4
8		Звіт Про Реалізацію Напоїв	2	40	5	3	7	1	10	5	6	5	2,4
9		Звіт Про Реалізацію Напоїв	2	40	5	3	7	1	10	5	6	5	2,4
10		Звіт Про Реалізацію Напоїв	2	40	5	3	7	1	10	5	6	5	2,4
11		Звіт Про Реалізацію Напоїв	2	40	5	3	7	1	10	5	6	5	2,4
12	Фанта		10	200	25	15	35	5	50	25	30	25	2,4
18	Кока-кола		10	200	25	25	30	25	50	25	30	0	3
24	Миринда		10	200	25	10	25	15	50	25	30	30	3,2
30	Байкал		10	200	25	15	20	35	50	25	30	10	2,5
33	Монастирський квас		4	80	10	8	6	16	20	10	12	2	3,2
35	Содова		2	40	5	6	3	5	10	5	6	2	2,8
37	Буратіно		2	40	5	4	6	2	10	5	6	4	2,7

Рисунок 4.23 – Приклад консолідованого звіту зі зв'язками

**Зведена таблиця** – це плоска або об'ємна ( що складається з декількох шарів, або сторінок) прямокутна таблиця, що дозволяє просумувати або іншим способом підсумувати великі обсяги даних із розташованого в іншому місці робочої книги вхідного списку даних або декількох списків даних. Замість звичайного списку робочого листа *Excel* у якості вхідних даних можуть використовуватися дані з іншої зведеної таблиці або запиту до зовнішньої бази даних. Для підведення підсумків можна вибрати підходящу функцію або інший метод обчислень. Ці обчислення робляться для одного або декількох полів вхідного списку, оголошених як поле даних. При цьому деякі інші поля вхідного списку використовуються для угруповання даних у рядках і стовпцях зведеної таблиці.

Якщо консолідація дозволяє одержувати підсумкові таблиці з ряду вхідних таблиць, то зведені таблиці можуть формуватися як на основі однієї об'ємної таблиці шляхом підрахунку підсумків по визначених стовпцях і рядках, так і на основі ряду вхідних таблиць. Тобто консолідацію у визначеній мері можна розглядати окремим випадком такого могутнього засобу як зведені таблиці. У вхідних таблицях при формуванні зведених таблиць заголовні рядки повинні мати просту (без підрозділів) структуру.

Виклик **майстра зведених таблиць** виконується через:

меню **Данные** → к. **Сводная таблица** → відповідати на питання діалогових вікон майстра.

#### Приклад 4.14.

1. На **Листе 1** створити **Відомість реалізації овочів** (рисунок 4.24)
2. Створити зведену таблицю підсумкових сум реалізації по найменуваннях товару.
3. Створити зведену таблицю підсумкових сум реалізації по матеріально-відповідальних особах і датам.
4. Створити зведену таблицю підсумкових сум реалізації і торгових знижок по найменуваннях товару й датам.

	A	B	C	D	E	F	G
1	Відомість реалізації овочів						
2							
3	Дата	Матеріально-відповідальна особа	Найменування товару	Сума	Торгова знижка	Знижка на відходи	Разом
4	02.02.2005	Жук Н.И.	Помідори	1200	20	15	1235
5	02.02.2005	Орлова К.П.	Капуста	1500	15	12	1527
6	02.02.2005	Нилова В.В.	Морква	500	10	18	528
7	05.02.2005	Орлова К.П.	Капуста	600	10	20	630
8	05.02.2005	Жук Н.И.	Огірки	800	15	13	828
9	05.02.2005	Нилова В.В.	Капуста	1100	13	10	1123
10	10.02.2005	Орлова К.П.	Капуста	100	20	12	132
11	10.02.2005	Нилова В.В.	Капуста	600	15	15	630
12	15.02.2005	Жук Н.И.	Помідори	400	16	13	429
13	15.02.2005	Орлова К.П.	Капуста	700	15	17	732
14	23.02.2005	Нилова В.В.	Морква	500	20	15	535
15	23.02.2005	Жук Н.И.	Огірки	600	20	16	636
16	23.02.2005	Орлова К.П.	Морква	900	14	12	926
17	Разом:			9500	203	188	9891

Рисунок 4.24 – Форма заповненої відомості реалізації овочів

#### Виконання:

1. В чарунках **A1:G17** створити документ приведеної на рисунку 4.24 форми.
2. Створення зведеної таблиці підсумкових сум реалізації по найменуваннях товару:

п.м. **Данные** → к. **Сводная таблица** → у вікні **Мастер сводных таблиц и диаграмм – шаг 1 из 3** уключити перемикач у списке или базе данных **Microsoft Excel** → клацнути по кнопці **Далее** → у вікні **Мастер сводных таблиц и диаграмм – шаг 2 из 4** виділити діапазон із вхідними даними – обвести рухомою пунктирною рамкою чарунки **A3:G16** → кнопка **Далее** → у вікні **Мастер сводных таблиц и диаграмм – шаг 3 из 3** клацнути по кнопці **Макет** (для **Excel 97** вікно з макетом з'явиться автоматично на третьому кроці роботи Майстра) → у вікні **Мастер сводных таблиц и диаграмм – макет** перетягнути мишкою кнопку поля **Наименование товара** в область **Строка**, а кнопку **Итого** – в область **Данные** (рисунок 4.25) → кнопка **Ок** → у вікні **Мастер сводных таблиц – шаг 3 из 3** уключити перемикач **новый лист** для розміщення зведеної таблиці на новому листі → клацнути по кнопці **Готово**.



У результаті в книгу буде вставлений новий лист із зведеною таблицею сум реалізації по найменуваннях товару (рисунок 4.26).

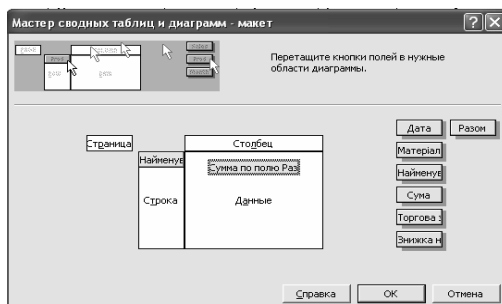


Рисунок 4.25 – Діалогове вікно для формування структури зведеної таблиці сум реалізації по найменуваннях товару

	А	В
1		
2		
3	Сумма по полю Разом	
4	Найменування товару ▼	Итого
5	Капуста	4774
6	Морква	1989
7	Огірки	1464
8	Помідори	1664
9	Общий итог	9891

Рисунок 4.26 – Зведена таблиця сум реалізації по найменуваннях товару

3. Створення зведеної таблиці підсумкових сум реалізації по матеріально-відповідальних особах і датам:

п.м. **Данные** → к. **Сводная таблица** → у вікні **Мастер сводных таблиц и диаграмм** – шаг 1 из 3 уключити перемикач у списке или базе данных **Microsoft Excel** → кнопка **Далее** → у вікні **Мастер сводных таблиц и диаграмм** – шаг 2 из 3 виділити діапазон, із вхідними даними – обвести рухомою пунктирною рамкою чарунки **A3:G16** → кнопка **Далее** → у вікні, що з'явиться, **Microsoft Excel** клацнути по кнопці **Нет** для створення незалежної зведеної таблиці → у вікні **Мастер сводных таблиц и диаграмм** – шаг 3 из 3 клацнути по кнопці **Макет** → у вікні **Мастер сводных таблиц и диаграмм** – макет перетягнути мишкою кнопки полів **Материально-відповідальна особа** і **Дата** в область **Строка**, а кнопку **Разом** – в область **Данные** (рисунок 4.27) → кнопка **Ок** → у вікні **Мастер сводных таблиц** – шаг 3 из 3

включити перемикач **новый лист** для розміщення зведеної таблиці на новому листі → клацнути по кнопці **Готово**.

У результаті в книгу буде вставлений новий лист із зведеною таблицею підсумкових сум реалізації по матеріально-відповідальних особах і датам (рисунк 4.28).

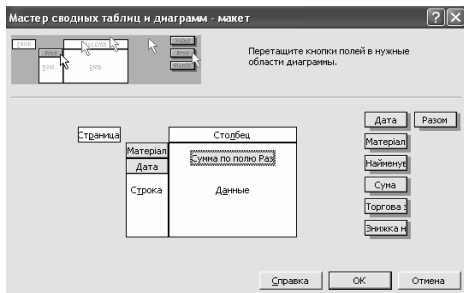


Рисунок 4.27 – Діалогове вікно для формування структури зведеної таблиці сум реалізації по матеріально-відповідальних особах і датам

	А	В	С
1			
2			
3	Сумма по полю Разом		
4	Материально-відповідальна особа	Дата	Итого
5	Жук Н.И.	02.02.2005	1236
6		06.02.2005	828
7		15.02.2005	429
8		23.02.2005	636
9	Жук Н.И. Итого		3128
10	Нилова В.В.	02.02.2005	528
11		06.02.2005	1123
12		10.02.2005	630
13		23.02.2005	535
14	Нилова В.В. Итого		2816
15	Орлова К.П.	02.02.2005	1527
16		06.02.2005	630
17		10.02.2005	132
18		15.02.2005	732
19		23.02.2005	926
20	Орлова К.П. Итого		3947
21	Общий итог		9891

Рисунок 4.28 – Зведена таблиця сум реалізації по матеріально-відповідальних особах і датам.

4. Створення зведеної таблиці підсумкових сум реалізації і торгових знижок по найменуваннях товару й датам:

п.м. **Данные** → к. **Сводная таблица** → у вікні **Мастер сводных таблиц и диаграмм – шаг 1 из 3** включити перемикач у списке или базе данных **Microsoft Excel** → кнопка **Далее** → у вікні **Мастер сводных таблиц и диаграмм – шаг 2 из 3** виділити діапазон, із вхідними даними – обвести рухомою пунктирною рамкою чарунки **A3:G16** → кнопка **Далее** → у вікні **Microsoft Excel** клацнути по кнопці **Нет** для створення незалежної зведеної таблиці → у вікні **Мастер сводных**

таблиц и диаграмм – шаг 3 из 3 клацнути по кнопці **Макет** → у вікні **Мастер сводных таблиц и диаграмм – макет** перетягнути мишкою кнопки полів **Найменування товару** и **Дата** в область **Строка**, а кнопки **Разом** и **Торгова знижка** – в область **Данные** (рисунок 4.29) → кнопка **Ok** → у вікні **Мастер сводных таблиц – шаг 3 из 3** уключити перемикач **новый лист** для розміщення зведеної таблиці на новому листі → клацнути по кнопці **Готово**.

У результаті в книгу буде вставлений новий лист із зведеною таблицею.

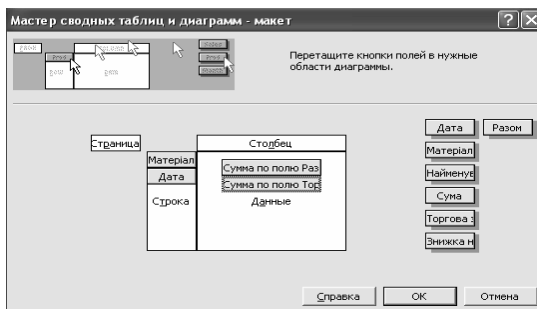


Рисунок 4.29 – Діалогове вікно для формування структури зведеної таблиці підсумкових сум реалізації і торгових знижок по найменуваннях товару й датам реалізації

**Примітка** – Коли числові дані у вікні макета попадають в область даних, то за замовчуванням Майстер пропонує обробити них за допомогою функції **Сумма**. Для зміни функції потрібно виконати подвійне клацання по необхідному полю, розміщеному в області даних і в діалоговому вікні **Вычисление поля сводной таблицы**, що з'явившомуся, (рисунок 4.30) вибрати операцію для обробки даних.

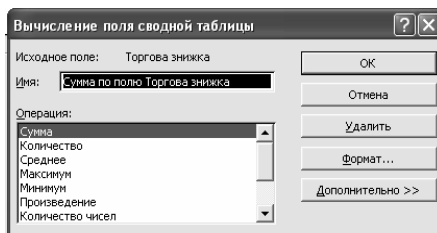


Рисунок 4.30 – Діалогове вікно для вибору функції обробки даних

**Приклад 4.15.** – Створити щоденні звіти про реалізації безалкогольних напоїв по кожному з трьох реалізаторов – Сидоров, Петрова, Жукова за два дні (зразок із заповненням на рисунку 4.31) і, використовуючи модуль зведеної звітності, побудувати:

1. консолідований звіт по реалізації безалкогольних напоїв по датах;
2. посторінковий консолідований звіт по реалізації безалкогольних напоїв в асортименті, сторінки по МОЛ і датам;
3. зведений звіт про реалізації по матеріально-відповідальних особах у кількісному вираженні;
4. зведений звіт про реалізації по матеріально-відповідальних особах у сумовому вираженні;
5. побудувати зведену діаграму по реалізації в асортименті по датах.

	A	B	C	D	E	F	G	H	I	J
1	<b>Звіт про реалізації безалкогольних напоїв</b>									
2	матеріально-відповідальна особа			Сидоров		Дата		01.01.2005		
3	упаковки					грн.				
4	Найменування товару	Залишок на початок_K	Прихід_K	Реалізовано (кількість)	Залишок на кінець_K	Ціна	Залишок на початок_C	Прихід_C	Реалізовано (сума)	Залишок на кінець_C
5	Фанта	2	30	9	23	2,4	4,8	72	21,6	55,2
6	Кока-кола	5	25	5	25	3	15	75	15	75
7	Миринда	4	22	4	22	3,2	12,8	70,4	12,8	70,4
8	Байкал	1	17	7	11	2,5	2,5	42,5	17,5	27,5
9	Монастирський квас	0	50	5	45	3,2	0	160	16	144
10	Разом	12	144	30	126		35,1	419,9	82,9	372,1

Рисунок 4.31 – Форма заповненого звіту по реалізації безалкогольних напоїв

### Виконання.

Відкрити нову книгу, додати в неї необхідну кількість листів, об'єднати шість листів і створити форму звіту відповідно до рисунка 4.24 (див. Приклад 4.9); розгрупувати листи і заповнити їх вхідними даними. Матеріально-відповідальні особи – Сидоров, Петрова і Жукова, дати – 01.01.2001 і 02.01.2001; асортимент зазначений у таблиці рисунка 4.31. Перейменувати листи з вхідними даними, дати їм імена матеріально – відповідальних осіб із номером дати.

1). Додати новий лист, перемістити його в кінець книги, дати назву **Консол\_дати**. Активізувати чарунку **A3**. Тепер приступити до побудови звіту:

м. **Данные** → к. **Сводная таблица** → вибрати селекторні кнопки в **нескольких диапазонах консолидации** і **сводная таблица** → кнопка **Далее** → селекторна кнопка **Создать поля страницы** → кнопка **Далее** → указати у полі **Диапазон** діапазон **A4:J9** із листа **Сидоров1**: → кнопка **Добавить** → аналогічні дії виконати для інших листів вхідних звітів → клацнути по селекторній кнопці з номером **1** (одне сторінкове поле по даті) → виділіть у списку діапазонів рядок по **Сидоров1** → у полі **Первое поле** укажіть відповідну дату **01.01.2005** → аналогічні дії проробити для кожного з інших рядків списку діапазону (рисунок 4.32) → кнопка **Далее** → вибрати селекторну кнопку **существующий лист** і проконтролювати посилання на чарунку (повинна бути **A3**) → кнопка **Готово**.

У зведеній таблиці, що з'явилася, частина якої показана на малюнку 4.33, може виявитися далеко не те, що очікувалося: розтягнуті, упорядковані за алфавітом графі, неправдоподібні результати, загальні підсумки, що не потрібні. Це зв'язано з параметрами таблиці і її полів.

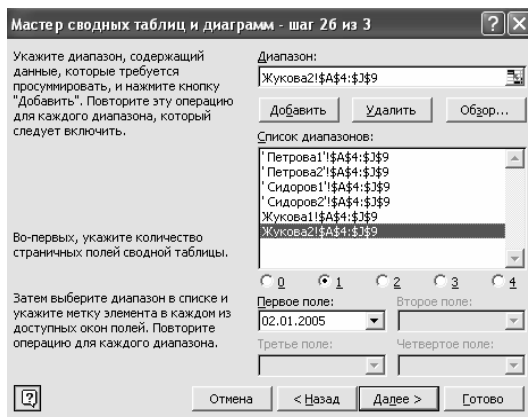


Рисунок 4.32 – Вікно Майстра зведених таблиць для введення діапазонів вхідних даних

	A	B	C	D	E	F	G	H	I	J	K
1	Страница1	(Все)									
2											
3	Сумма по полю Значение	Столбец									
4	Строка	Залишок на кінець_K	Залишок на кінець_C	Залишок на початок_K	Залишок на початок_C	Прихід_K	Прихід_C	Реалізовано (кількість)	Реалізовано (сума)	Ціна	Общий итог
5	Байкал	86	165	6	15	102	255	42	105	15	771
6	Кока-кола	150	450	30	90	150	450	30	90	18	1458
7	Миринда	132	422,4	24	76,8	132	422,4	24	76,8	19,2	1329,6
8	Монастирський квас	270	864	0	0	300	960	30	96	19,2	2539,2
9	Фанта	138	331,2	12	28,8	180	432	54	129,6	14,4	1320
10	Общий итог	756	2232,6	72	210,6	864	2519,4	180	497,4	85,8	7417,8

Рисунок 4.33 – Фрагмент первинного консолідованого звіту для Excel 2000

Щоб прибрати непотрібні елементи в стовпцях (рядках), необхідно в списку, що випадає, **Столбец (Строка)** прибрати відповідні прапорці. У нашому прикладі – це **Ціна** й **ОК**. Загальний підсумок по рядках для даного приклада не має змісту, тому його варто прибрати:

клацнути в будь-якому місці консолідованої таблиці правою кнопкою миші → к. **Мастер** → кнопка **Параметры** → зняти прапорець **Общая сумма по строкам** → **ОК**.

У цілому загальні підсумки завжди підраховуються по тій же функції, що і поле даних, для якого вони обчислюються, і їхній вивід можна установлювати відразу при формуванні звітної таблиці, використовуючи кнопку **Параметры** на третьому кроці майстра зведених таблиць.

Неправдоподібні результати зв'язані з невірно заданою функцією обчислення. За замовчуванням до числових полів застосовується формула **Сумма**, а до будь-яких нечислових полів – функція **Кількість значень**. Т.к. у вхідних листах були виділені і числові й нечислові поля, то була застосована остання функція. На попередніх кроках виконання приклада нечислові поля були вилучені зі зведеної таблиці, тому необхідно змінити функцію. Змінити функцію для підрахунку можна в такий спосіб: клацнути по елементу в області даних зведеної таблиці → к. **Параметри поля** → у діалоговому вікні (рисунок 4.34) у полі **Операция** вибрати потрібну – **Сумма** → **ОК**. Кнопка **Дополнительно >>** дозволяє розширити перелік доступних операцій, а кнопка **Формат** – відформатувати представлення результатів, що виводяться.

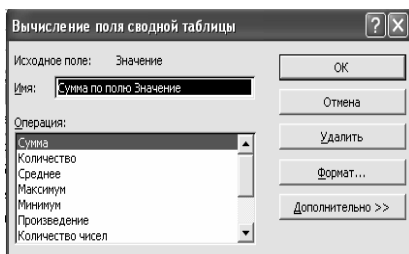


Рисунок 4.34 – Вікно для вибору підсумкових функцій

За замовчуванням поля зведеної таблиці сортуються за алфавітом. Для даного прикладу це неприйнятно. Графу **Прихід\_С** варто перемістити на місце звичайним способом – виділити й перетягнути на місце після **Залишок на початок\_С**. У більш складному випадку змінити порядок сортування можна шляхом набору в заголовній частині графи потрібної назви. Це буде сприйнято майстром зведених таблиць як вимога до переміщення графи з набраною назвою в дане місце. Нестандартний порядок сортування зберігається при відновленні й реорганізації таблиці.

Далі можна відформатувати таблицю – зменшити ширину граф і створити багаторядковий текст у заголовній частині таблиці стандартним способом: виділити **B4:I4** → м. **Формат** → к. **Ячейки** → вкладка **Вирівнювання** → виставити прапорець **Переносить по словам** й установити вирівнювання **по центру**. Можна зробити рамки звичайним способом. У результаті виконаної роботи консолідований звіт придбає вид (рисунок 4.35)

**Примітка** – Формати границь зведеної таблиці не зберігаються при реорганізації таблиці.

	A	B	C	D	E	F	G	H	I
1	Страница1	(Все)							
2									
3	Сумма по полю Значение	Столбец							
4	Строка	Залишок на початок_К	Прихід_К	Реалізовано (кількість)	Залишок на кінець_К	Залишок на початок_С	Прихід_С	Реалізовано (сума)	Залишок на кінець_С
5	Байкал	6	102	42	66	15	255	105	165
6	Кока-кола	30	150	30	150	90	450	90	450
7	Миринда	24	132	24	132	76,8	422,4	76,8	422,4
8	Монастирський квас	0	300	30	270	0	960	96	864
9	Фанта	12	180	54	138	28,8	432	129,6	331,2
10	Общий итог	72	864	180	756	210,6	2519,4	497,4	2232,6

Рисунок 4.35 – Консолідований звіт по реалізації безалкогольних напоїв по датах

2). Побудова посторінкового консолідованого звіту по реалізації безалкогольних напоїв в асортименті:

м. **Данные** → к. **Сводная таблица** → вибрати селекторні кнопки в **нескольких диапазонах консолидации** и **сводная таблица** → кнопка **Далее** → селекторна кнопка **Создать поля страницы** → кнопка **Далее** → у діалоговому вікні указати й додати діапазони вхідних даних **A4:19** → виділити кількість сторінкових полів –2 (МВО і дати) → виділити в **Списке диапазонов** перший лист вхідних даних і заповнити **Первое поле** и **Второе поле** (рисунок 4.36) → потім виділити другий діапазон і указати потрібні елементи (у полях можна скористатися списками, що випадають) → кнопка **Далее** → вибрати селекторну кнопку **новый лист** → **ОК**.

У результаті в книгу буде вставлений новий лист із зведеною таблицею, який необхідно відформатувати зазначеним вище способом (рисунок 4.37).

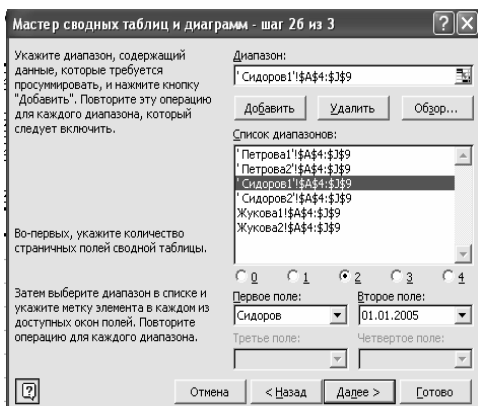


Рисунок 4.36 – Вікно Майстра зведених таблиць для вказівки діапазонів вхідних даних при посторінковій організації звіту.

	A	B	C	D	E	F	G	H	I
1	Страница1	(Все)							
2	Страница2	(Все)							
3									
4	Сумма по полю Значение	Столбец							
5	Строка	Залишок на початок_K	Прихід_K	Реалізовано (кількість)	Залишок на кінець_K	Залишок на початок_C	Прихід_C	Реалізовано (сума)	Залишок на кінець_C
6	Байкал	6	102	42	66	15	255	105	165
7	Кока-кола	30	150	30	150	90	450	90	450
8	Миринда	24	132	24	132	76,8	422,4	76,8	422,4
9	Монастирський квас	0	300	30	270	0	960	96	864
10	Фанта	12	180	54	138	28,8	432	129,6	331,2
11	Общий итог	72	864	180	756	210,6	2519,4	497,4	2232,6

Рисунок 4.37 – Посторінковий консолідований звіт про реалізації в асортименті по МОЛ і датам



У списках сторінок, що розкриваються, можна вибрати дані по окремим МВО, окремим датам і по тому й іншому разом.

3). Побудова Зведеного звіту про реалізації по матеріально-відповідальних особах у кількісному вираженні виконується аналогічно попереднім звітам. У списку Звіту **Столбец**, що випадає, залишити прапорці тільки у реквізитів у кількісному вираженні (рисунок 4.38).

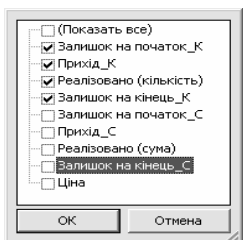


Рисунок 4.38 – Склад списку Зведеного звіту **Столбец**, що випадає, в кількісному вираженні

4). Побудова Зведеного звіту про реалізації по матеріально-відповідальних особах у сумовому вираженні виконується аналогічно попередньому звітові.

Форми зведених звітів у кількісному й сумовому вираженнях показані на рисунку 4.39.

	A	B	C	D	E
1	<b>Звіт у кількісному вираженні</b>				
2	Страниця1	(Все)			
3					
4	Сумма по полю Значение	Столбец			
5	Строка	Залишок на початок_К	Прихід_К	Реалізовано (кількість)	Залишок на кінець_К
6	Байкал	6	102	42	66
7	Кока-кола	30	150	30	150
8	Миринда	24	132	24	132
9	Монастирський квас	0	300	30	270
10	Фанта	12	180	54	138
11	Общий итог	72	864	180	756

	A	B	C	D	E
1	<b>Звіт у сумовому вираженні</b>				
2	Страниця1	(Все)			
3					
4	Сумма по полю Значение	Столбец			
5	Строка	Залишок на початок_С	Прихід_С	Реалізовано (сума)	Залишок на кінець_С
6	Байкал	15	255	105	165
7	Кока-кола	90	450	90	450
8	Миринда	76,8	422,4	76,8	422,4
9	Монастирський квас	0	960	96	864
10	Фанта	28,8	432	129,6	331,2
11	Общий итог	210,6	2519,4	497,4	2232,6

Рисунок 4.39 – Звіти про реалізації по МВО Сидорову

**Реорганізація** (зміна структури) зведених таблиць здійснюється за допомогою виклику майстра зведених таблиць: клацнути правою кнопкою миші по таблиці → к. **Мастер** → далі працювати з майстром як у попередніх пунктах при побудові таблиць.

**Відродження** зведених таблиць необхідно при зміні вмісту вхідних таблиць – клацнути правою кнопкою миші по таблиці → к. **Обновить данные**. Програма виконає перерахування зведеної таблиці. Можна в параметрах таблиці виставити прапорець **Обновлять при открытии**.

5). Побудова зведеної діаграми по реалізації в асортименті по датах виконується по відповідному зведеному звіті – у контекстному меню к. **Сводная диаграмма**. Можна відразу в майстру зведених таблиць вибрати селекторну кнопку **сводная диаграмма (со сводной таблицей)**, по якій одночасно будуть будуватися зведена таблиця й діаграма. Зведена діаграма автоматично з'являється на новому листі діаграм і користувач повинен привести її у потрібний для себе вид: клацнути правою кнопкою миші по діаграмі → к. **Тип диаграммы** → вкладка **Нестандартные** → вибрати **График/гистограмма** → **ОК** → контекстне меню → к. **Параметры диаграммы** → вкладка **Заголовки** → внести назву діаграми і осей **У** (Кількість і Сума) → **ОК**. Повинна вийти діаграма аналогічна показаній на рисунку 4.40.

Примітка – при необхідності можна розмістити зведену діаграму на однім листі зі зведеною таблицею – контекстне меню → к. **Размещение** → вибрати селекторну кнопку **на имеющемся** й у списку, що випадає, указати назву потрібного листа → **ОК**. Зведена діаграма буде перенесена на указаний лист.

На дані зведеної таблиці можна посилатися з інших таблиць. При цьому можна використовувати звичайні посилання. Але звичайні посилання при реорганізації таблиці повертають невірний результат. Тому варто використовувати у формулі функцію **ПОЛУЧИТЬ. ДАННЫЕ. СВОДНОЙ. ТАБЛИЦЫ()** з указівкою аргументів – перший указує зведену таблицю, другий – описує дані, що викликаються. Більш докладно див. у довідковій системі.

Реалізація напоїв в асортименті по датам

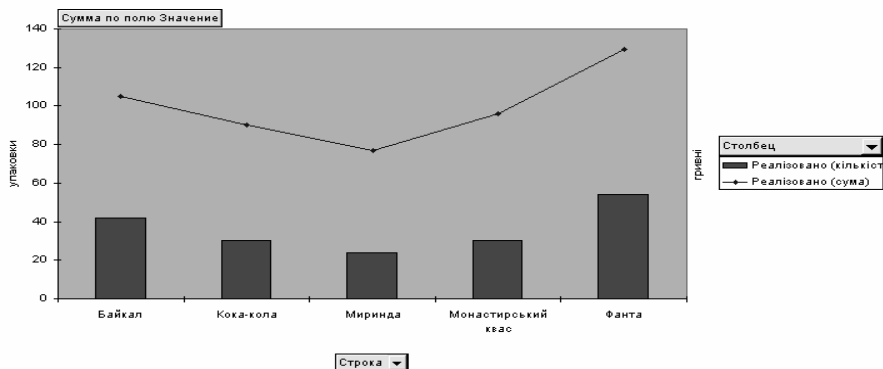


Рисунок 4.40 – Приклад зведеної діаграми

## Вправи для закріплення матеріалу

- Увійти в оболонку Windows і запустити Microsoft Excel.
- Переглянути вміст рядка меню й склад команд кожного з пунктів.
- Переглянути склад панелей інструментів для роботи в Excel. Залишити на екрані панелі **Стандартная** й **Форматирование**.
- Забрати й відновити елементи екрана:
  - рядок формул
  - смуги прокручування
  - ярлички листів
  - рядок стану
  - сітку
- Установити режим перегляду робочої книги **Обычный**, масштаб – **100%**
- Уставити в Книгу два нових листи і перемістити їх у кінець списку.
- Перейменувати Лист1, привласнивши йому своє прізвище.
- Видалити вставлені листи.
- В чарунки A2, A3, A4 і C3 відповідно ввести текст: “Текст”, “Числа”, “Формули”, “Вправа”.
- В чарунки C3, D3, E3 ввести довільні значення чисел.
- В чарунки C4 і D4 ввести формули:  
 $=C3/(D3-E3)$  і  $=(C3+E3)*5-D3$
- В чарунку E4 ввести формулу підсумовування чисел у діапазоні C3:E3, використовуючи функцію **Автосумма**.
- Установити режим виводу формул на екран, переглянути створені формули, скасувати режим виводу формул.
- Виконати автозаповнення найменуваннями місяців у рядку 7. Якщо такий список відсутній, то створити його.

15. Виділити по черзі і зняти виділення:
  - а) чарунки C2, A2, A3, A4
  - б) блок чарунок A2:E5
  - в) стовпці A, C, E
  - г) рядка 2, 4, 7
16. Вивчити можливості форматування чисел (пункт меню **Формат** → команда **Ячейки** → вкладка **Числа**). Установити різний формат для чисел в чарунках C3, D3, E3.
17. Відформатувати за своїм розсудом чарунки A2, A3, A4 (пункт меню **Формат** → команда **Ячейки** → вкладки **Шрифт, Вид, Выравнивание**).
18. Скопіювати вміст чарунок A2:E4 на Лист 2 робочої Книги.
19. Випробувати можливості зміни ширини стовпців (висоти рядків) вручну за допомогою мишки і з використанням команд пункту меню **Формат**.
20. Уставити порожні стовпці між стовпцями A і B; D і E.
21. Уставити два порожні рядки між рядками 1 і 2.
22. Видалити вставлені стовпці й рядки.
23. Зберегти робочу Книгу під ім'ям «Вправа».
24. Вийти з Microsoft Excel.

## Питання для самоконтролю

1. У чому розходження абсолютних і відносних посилань на чарунки?
2. Правило запису формул у Excel?
3. Чим відрізняються функції від формул?
4. Як указати вкладені функції?
5. Сутність формули масиву?
6. Призначення імен чарунок і діапазонів?
7. Консолідація і зведені таблиці – розходження?
8. Сутність консолідації по розміщенню і по категоріях?

## Завдання для самостійного виконання

### Завдання 1

1. Увійти в оболонку Windows і запустити Microsoft Excel.
2. На Листі 1 створити документ наступної форми:

#### Відомість реалізації товарів

Найменування товару	Одиниці виміру	Ціна	Кількість	Вартість (гр.3*гр.4)	Торгова знижка	
					Відсоток	Сума (гр.5*гр.6/100)
				*		*
				*		*
				*		*
				*		*
Разом:				*		*

- В чарунках із символом “\*” необхідно розмістити розрахункові формули.
3. Відформатувати таблицю аналогічно приведеної у завданні.
  4. Зберегти порожню форму документа під ім'ям *Реалізація 1*.
  5. Створити список, що містить найменування товарів (елементи списку вибрати самостійно).
  6. Виконати автозаповнення елементами створеного списку чарунок графі ”**Найменування товару**”. Заповнити довільними даними інші графи документа.
  7. Заповнити документ довільними вхідними даними.
  8. Підготувати документ до друку в режимі попереднього перегляду:
    - a) змінити орієнтацію папера, масштаб для розміщення, виконати центрування на сторінці;
    - b) створити нижній колонтитул, помістивши в ньому дату створення документа і своє прізвище.
  9. Роздрукувати документ.
  10. Скопіювати документ на Лист2.
  11. Установити режим виводу формул.
  12. Зменшити ширину стовпців. Роздрукувати документ із формулами. Повернутися на Лист1.
  13. Створити кругову діаграму аналізу вартості товарів, розмістивши її на одному листі з документом.
  14. Створити гістограму аналізу *суми торгової знижки* на реалізовані товари, розмістивши її на окремому листі робочої книги.
  15. Відформатувати створені діаграми за своїм розсудом (колір, шрифт, фон і т.д.)
  16. Зберегти робочу Книгу під ім'ям *Реалізація 2*.
  17. Завершити роботу з Microsoft *Excel*.

## Завдання 2

Для таблиць із приклада 4.15, створити

- a) зведений звіт МВО по видах операцій (залишки, прихід, витрата);
- b) консолідований звіт по МВО;
- v) побудувати зведену діаграму по реалізації товарів у сумовому вираженні по МВО.

## Завдання 3

Створіть таблицю за приведеною формою.

Заповніть таблицю довільними даними (не менш 15 рядків) і створіть зведені звіти:

- a) одержання товарів у розрізі країн-постачальників;
- b) одержання товарів по роках і кварталам;
- v) одержання по кодах товарів у кількісному й сумовому вираженні.

Рік	Квартал	Код товару	Країна-постачальник	Кількість	Сума
1999	1	124512	Україна	149	349,11
1999	1	124512	Польща	72	568,50
			і т.д.		

#### Завдання 4

Для підбора емпіричної кривої за результатами 9 експериментів необхідно розрахувати величини, представлені в нижченаведеній таблиці. Використовуючи формули масивів заповнити приведену таблицю.

№	Y	E	$Y_i^2$	$X_i = E_i - \bar{E}$	$Y_i * X_i$	$X_i^2$	$X_i^2 * Y_i$	$X_i^3$	$X_i^4$	$X_i^3 * Y_i$	$X_i^5$	$X_i^6$
1												
2												
3												
4												
5												
6												
7												
$\Sigma$												

#### Література

1. Бородкіна І.Л., Матвієнко О.В. Практичний курс з комп'ютерних технологій підготовки даних: Навчальний посібник. – К.: Центр навчальної літератури, 2004. – 448с.
2. Войтюшенко Н.М., Остапєць А.І. Комп'ютерна техніка і програмування. Частина 1. Основи підготовки користувача ПК: Навчальний посібник по базовій підготовці для студентів денної і заочної форм навчання. – Донецьк, ДонДУЕТ, 2001 – 150 с.
3. Додж М., Кіната К., Стінсон К., Эффективная работа с *Excel 2000* – СПб: Издательство «Питер», 2000. – 1056 с.
4. Информатика: Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів / за ред. О.І.Пушкаря. – К.: Видавничий центр “Академія”, 2003 – 704 с.
5. Компьютеры, сети, Интернет. Энциклопедия./Новиков Ю и др. – СПб.: Питер, 2002. – 928 с.
6. Хэлворсон М., Янг М. Эффективная работа с *Microsoft Office 2000*. – СПб: Издательство «Питер», 2000. – 1232с.

## Розділ 5. Підготовка прилюдних виступів і презентацій

### 5.1 Загальна характеристика пакета PowerPoint

Основна функція *PowerPoint* – це підготовка матеріалів для прилюдного виступу або рекламної інформації. В даний час тексти для показу можна готувати за допомогою текстового процесора, таблиці чи діаграми – за допомогою електронних таблиць, малюнки чи схеми – за допомогою графічних пакетів. Однак подібний підхід до справи приводить до деякого різнобою в стилі оформлення підготовлених матеріалів, і загальний результат виглядає не дуже вражаюче. Сучасний пакет підготовки презентацій не тільки допоможе досягти стильової єдності інформації, що представляється, істотно поліпшить зовнішній вигляд матеріалів і полегшить їхню підготовку за допомогою спеціальних графічних засобів, але також допоможе спланувати виступ. Можна продемонструвати необхідні графічні матеріали безпосередньо на комп'ютері за допомогою *PowerPoint*

Перш, ніж починати працювати з пакетом, необхідно продумати зміст і структуру матеріалу, що буде оформлюватися. Презентація не може включати занадто багато слайдів, тому що це утомливо для слухача, з одного боку, а, з іншого боку, сам доповідач завжди обмежений у часі. Тому матеріал, що виноситься на слайди, повинний бути виразним і ємним.

Насамперед, це комплект підготовлених для показу слайдів, в яких відбиваються план та основні положення виступу, необхідні схеми, таблиці, діаграми й малюнки. Однак матеріали презентації мають ще й інші форми представлення: структура презентації, роздавальні матеріали й сторінки заміток.

Запуск PowerPoint: кнопка **Пуск** → к. **Програми** → п. **PowerPoint** → у діалоговому вікні (рисунок 5.1) вибрати один із режимів роботи: створення нової презентації одним із трьох способів або відкриття існуючої презентації (для Office 97 – а) чи на вкладці **Общие** (для Office 2000 – б) → **ОК**.

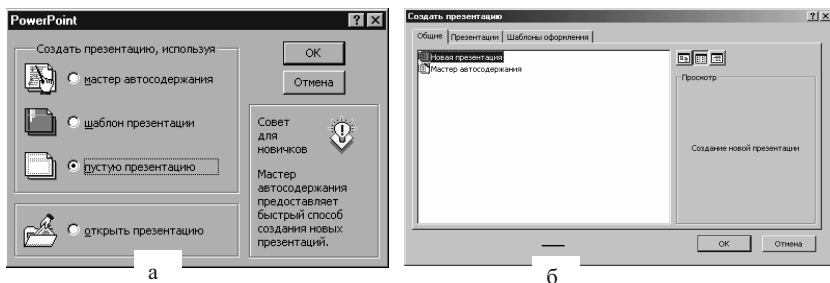


Рисунок 5.1 – Діалогове вікно, що з'являється при запуску *PowerPoint*

## 5.2 Створення примітивної презентації

За замовчуванням пропонується скористатися допомогою **Мастера автозмісту**. Це дуже зручний інструмент. З його допомогою можна дуже швидко створювати типові презентації, використовуючи як шаблон уже наявні варіанти презентацій, що складаються зі спеціальних макетів слайдів. Готовий шаблон потрібно заповнити власним текстом, потім доповнити, якщо це потрібно, допоміжними слайдами з графікою й діаграмами, відредагувати їхній зовнішній вигляд – і презентація готова. Однак, щоб краще познайомитися з основними можливостями програми, варто почати з порожньої презентації.

### Слайди: створення й збереження

Після завантаження програми з'являється діалогове вікно **Создать слайд** (рисунок 5.2), у якому необхідно вибрати одну з додатних розміток для нового слайда. Можна вибрати кожен з 24 наявних стандартних розміток слайда. Для відображення наявних розміток використовується смуга прокручування праворуч від їхніх схематичних зображень. Для першого слайда презентації найкраще підійде розмітка **Титульный слайд**. Назва розмітки з'являється в правій нижній частині діалогового вікна.



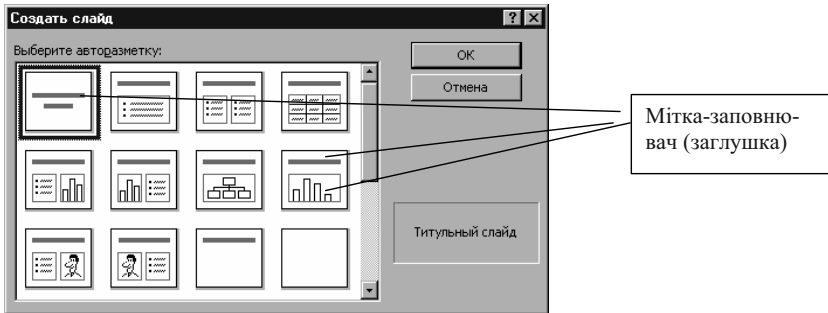


Рисунок 5.2 – Діалогове вікно **Создать слайд**

Кожен знову створений слайд містить мітки-заповнювачі, чи заглушки. В області заголовка слайда буде присутня заглушка для заголовка слайда, а в основній області слайда, у залежності від того, яка з розміток використана, будуть присутні різні мітки-заповнювачі для тексту чи об'єктів, що вставляються на слайд. Для введення тексту потрібно клацнути по мітці-заповнювачу (заклушці) і почати вводити текст. Вихідна кількість, розміри, розташування й форматування міток-заповнювачів визначаються обраною розміткою слайда. Якщо потрібно, можна змінити накреслення чи розмір шрифту, який використовувався, а також за допомогою перетаскування мишею змінити розташування вставленого тексту на слайді. Узагалі можна довільним образом змінювати форматування об'єктів на слайді *PowerPoint*, подібно тому, як можна змінити форматування фрагмента тексту в документі *Word*. На рисунку 3 замість тексту уставлено 2 фігурних тексти звичайним для *Word* чином: кнопка **Добавить объект WordArt** на панелі **Рисование** або м. **Вставка** → к. **Рисунок** → п. **Объект WordArt**. При цьому заглушки були вилучені – виділити клацанням заглушку і натиснути **[Delete]**.

Для надання привабливого виду слайдам варто підібрати шаблон презентації (дизайн чи оформлення презентації), що підходить, а потім можна внести необхідні зміни в її зовнішній вигляд. Меню **Формат** → к. **Применить шаблон оформления** → у діалоговому вікні вибрати шаблон (праворуч у вікні розташований зразок, для приклада обраний шаблон «румянец») → кнопка **Применить**. Обраний шаблон буде розповсюджений на всі слайди презентації.

Зауваження: Зміна оформлення презентації може трохи змінити взаємне положення об'єктів на готових слайдах.

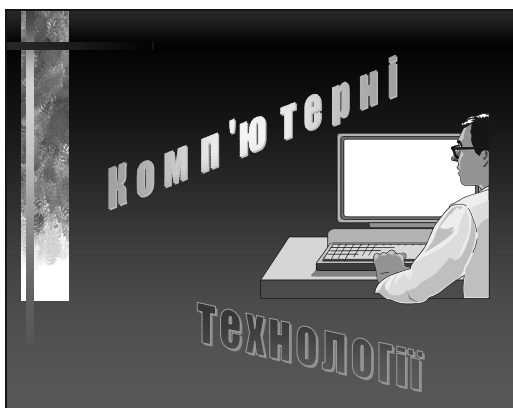


Рисунок 5.3 – Титульний слайд презентації *Комп'ютерні технології*

Крім того, титульний слайд ілюстрований картинкою, хоча в шаблоні вона не передбачена. Для вставки картинок у будь-який слайд можна використовувати м. **Вставка** → к. **Рисунок** → п. **Картинки** → вибрати й уставити потрібну картинку → розмістити її на слайді.

Корисно відразу ж зберегти новий файл презентації і визначити для нього придатне ім'я: м. **Файл** → к. **Сохранить как** → дати ім'я файлу *Ком-п'ютерні технології* → кнопка **Сохранить**.

### Маркірований список

Другим зручно поставити слайд із планом презентації, який можна представити як маркірований список. Меню **Вставка** → к. **Создать слайд (Новый слайд)** → вибрати шаблон **Маркированный список** → **ОК**. Для вставки нового слайда можна скористатися також кнопкою **Создать слайд** на панелі інструментів **Стандартная** чи клавішами **[Ctrl]+[M]**.

Буде вставлений новий слайд, оформлений відповідно до обраного раніше шаблону презентації, вміст якого буде відповідати обраній розмітці слайда (рисунок 5.4).

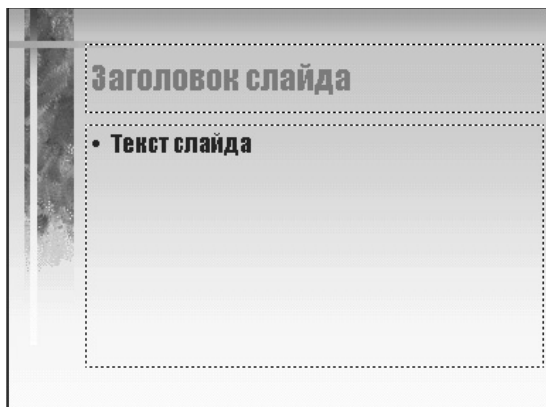




Рисунок 5.4 – Макет слайда з маркірованим списком

Увести текст заголовка – Комп’ютерні технології. Для заміни заголовка в основній області слайда на текст варто клацнути по ній мишею і ввести потрібний текст (рисунок 5.5). Після натискання клавіші [Enter], буде створений новий елемент маркірованого списку на цьому ж рівні. Можна користуватися клавішами [Tab] і [Shift]+[Tab] чи кнопками панелі інструментів **Форматирование** **Понизить уровень** і **Повысить уровень**   для того, щоб оформити маркірований список потрібним чином.

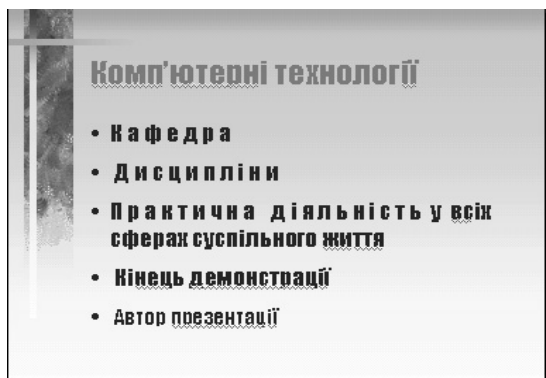


Рисунок 5.5 – Сформований текстовий слайд із маркірованим списком

## Таблиці

У створювані слайди можна впроваджувати найрізноманітніші об'єкти, які підготовлені засобами Microsoft Office і інших додатків. Ось далеко не повний перелік: діаграми *Microsoft Graph*, фігурний текст *Microsoft WordArt*, організаційні діаграми (блок-схеми) *Microsoft Organization Chart*, робочі аркуші *Excel*, таблиці *Word*. Крім того, можна прикрасити слайд додатковими малюнками, створеними за допомогою панелі інструментів **Рисование** PowerPoint.

Щоб уставити слайд із таблицею необхідно:

м. **Вставка** → к. **Создать слайд (Новый слайд)** → вибрати шаблон **Таблица** → **ОК**.

Увести заголовок слайда – “Склад частин презентації” і клацнути мишею по рамці заглушки. З'являться маркери маніпулювання розмірами майбутнього (ще не створеного) об'єкта. Можна розтягти рамку заглушки так, щоб вона займала майже усе вільне місце на слайді, а потім зробити подвійне клацання по заглущі таблиці → указати кількість рядків і стовпців → **ОК** → заповнити чарунки таблиці (рисунок 5.6). При заповненні варто використовувати більш великий розмір шрифту, чим у Word. У прикладі таблиці в заголовній частині використаний шрифт Times New Roman Cyr розміру 28, а в предметній частині – розміру 24 (установлюється за допомогою кнопок панелі **Форматирование**).

Для вирівнювання тексту в чарунках необхідно виділити останні, а потім для Office 97:

м **Формат** → к. **Абзац** → вкладка **Выравнивание** → вибрати **По центру** → **ОК** чи відповідна кнопка на панелі інструментів **Форматирование** → контекстне меню → к. **Выравнивание** → п. **Центрировать по вертикали**. Для Office 2000: м. **Формат** → к. **Таблица** → вкладка **Надпись** → у полі **Выравнивание текста** вказати **Посередине по центру** → **ОК**.

Склад частин презентації		
Кафедра	Дисципліни	Практична діяльність у всіх сферах суспільного життя
Історичний аспект	Область народження	Навчання
Керівництво	Місце серед інших дисциплін	Робота
Сучасний стан	Можливості	Відпочинок

Рисунок 5.6 – Слайд із таблицею

Клацання поза рамкою таблиці дозволить побачити дійсний вид слайда. Якщо таблиця чи її частини видно погано, то необхідно змінити колір шрифту, ліній, заливання. Виконати подвійне клацання по місцю розташування таблиці, щоб повернутися в режим її редагування. Виділити таблицю → м. **Формат** → к. **Шрифт** → у полі **Цвет** вибрати потрібний колір букв → **ОК**.

Для зміни кольору ліній і заливання в Office 97 вибрати м. **Формат** → к. **Границы и заливка** → вкладка **Границы** → указати тип і колір ліній → вкладка **Заливка** → указати колір і спосіб заливання → **ОК**. У Office 2000 –

м. **Формат** → к. **Цвета и линии** → вкладка **Границы** → указати тип і колір ліній → вкладка **Заливка** → указати колір і спосіб заливання → **ОК**.

### 5.3 Представлення презентації

Презентація – це насамперед слайди з текстом, малюнками і впровадженими об'єктами. Однак, матеріали презентації мають ще й інші форми представлення: структура презентації, роздавальні матеріали, сторінки заміток, сортувальник слайдів і показ слайдів. Переключатися з одного режиму представлення презентації в інший можна за допомогою команд меню **Вид** або відповідних кнопок (для режиму роздавальних матеріалів кнопки немає), які розташовані у лівому нижньому куті вікна презентації.

Уся текстова інформація, що є на слайдах презентації, може бути подана у вигляді *структури презентації* (рисунок 5.7). Працюючи зі

структурою, можна бачити і редагувати цю текстову інформацію – як текст заголовків, так і основний текст слайдів із відповідними відступами (але не написи на малюнках і впроваджених об'єктах). При цьому можна доповнювати текстовий зміст слайдів і змінювати рівень підпорядкованості маркірованих списків. Усі зроблені зміни тексту слайдів презентації відбиваються й у всіх інших її представленнях: безпосередньо в структурі, на самих слайдах, на відповідних роздавальних матеріалах і сторінках заміток.

Структуру презентації можна доповнити новими текстовими слайдами. Після натискання клавіші **[Enter]** у структуру презентації вводиться ще один рядок на тім же самому рівні структури, що і попередній. Можна змінити рівень чергового рядка в структурі: помістити крапку введення у відповідний рядок → змінити відступ цього рядка за допомогою кнопок **Повысить уровень** і **Понизить уровень** на панелі інструментів **Форматирование** або **Структура**. Натиснувши комбінацію клавіш **[Ctrl]+[Enter]**, можна перейти до створення нового слайда з будь-якого рівня структури. У такий спосіб зручно створити всі заголовки слайдів і весь основний текст на них. Як і в режимі слайдів, у режимі структури можна модифікувати стиль тексту, змінюючи шрифт, що використовувався, його розмір чи накреслення. Наприклад, можна виділити ключове слово курсивом або використовувати напівжирний шрифт. Однак у режимі структури не можна змінити колір тексту чи задати використання тіні, такі зміни можна зробити тільки в режимі роботи зі слайдами.

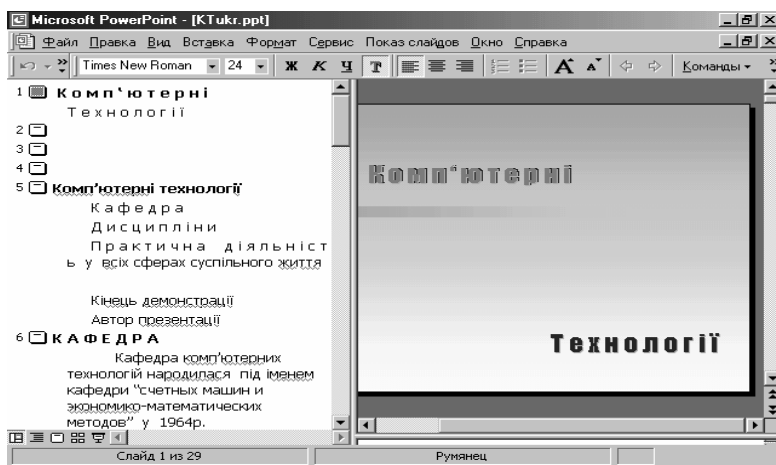


Рисунок 5.7 – Структура презентації

Пакет PowerPoint дозволяє представити презентацію у виді слайдів із замітками. Кожен слайд презентації розміщується у верхній частині *сторінки заміток*, а в нижній частині цього листа можуть розташовуватися досить докладні коментарі до цього слайда, які можна використовувати під час виступу (рисунок 5.8).

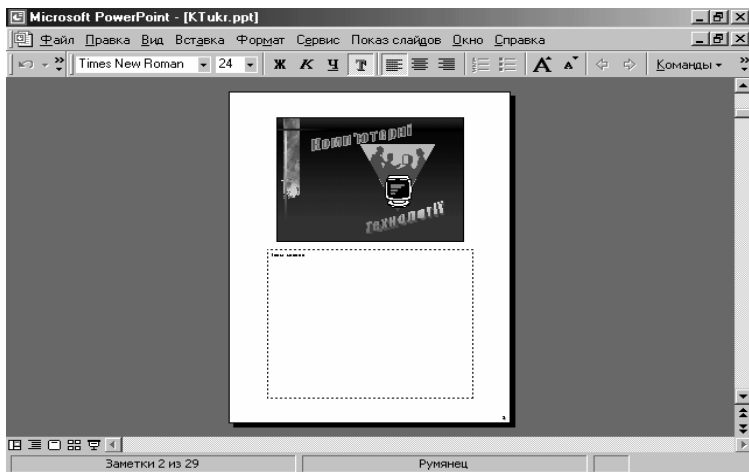


Рисунок 5.8 – Режим «Страницы заметок»

Перед тим як вводити текст заміток у цьому режимі, варто попередньо збільшити масштаб зображення для того, щоб користувач міг краще бачити текст, що вводиться. На стандартній панелі інструментів для цього є спеціальний список **Масштаб**, що розкривається. Текст буде нормально відображатися і видимим для читання в масштабі 60% – 80%. Можна скористатися для цього і командами меню: м. **Вид** → к. **Масштаб** → 66% → **ОК**.

Текст, введений у сторінки заміток, ніяк не відбивається на змісті слайда. Змінити в цьому режимі самі слайди не можна. Можна, однак, змінити оформлення сторінок заміток за допомогою м. **Вид** → к. **Образец** → п. **Образец заметок**.

Зразок заміток дозволяє змінити відносні розміри відображення слайда й області заміток, використовувані шрифти, помістити на сторінки заміток додаткові елементи оформлення, наприклад, колонтитули.

Для відповідального виступу обов'язково акуратно приготуйте й роздрукуйте сторінки заміток за допомогою м. **Файл** → к. **Печать** → у

полі **Печать** вибрати **Заметки** → **ОК**. Зовсім не обов'язково читати текст по папірцях, важливо, щоб ці папірці були.

**Режим сортировщика слайдов** (рисунок 5.9) призначений для роботи з презентацією в цілому, а не з окремими слайдами. У цьому режимі можна змінити порядок проходження слайдів презентації, задати відео ефекти, які використовуються при переході до наступного слайду або при заповненні слайда об'єктами.

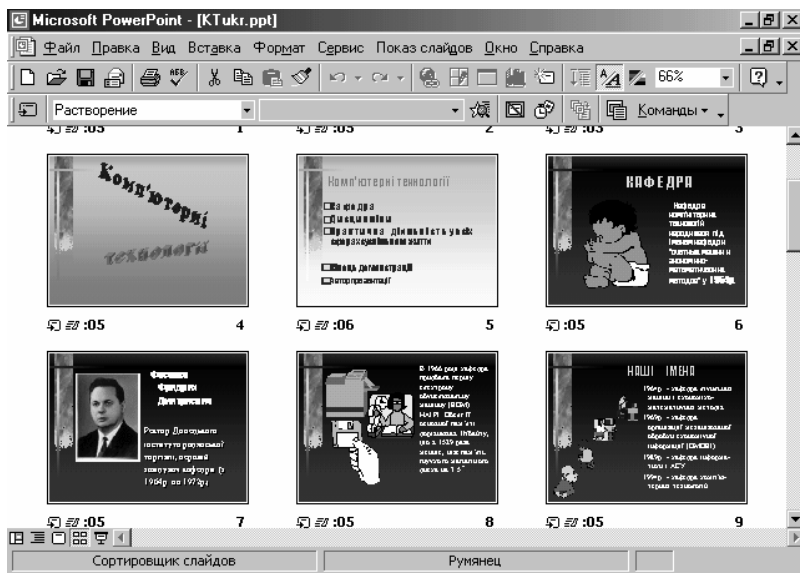



Рисунок 5.9 – Презентація в режимі “сортировщика слайдов”

У цьому ж режимі можна також визначити деякі слайди як сховані. На панелі інструментів **Сортировщик слайдов** є для цього спеціальна кнопка **Сховати слайд** . Сховані слайди не виводяться на екран при звичайному перегляді презентації. Проте є можливість при необхідності вивести на екран і сховані слайди: клацнути правою кнопкою миші в будь-якому місці попереднього слайда → к. **Переход** → п. **Схований слайд**. Таким чином, можна підготувати деякі додаткові презентаційні матеріали, що можуть бути використані чи пропущені в залежності від зацікавленості й реакції аудиторії.

Крім того, кнопка **Режим сортировщика слайдов** може служити не тільки для переходу в сам режим сортувальника слайдів, але і для ви-



клику режиму *образца раздаточных материалов*. Його виклик здійснюється клацанням по цій кнопці при натиснутій клавіші [Shift], чи за допомогою:

м. Вид → к. Образец → п. Образец раздаточных материалов (Образец выдач для Office 2000).

Роздавальні матеріали являють собою аркуші стандартного формату, на яких розміщені одночасно від 2 до 9 слайдів презентації. У режимі оформлення роздавальних матеріалів можна помістити додаткові текстові або графічні елементи на аркуші роздавальних матеріалів. Для того щоб можна було правильно розрахувати місце, де варто розташовувати додаткову інформацію, на аркуші пунктиром відображаються прямокутники, у які можуть бути поміщені слайди презентації.

Нарешті, остання кнопка в лівому нижньому куті вікна презентації служить для *показа слайдов*. У цьому режимі на екрані комп'ютера виводяться лише слайди підготовленої презентації.

Клацання лівою кнопкою миші забезпечує перехід до показу наступного слайда або наступного елемента слайда першого рівня, якщо в режимі сортувальника слайдів були визначені спеціальні відеоефекти побудови слайдів. Клацання правою кнопкою миші служить для виклику контекстного меню. Для переходу від слайда до слайда можна використовувати і клавіатуру – клавіша [Page Down] викликає перехід на наступний слайд, а клавіша [Page Up] – на попередній. Таку ж дію роблять клавіші зі стрілками. Клацання по наявній у лівому куті кнопці також розкриває контекстне меню, у якому можна вибрати потрібну команду.

Контекстне меню дозволяє виконати під час показу слайдів усякого роду допоміжні дії. Як уже згадувалося, воно використовується для показу схованих слайдів. Можна також перейти відразу до потрібного слайда за допомогою к. Переход → п. Выбор слайда по имени, замінити зображення поточного слайда чорним екраном і зворотно, вивести на екран хронометр, щоб стежити за часом. Команда **Перо** переводить мишу в режим малювання – користувач може зробити на слайді деякі тимчасові позначки безпосередньо під час презентації – наприклад, підкреслити чи обвести ключове поняття або іншим способом виділити потрібний фрагмент слайда і навіть (якщо, звичайно, ви досить спритно володієте мишею) зробити додатковий напис. Нарешті, натискання клавіші [Esc] чи клацання мишею на останньому слайді завершують показ слайдів.

## 5.4 Створення витонченої презентації

Нові інформаційні технології породили нову якість поширення інформації, віддавши всю міць інформаційних засобів телебачення у владу користувача. Тепер мандрівник по World Wide Web може, подібно читачу звичайної книги, перегорнути сторінку Web, подібно відвідувачу виставки, залишити свої зауваження в "книзі відгуків", нарешті, прокоментувати побачене, відправивши електронне повідомлення своєму приятелю.

У визначеній мірі ці можливості відбиваються в презентації, яку можна зробити за допомогою *PowerPoint*. Попередній матеріал по даному питанню дає засоби для створення примітивної презентації, що мало чим відрізняється від надрукованого на аркушах тексту. Але *PowerPoint* дозволяє створити презентацію, у якій використовуються можливості невимовні поліграфічними засобами, такі як рух, музичний супровід, озвучування, вставка кліпів.

### Форматування слайдів

Як було показано раніше, обрана розмітка слайдів використовується для всіх слайдів презентації. Виділити частини презентації можна за допомогою використання нової колірної гами. Для цього: м. **Формат** → к. **Цветовая схема слайда** → вкладка **Специальная** → виділити елемент колірної схеми (Фон, Текст і лінії і т.д.) → кнопка **Изменить цвет** → вибрати бажаний колір у палітрі кольорів → при необхідності повторити теж саме для інших елементів схеми → кнопка **Применить** (для одного слайда) чи **Применить ко всем**.

У деяких випадках чудового ефекту можна домогтися, помістивши на слайд як фон який-небудь графічний файл придатний за змістом (рисунок 5.10). Щоб графічне зображення не закривало текст, варто викликати контекстне меню → к. **Порядок** → п. **На задний план**.

### Анімація

Слово *анімація* буквально означає "оживлення". Насправді під цим терміном мається на увазі рух об'єктів на екрані. *PowerPoint* має кілька різних засобів анімації.

Першим є анімаційний ефект при переході до наступного слайда. Цей ефект може бути заданий (чи не заданий) для кожного слайда окремо чи для всіх слайдів відразу (рисунок 5.11): м. **Показ слайдов** → **Переход слайда (Смена слайдов)** → у списку рамки **Ефект**, що випадає, вибрати, наприклад, **Растворение** й установити потрібну селекторну кнопку, наприклад, **медленно** → у рамці **Продвижение** установити **прапорці по щелчку й автоматически после 00:05** секунд → у рамці **Звук** у списку, що випадає, вказати ім'я звукового файлу і при необхідності виставити прапорець **непрерывно** → кнопка **Применить** (для одного слайда) чи **Применить ко всем**.



Рисунок 5.10 – Використання файлу фотографії як фон слайда

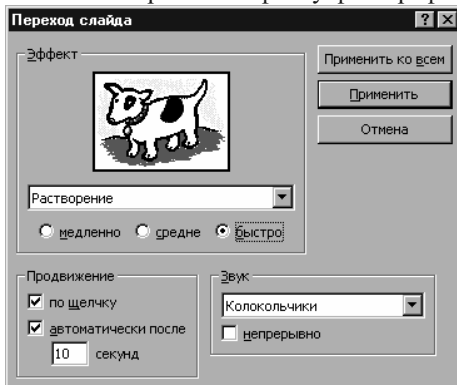


Рисунок 5.11 – Діалогове вікно установки параметрів анімації при зміні слайдів

Другий анімаційний ефект зв'язаний із процесом побудови слайдів зі складових його об'єктів. Для кожного об'єкта слайда ефект можна задати по окремоті (рисунок 5.12): м. **Показ слайдов** → к. **Настройка анимации** → вкладка **Порядок и время** → у списку **Объекты для анимации** виставити прапорець для першого об'єкта, що рухається, наприклад, для титульного слайда (рисунок 5.3) це **WordArt: склон вверх1**, при цьому він з'явиться в списку **Порядок анимации** → виставити селекторну кнопку **автоматически**, через 01 секунд після попередньої події → аналогічно зробити з іншими об'єктами слайда, для яких варто задати ефекти анімації.

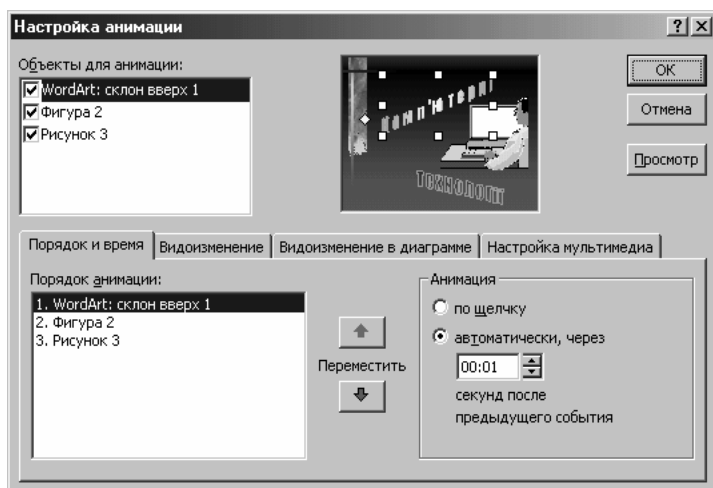


Рисунок 5.12 – Діалогове вікно для настроювання анімації Office 2000

Змінити порядок анімації об'єктів можна за допомогою кнопок із стрілками праворуч від списку **Порядок анимации**.

Визначення ефектів анімації для кожного об'єкта: вкладка **Эффект (Видоизменение)** → вказати об'єкт у списку **Объекты для анимации** → у рамці **Выберите эффект и звук** у списках, що випадають, вибрати ефект, спосіб його появи і звуковий файл супроводу, наприклад, для першого об'єкта титульного слайда це **будуть Выполнение, Снизу, Камера** → аналогічно зробити з іншими об'єктами слайда, для яких варто задати ефекти анімації → **ОК**.

Якщо об'єктом є текст, то будуть доступні елементи рамки **Появление текста**, які також можна настроювати. При настроюванні анімаційних ефектів рекомендується активно користатися кнопкою **Промотр** для візуального контролю дій, що задаються.

Третій прийом анімації полягає у вставці активного об'єкта, що може жити власним життям, наприклад, звукового чи відеокліпу. Вставка відеокліпу виконується аналогічно вставці будь-якого іншого об'єкта: м. **Вставка** → к. **Кино и звук (Фильми и звук)** → п. **Фильм из файла** → указати папку й ім'я файлу з фільмом → **ОК**.

Основна відмінність об'єктів із власною анімацією, зокрема відеокліпів, від інших об'єктів полягає в тому, що для них має сенс поняття **відтворення**. Параметри відтворення визначаються на однойменній вкладці (**Настройка мультимедіа**) діалогового вікна **Настройка анимации**.

### Звукове оформлення

Насамперед, варто усвідомлювати те, що для роботи зі звуком комп'ютер повинний бути обладнаний звуковою картою, звуковими колонками й мікрофоном (для запису розмови). Узагалі, *PowerPoint* дозволяє просто записати все повідомлення, яким доповідач хотів би супроводити показ слайдів. Така презентація може бути переглянута й прослухана користувачем у будь-який зручний для нього час.

У попередньому параграфі при настроюванні анімації вже були установлені звукові ефекти для об'єктів, що рухаються. Однак це не дає цільного звукового оформлення всієї презентації. *PowerPoint* дозволяє накласти на слайди звукову доріжку, тобто забезпечити всі чи деякі слайди досить вигадливим музичним фоном. Для цього потрібно, поперше, установити надбудову *Microsoft PowerPoint Custom Soundtrack*: м. **Сервис** → к. **Надстройка** → кнопка **Загрузить (Добавить)** → указати CD-диск і **ppmusic** → **Закрити (ОК)**.

Після цього виберіть м. **Показ слайдов** → к. **Custom soundtrack.....** З'явиться діалогове вікно (рисунок 5.13), за допомогою численних списків, що розкриваються, цього вікна потрібно підібрати придатний до даного слайда музичний супровід. Попередньо прослухати його можна за допомогою кнопок **Sample** і **Sample music**.

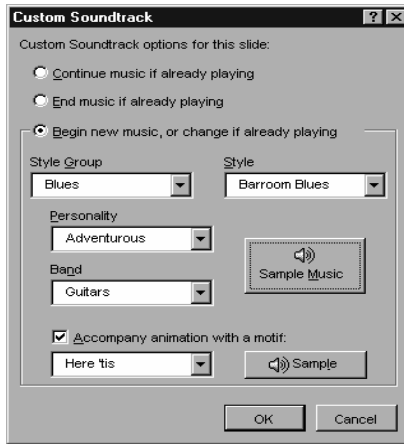


Рисунок 5.13 – Діалогове вікно Custom Soundtrack для вибору фонової музики

Якщо була задана фонова музика для одного слайда, то варто зробити це і для всіх інших, тому що раз почавшись, фонова музика буде продовжувати грати і при показі наступних слайдів. Якщо зміст слайда вимагає тиші, то виберіть перемикач **End music if already playing**.

В останній слайд приклада презентації включений власний звукозапис доповідача. Це робиться в такий спосіб:

підготувати текст → м. **Вставка** → к. **Кино и звук (Фильми и звук)** → п. **Записать звук** → у діалоговому вікні

**Звукозапис** (рисунок 5.14) кнопка **Запись** (з червоним кружком) → проговорити текст у мікрофон → кнопка **Стіл** (із чорним квадратиком) →

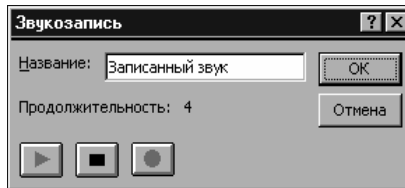


Рисунок 5.14 – Діалогове вікно **Звукозапись**

прослухати отриманий запис за допомогою кнопки **Воспроизведение** (із сірим трикутником) → якщо усе в порядку, то **ОК**.

Якщо якість запису не задовольняє, то можна виконати його на-строювання: м. **Показ слайдов** → к. **Звукозапис** → у діалоговому вікні, що з'явилося, поміняйте параметри звукозапису.

При включенні звукозапису на слайді з'являється значок записано-го кліпу у виді гучномовця. Щоб він не плутався з іншими об'єктами, перетягніть його в лівий нижній кут. При демонстрації слайдів можна відтворити кліп, клацнувши по цьому значку. Для того щоб домогтися автоматичного відтворення, потрібно вказати параметри анімації цього об'єкта, наприклад, так, як зазначено на рисунку 5.15.

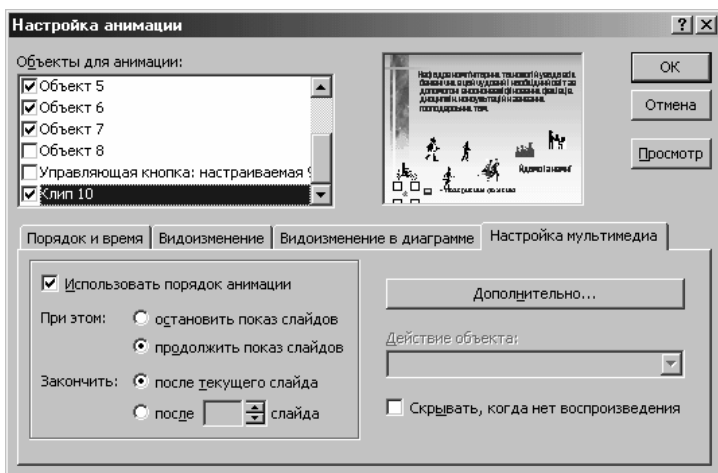


Рисунок 5.15 – Діалогове вікно для настроювання автоматичного відтворення звукозапису в Office 2000

## Дії

Презентацію, створену в результаті вище описаних дій, користувач може переглядати за допомогою клацання миші. Для її автоматичного відтворення необхідно виконати:

м. **Показ слайдов** → к. **Настройка презентации** → у діалоговому вікні зробити настроювання відповідних параметрів – у рамці **Показ слайдов** установити селекторну кнопку **автоматический** → у рамці **Слайди – Усе** → у рамці **Смена слайдов – по времени** → **ОК** (рисунок 5.16).

Примітка – Слайди, позначені як сховані, при автоматичному показі пропусаються. Зупинка автоматичної демонстрації – [Esc].

Якщо презентація – це просто послідовність слайдів, яку можна тільки пасивно переглянути, то це не цікаво. Потрібно надати можливість слухачу активно впливати на процес демонстрації презентації, і такі засоби є в *PowerPoint*.

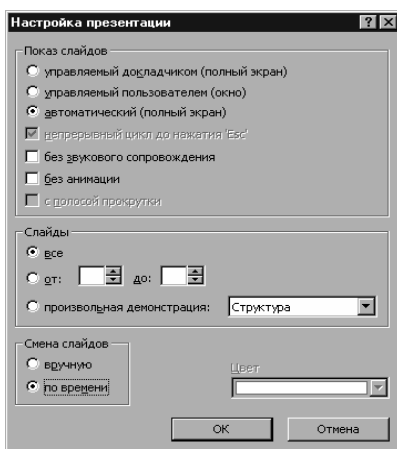


Рисунок 5.16 – Діалогове вікно для налаштування автоматичної демонстрації презентації

Перший засіб називається *разделы (показы) презентации*. Він дозволяє створювати іменовані послідовності слайдів презентації, до яких слухач чи доповідач може звертатися. Можна сказати, що розділ є презентацією в презентації. Розділи створюються й змінюються за допомогою: м. **Показ слайдов** → к. **Произвольная демонстрация (Произвольный показ)** → кнопка **Создать** → у діалоговому вікні, що відкрилося, дати ім'я розділу (показу) → у **списку Слайды презентации** послідовно виділяти потрібні слайди → кнопка **Добавить** для перенесення в список **Слайды демонстрации** → **ОК**.

Аналогічним образом можна змінити довільну демонстрацію.

Примітка – Розділи можуть перетинатися, тобто ті самі слайди можуть входити в кілька розділів. Слайди в розділі можуть йти в будь-якому порядку, безвідносно до їх порядку у початковій презентації.



Довільна презентація має очевидне застосування: під час показу основної презентації можна в будь-який момент перейти до показу довільної презентації (розділу): контекстне меню → к. **Произвольная презентация (показ)** → указати ім'я розділу (показу).

Можливість переходу до довільної демонстрації і створення схожих слайдів дозволяє доповідачу заздалегідь підготувати кілька сценаріїв показу і гнучко варіювати їх відповідно до ходу показу й реакції аудиторії.

Другий засіб називається *елементи управління*. PowerPoint дозволяє поміщати на слайди різноманітні елементи керування: кнопки, прапорці, списки, перемикачі і т.д.

Елементи керування можна помістити на слайд різними способами: часто використовувані елементи керування – готові кнопки – найпростіше вставити за допомогою палітри кнопок, що з'являється по м. **Показ слайдов** → к. **Управляющие кнопки**, а більш складні краще вставляти з допомогою панелі інструментів **Элементы управления**. Перший елемент у стандартній палітрі кнопок найцікавіший – він дозволяє вставити кнопку, що настроюється. При виборі кнопки на палітрі покажчик миші приймає вид тонкого хрестика – інструмента малювання. Варто намалювати кнопку в потрібному місці слайда. Після цього з'явиться діалогове вікно (рисунок 5.17), у якому необхідно вибрати вкладку **По щелчку мыши** → селекторну кнопку **Перейти по гиперссылке** → клацнути по назві довільної демонстрації (показу) → указати у вікні, що відкрилося, ім'я розділу (показу) → виставити **прапорець Показать и вернуться** → **ОК** → **ОК**.

Примітка – Дія може бути зв'язана з будь-яким об'єктом на слайді. Щоб задати дію, потрібно виділити об'єкт, клацнути правою кнопкою миші і вибрати в контекстному меню команду **Настройка действия**.

Показ довільної презентації є тільки одним із видів дій. Набір можливих дій залежить від типу об'єкта. Можна зв'язати з елементом керування (чи іншим об'єктом) запуск додатка.

Наступний засіб, за допомогою якого користувач може вплинути на хід показу презентації, – гіперпосилання. Гіперпосилання можна зв'язати з будь-яким об'єктом на слайді, але найчастіше його зв'язують із деяким текстом. Текст, із яким зв'язане гіперпосилання, виділяється з навколишнього тексту (зокрема, підкресленням). При влученні покажчика миші на гіперпосилання він приймає форму вказівного пальця.

Якщо клацнути по гіперпосиланню, то відбудеться перехід до того об'єкта, на який воно вказує. Процедура вставки гіперпосилання аналогічна такій же процедурі в *Word*: виділити об'єкт м. **Вставка** → к. **Гіперссылка** → вказати об'єкт, що викликається ( слайд поточної презентації, файл або документ додатку) → **ОК**.

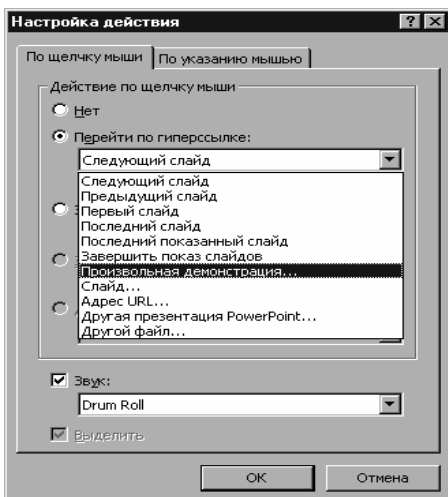


Рисунок 5.17 – Діалогове вікно **Настройка действий**.

Додані на слайди презентації елементи керування й гіперпосилання дозволять користувачу інтерактивно втрутитися в процес показу презентації, якщо він захоче.

## 5.5 Переміщення і демонстрація презентації на інших ПК

Звичайно презентації створюються для демонстрації, тому в остаточному підсумку виникне питання про їхній перенос на інші машини або демонстрацію відразу на декількох машинах. Якщо врахувати, що в презентацію впроваджені різні об'єкти, і що презентація складається не з одного файлу, то виникає досить багато технічних питань. Щоб не утрудняти себе щоразу їхнім рішенням, потрібно скористатися **Мастером упаковки**: м. **Файл** → к. **Упаковать** → кнопка **Далее** → вистави-

ти прапорець **Активную презентацию** → кнопка **Далее** → виставити прапорець **Включить связанные файлы і Внедрить шрифты TrueType** → кнопка **Далее** → виставити селекторну кнопку **Включить PowerPoint Viewer** → кнопка **Далее** → кнопка **Готово**.

Якщо **Мастер упаковки** повідомить, що не може знайти файла *ppview32.exe*, то не звертайте на це уваги. Цього файлу дійсно може не бути на твердому диску комп'ютера, тому що він не встановлюється за замовчуванням. Його необхідно скопіювати з інсталяційного диска або працювати без нього.

Для розпакування презентації варто запустити з першої дискети архіву програму *pngsetup.exe* і вказати папку, у яку потрібно розпакувати презентацію. Після цього презентацію можна показувати.

#### Примітка -

1. Програма *PowerPoint Viewer* (файл *ppview32.exe*) використовується для демонстрації презентації без самої програми *PowerPoint*.
2. Зовнішні додатки, що викликаються за допомогою кнопок на слайдах PowerPoint, упаковані не будуть. Їх потрібно переписати й перенести окремо.

Розвиток комп'ютерних комунікацій, повсюдне розповсюдження мереж ПК дозволяє демонструвати презентацію з робочого місця доповідача. Для цього в PowerPoint є **Мастер конференци** (у Office 97), що дозволяє з комп'ютера доповідача синхронно провести показ слайдів на декількох комп'ютерах учасників конференції. У Office 2000 передбачені більш широкі можливості по роботі в мережах: м. **Сервис** → к. **Совместная работа**.

## Питання й завдання для самостійного виконання

1. У яких галузях і з якою метою можна використовувати пакет *PowerPoint*?
2. У чому відмінність презентації від звичайної доповіді?
3. Призначення розміток слайда?
4. Анімація – що це і для чого?
5. Якими оформлювальними засобами можна виділити різні частини презентації?
6. Як зробити користувача (слухача) активним учасником презентації?
7. Чи можна переносити презентацію з одного комп'ютера на інший і як?
8. Чи можлива демонстрація презентації відразу на декількох ПК?
9. Створити презентацію про свою майбутню спеціальність.
10. Створити презентацію, що рекламує Ваш факультет.
11. Створити презентацію з результатами освоєння матеріалів і пакетів, вивчених по даному посібнику.

## Література

1. Войтюшенко Н.М., Остапець А.І. Комп'ютерна техніка і програмування. Частина 1. Основи підготовки користувача ПК: Навчальний посібник по базовій підготовці для студентів денної і заочної форм навчання. – Донецьк, ДонДУЕТ, 2001 – 150 с.
2. Компьютеры, сети, Интернет. Энциклопедия./Новиков Ю и др. – СПб.: Питер, 2002. – 928 с.
3. Новиков Ф., Яценко А. Office 97 в целом. – СПб: BHV – Санкт-Петербург, 1998 – 624с.
4. Хэлворсон М., Янг М. Эффективная работа с Microsoft Office 2000 – СПб: Издательство «Питер», 2000. – 1232 с.

## Розділ 6. Комп'ютерні мережі і мережні технології

Сьогодні вже нікого не здивуєш словами «комп'ютерна мережа», багато хто в ній уже виконував якісь операції і переконався, що це досить просто. Але користувач, що ввійшов у мережу, бачить тільки «вершину айсберга». Для того щоб ефективно працювати з комп'ютерними комунікаціями, необхідно мати уявлення про те, що це, як воно працює, і що можна одержати за допомогою комп'ютерних мереж.

**Комп'ютерна мережа** являє собою сукупність комп'ютерів, об'єднаних засобами передачі даних.

Найважливішими характеристиками мережі є: швидкодія, пропускна здатність, вірогідність, надійність.

Наприклад, сучасні широкополосні канали забезпечують швидкість передачі даних 1 Гбіт/с і більше. Вірогідність переданої інформації повинна бути не більш  $10^{-6} - 10^{-7}$  помилок на один знак.

Сучасні мережі можна класифікувати по різних ознаках:

- ❖ по далькості комп'ютерів – локальні, регіональні, глобальні;
- ❖ по топології – статичні, у яких структура взаємозв'язків фіксована; динамічні, у яких конфігурація взаємозв'язків може бути оперативно змінена програмними засобами;
- ❖ призначенню – мережі загального призначення і корпоративні;
- ❖ принципам керування – централізовані й децентралізовані;
- ❖ видам середовища передачі і т.д.

Основою більш великих мережних формувань є локальні мережі, від особливостей побудови яких багато в чому залежать і способи подальшого підключення й функціонування в регіональних і глобальних мережах.

## 6.1 Локальні комп'ютерні мережі

### 6.1.1 Локальні комп'ютерні мережі і їхні можливості

**Локальна комп'ютерна мережа (ЛКМ)** – мережа передачі даних, що зв'язує ряд комп'ютерів, що знаходиться на обмеженій площі (максимум 10 км у кабельних мережах і до 20 км із радіозв'язком). Користувачі можуть працювати в автономному режимі, розділяти ресурси комп'ютера й інформацію, можуть спільно працювати над проектами і задачами, що вимагають тісної координації й взаємодії.

Можливості ЛКМ:

1. Поділ файлів. ЛКМ дозволяє багатьом користувачам одночасно працювати з одним файлом, що зберігається на центральному файлі-сервері. Якщо це авто-номна програма (наприклад, текстовий редактор), то користувачі одночасно не можуть працювати з тим самим документом.

2. Передача файлів. ЛКМ дозволяє швидко копіювати файли будь-якого розміру з однієї машини на іншу без використання дискет.

3. Доступ до інформації й файлів. ЛКМ дозволяє запускати прикладні програми з будь-якої робочої станції, де б вона не була розташована.

4. Одночасне введення даних у прикладні програми. Мережні прикладні програми дозволяють декільком користувачам одночасно вводити дані, необхідних для роботи цих програм. Однак це повинні бути спеціальні мережні версії програм. Звичайні програми дозволяють працювати з набором файлів тільки одному користувачеві.

5. Поділ принтера й інших ресурсів ПК. Декілька користувачів на різних робочих станціях можуть спільно використовувати один лазерний принтер, стример, можна використовувати вільний дисковий простір іншого комп'ютера і т.д.

6. Електронна пошта. Можна використовувати ЛКМ як поштову службу і розсилати службові записки, доповіді, повідомлення іншим користувачам навіть у випадку відсутності абонента на місці, і для цього не потрібен папір.

З погляду структурних компонентів, ЛКМ – це сукупність комп'ютерів, кабелів, мережних адаптерів, апаратури для комутації, що працюють під керуванням мережної операційної системи і прикладного програмного забезпечення.

Комп'ютери, що входять у ЛКМ, поділяються на два типи: **робочі станції**, призначені для користувачів, і **файлові сервери**, що, як правило, не доступні для звичайних користувачів.

**Файловий сервер** – це комп'ютер, що обслуговує всі робочі станції. Він здійснює спільне використання файлів, розташовуваних на його дисках. Файлові сервери – це швидкодіючі комп'ютери, що оснащуються швидкодіючими високонадійними накопичувачами великої ємності, наприклад, знімними пакетами дисків. Сервери повинні бути високоякісними і високонадійними машинами, тому що при обслуговуванні комп'ютерної мережі вони багаторазово виконують роботу звичайної робочої станції.

Кожна робоча станція й файл-сервер містять карти мережних адаптерів, що за допомогою мережних кабелів з'єднуються між собою. Робоча станція відправляє запит через мережний адаптер до файлового сервера й одержує відповідь через мережний адаптер, коли файловий сервер готовий передати чергову порцію файлу. Незалежно від типу адаптери за підтримкою програмного забезпечення виконують 7 основних операцій при прийманні й передачі повідомлень:

1. Передача даних. Дані передаються з ОЗП ПК в адаптер або у зворотному порядку через програмувальний канал, канал прямого доступу до пам'яті або поділювану пам'ять.
2. Буферізація. Під час обробки, у мережному адаптері дані зберігаються в буфері. Буфер потрібний для узгодження швидкостей обробки даних різними компонентами ЛКМ.
3. Формування пакета. Мережний адаптер повинен розділити дані на порції (або при прийманні з'єднати їх) і додати до пакета заголовки і закінчення. Після цього пакет готовий до передачі.
4. Доступ до кабелю при передачі пакета. Адаптер чекає сигналу звільнення мережі (захоплення маркера, приймання посланого раніше пакета...).

5. Перетворення даних. Дані передаються в кабель у послідовній формі, біт за бітом. Тому перед початком передачі вони повинні бути перетворені з рівнобіжної форми в послідовну. При прийманні повідомлень – навпаки.
6. Кодування/декодування даних. На цьому етапі формуються електричні сигнали, використовувані для представлення даних.
7. Передача/приймання імпульсів. Електричні сигнали даних передаються в кабель.

Карта адаптера приймає всі повідомлення, що передаються по кабелю, і відбирає ті з них, що безпосередньо адресовані даній робочій станції. Отримане повідомлення затримується адаптером доти, поки робоча станція не буде готова прийняти це повідомлення. У випадку, коли робоча станція збирається надіслати запит на файловий сервер, адаптер чекає перерви в потоці даних (трафіку) мережі і виставляє своє повідомлення у вікно, що утворилося. Адаптер також автоматично проводить перевірку того, що повідомлення було передано без помилок і якщо це не так, то передача повторюється.

З'єднання комп'ютерів у мережі здійснюється за допомогою кабелів. Існує безліч модифікацій мережних кабелів (тонкі й товсті коаксіальні, екрановані і неекрановані виті пари, оптоволоконні). Тип кабелю в значній мірі залежить від обраної карти адаптера. Питання про прокладку кабелю в мережі й монтажі з'єднань досить серйозне, тому що від нього істотно залежить якість передачі даних і робота всієї мережі в цілому. Використовуються також безпроводні канали передачі даних між комп'ютерами, що використовують радіохвилі.

### **6.1.2. Топологія ЛКМ і методи доступу до каналів зв'язку**

Комп'ютери, що працюють у мережі, можуть з'єднуватися між собою різним способом.

**Топологія мережі** – опис способів фізичного з'єднання робочих станцій і серверів. Топологія багато в чому визначає найважливіші характеристики мережі надійність, продуктивність, вартість, захищеність і т.д.



Способів з'єднання робочих станцій і сервера безліч. До статичних топологій відносяться такі, у яких між вузлами можливий тільки один прямий фіксований шлях. Прикладами статичної топології слугують: лінійна шина, зірка, дерево, кільце.

Нижче розглянуті «базові» топології, що мають на практиці безліч модифікацій.

Лінійна топологія (рисунок 6.1) являє собою структуру, у якій кожна робоча станція з'єднується з іншими робочими станціями і з файловим сервером за допомогою загального кабелю.

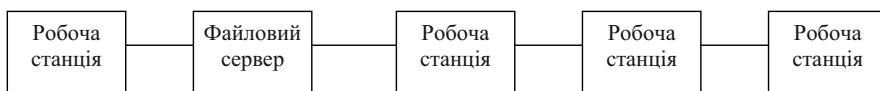


Рисунок 6.1- Лінійна топологія

Однією з переваг такої топології є можливість нарощування мережі без припинення її роботи, недоліком – відмовлення однієї робочої станції здатне привести до неможливості пересилання повідомлень. При роботі мережі такої топології найчастіше використовується метод доступу до каналів зв'язку «виявлення колізій» (зіткнень). Адаптери безупинно знаходяться в стані прослуховування мережі. При необхідності передачі даних робоча станція повинна дочекатися звільнення ЛКМ, і тільки після цього приступити до передачі. Але передача може початися двома або більш робочими станціями ЛКМ. Це і називається колізією. У результаті в мережу може потрапити тільки одне повідомлення, інші робочі станції повинні будуть повторити спробу відправити свої повідомлення, причому повтор здійснюється адаптерами самостійно без утручання робочої програми. Кожен адаптер працює зі своєю частотою, що дозволяє рознести звертання адаптерів до мережі за часом. У результаті всі робочі станції зможуть відправити свої повідомлення. Повідомлення, що йде по мережі, приймають усі робочі станції, і тільки та з них, якої адресоване повідомлення, посилає в ЛКМ підтвердження про приймання. Прикладом такої мережі є система Ethernet.

Якщо загальна шина, до якої приєднуються комп'ютери ЛКМ, замкнута, то одержуємо топологію кільця (рисунок 6.2). Прикладом такої топології є мережа Token Ring.

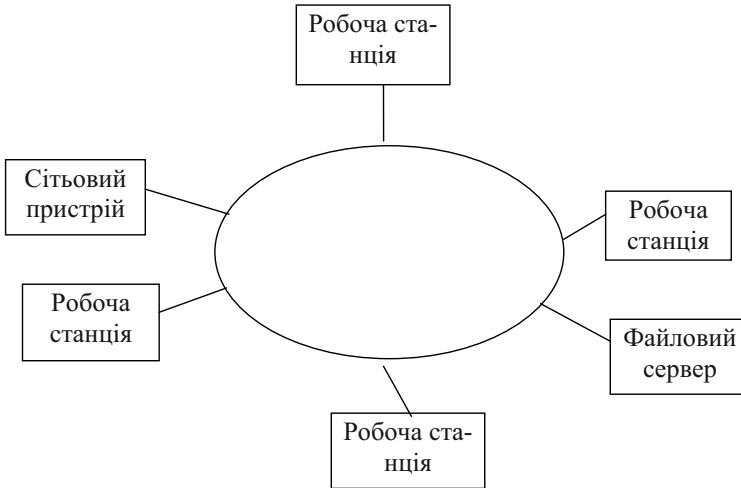


Рисунок 6.2 – Топологія кільця

У випадку, коли до кожної робочої станції або мережного пристрою підходить власний кабель з центрального вузла (звичайно, файлового сервера), конфігурація називається зіркою (рисунок 6.3). Указана топологія добре зарекомендувала себе випадках, коли у якості робочих станцій використовуються термінали.

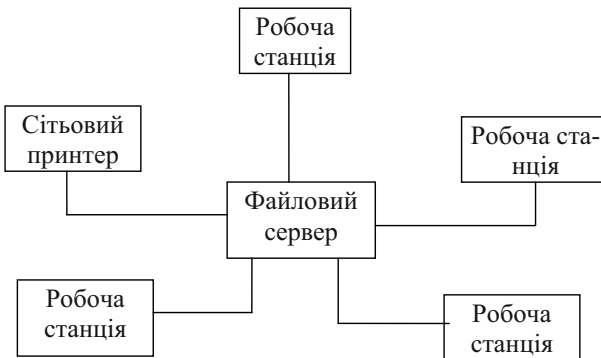


Рисунок 6.3 – Зіркоподібна топологія

Кабелі можуть періодично розгалужуватися у вузлових точках, утворюючи зіркоподібне дерево (рисунок 6.4).

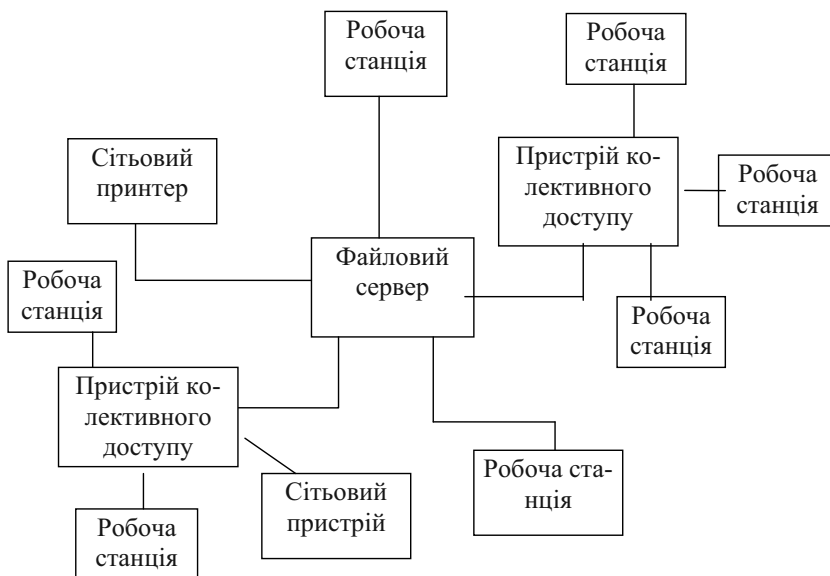


Рисунок 6.4 – Зіркоподібне дерево

У мережах трьох останніх видів, крім принципу дозволу колізій, може використовуватися метод доступу «передачі маркера». ЛКМ із передачею маркера

являє собою кільце. Навіть якщо мережа з'єднана кабелями у вигляді зірки, пакет у ній просувається від робочої станції до робочої станції (вузлу) доти, поки не повернеться в точку, де був породжений. Кожен вузол ЛКМ приймає пакет від верхнього за течією сусіда, відновлює рівні сигналів до номінальних і передає пакет сусідові нижче за течією. Переданий у такий спосіб пакет може містити дані або бути маркером. Маркер – 3 байтове повідомлення, що є ознакою вільної ЛКМ. Коли робочій станції треба передати пакет, її адаптер чекає надходження маркера, а потім перетворює його в пакет і передає результат далі в мережу. Пакет іде по мережі поки не знайде свого адресата, що встановлює ознаку підтвердження приймання пакета і відправляє його знову в мережу. Пакет продовжить рух у мережі, поки не повернеться у вузол,

із якого був пущений. Після здійснення безпомилкового приймання, цей вузол звільняє ЛКМ і випускає новий маркер. Для швидкодіючих мереж (більш 16 Мбіт/с) робоча станція передає маркер відразу після передачі свого пакета.

У динамічних топологіях з'єднання вузлів забезпечується електронними ключами, варіюючи установки яких можна змінювати топологію мережі. На відміну від статичних топологій, де вузлами є робочі станції, у вузлах динамічних топологій розташовуються елементи, які забезпечують комутацію, а робочі станції підключаються до входів і виходів мережі. Звичайно ключі в динамічних мережах групуються в так названі ступіні комутації. Наявність більш однієї ступіні комутації дозволяє забезпечити множинність шляхів між будь-якими парами робочих станцій.

Найбільш простий і дешевий вид динамічних мереж – мережа із шинною топологією (рисунок 6.5). При одношинній топології (рисунок 6.5 а) у кожен даний момент обмін повідомленнями може вести тільки одна пара станцій. Таку топологію часто використовують для об'єднання невеликої кількості робочих станцій у групу (кластер), а далі кластери поєднують у мережу на базі інших топологій. Багатошинна топологія (рисунок 6.5 б) припускає наявність  $n$  незалежних шин, до кожної з яких приєднується кожна робоча станція. Пропускна здатність зростає пропорційно числу шин, але керування такою мережею ускладнюється.

Існує безліч більш складних динамічних топологій мереж, що не є метою розгляду даного посібника.

Усі вище розглянуті топології припускали наявність файлу-сервера, на який покладаються функції керування обміну даними. Це мережі з централізованим керуванням.

Перевагою централізованих мереж є висока захищеність мережних ресурсів від несанкціонованого доступу, зручність адміністрування мережі, можливість створення мереж із великим числом вузлів.

Основні недоліки – уразливість системи при порушенні працездатності файлу-сервера, високі вимоги до ресурсів сервера і відповідно його велика вартість.

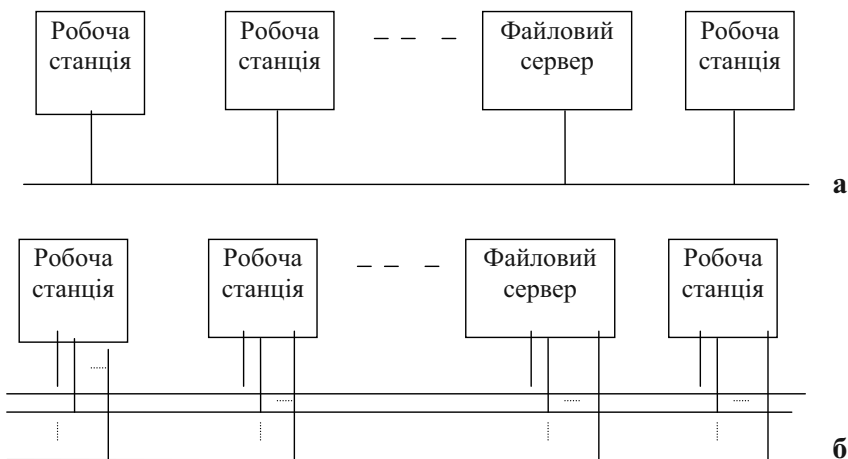


Рисунок 6.5 – Шинна топологія мережі: а – одношинна, б – багатошинна

Мережі з децентралізованим керуванням називаються одноранговими. **Однорангові мережі** – це мережі з рівноправних комп'ютерів, кожний з яких є і робочою станцією й сервером. У цьому випадку дисковий простір і файли всіх комп'ютерів стають загальними, якщо не поставлений спеціальний захист. При цьому комп'ютер може працювати повільніше через поділ ресурсів з іншими ПК. Але проте, однорангові обчислювальні мережі надійні, значно дешевші й вигідні для малих колективів (до 25 ПК). Однак такі мережі слабкіше з погляду захисту інформації й адміністрування.

### 6.1.3 Програмні засоби ЛКМ

Крім апаратних засобів ЛКМ необхідна мережна операційна система.

Для однорангових ЛКМ після включення комп'ютера завантажується автономна операційна система, а потім мережне програмне забезпечення у вигляді однієї або декількох резидентних програм.

Коли хтось з користувачів ЛКМ працює з прикладною програмою, що читає або записує файл на диску іншого ПК, мережна ОС активізується й бере керування цим комп'ютером на себе. Після задоволення запиту користувача вона повертає керування прикладній програмі. У зв'язі

зку з більш простою організацією мережних ОС для однорангових мереж набагато більше, ніж ОС для мереж на базі файлових серверів.

Прикладами ОС однорангових ЛКМ є Windows for Workgroups Microsoft Corporation, LANtastic фірми Artisoft, WEB фірми WebCorp, NetWare Lite and Personal NetWare фірми Novell.

Для ЛКМ на базі файлів-серверів мережна операційна система встановлюється на персональному комп'ютері, що буде виконувати функції сервера, а на робочих ПК інсталюється додаткове програмне забезпечення, що дозволяє їм обмінюватися даними з файловим сервером.

Існує безліч мережних операційних систем, що розрізняються своїми можливостями й умовами експлуатації.

Основними характеристиками мережних ОС є: залежність продуктивності від кількості підключених комп'ютерів, надійність роботи, рівень сервісу (обсяг і якість наданих послуг, керування функціонуванням мережі, можливість розробки прикладних програм у мережі і т.п.), захист інформації від несанкціонованого доступу, споживання ресурсів мережними засобами (необхідний обсяг оперативної й дискової пам'яті, споживана частка продуктивності і т.д.), типи підтримуваних топологій, наявність виходу в Internet і т.д.

Найбільш розповсюдженими мережними операційними системами є NetWare фірми Novell, Windows NT фірми Microsoft, різні ОС сімейства UNIX (FreeBSD, версії Linux), Vines фірми Banyan.

Наприклад, ОС NetWare здатна підтримувати робочі станції керовані різними автономними ОС (DOS, DOS і Windows, OS/2, UNIX і ін.); працювати з більшою, ніж будь-яка інша ОС, кількістю різних типів адаптерів; розростатися до величезних розмірів.

Windows 2000 Professional не є спеціалізованою мережною ОС, але вона базується на ядрі NT, цілком забезпечує роботу з виділеним сервером і дозволяє працювати в складних багаторівневих мережах.

У найпростішому випадку мережна ОС надає поділюваний накопичувач на жорсткому магнітному диску сервера як додатковий диск (наприклад, F) на кожній з робочих станцій мережі. Аналогічно мережна ОС дозволяє будь-якій робочій станції працювати на поділюваному мережному принтері так само, як якби він був підключений безпосередньо до цієї робочої станції.

Таким чином, більшість робочих програм на робочій станції нічого не будуть знати про наявність ЛКМ, навіть якщо вони будуть вико-

ристовувати файли на поділюваному диску або друкувати на мережному принтері.

У великих мережах можуть використовуватися декілька мережних ОС. Наприклад, часто спільно використовують ОС NetWare і Windows NT Server. Перша – для роботи з файлами і друку, друга – для обміну даними й роботи серверів додатків на різних платформах.

Компоненти мережної операційної системи на кожній робочій станції і файловому сервері взаємодіють один з одним за допомогою протоколів. Комп'ютерні протоколи – ієрархічна система правил взаємодії.

Міжнародною організацією по стандартизації визначено 7 рівнів протоколів для комунікаційних функцій:

- ☒ прикладний – прикладні програми користувача;
- ☒ представницький – ОС фіксує місце розташування переданих даних і забезпечує зв'язок із наступним рівнем;
- ☒ сеансів зв'язку – перевірка прав користувача й передача даних на транспортний рівень;
- ☒ транспортний – перетворення даних у пакети (форму для передачі даних по мережі);
- ☒ мережний – визначає маршрут руху даних по мережі;
- ☒ з'єднання – моделювання сигналів каналу зв'язку (операція кодування/декодування мережного адаптера);
- ☒ фізичний – реальна передача даних у виді імпульсів.

Протоколи верхніх рівнів реалізуються програмно, протокол з'єднання виконується мережною картою або модемом, фізичний рівень реалізується за межами комп'ютера в лініях зв'язку.

Найбільш розповсюджені протоколи – NetBIOS фірми IBM, IPX фірми Novell.

На найнижчому рівні комп'ютери в мережі обмінюються інформацією у вигляді пакетів повідомлень. Ці пакети складають фундамент, на якому базується робота ЛКМ. У різних системах комп'ютерних мереж пакети визначаються по-різному, але наступні елементи є загальними для усіх (рисунок 6.6):

- унікальна адреса відправника;
- унікальна адреса одержувача;
- ознака, що визначає вміст пакета;
- дані або повідомлення;
- контрольна сума (CRC) для виявлення помилок передачі.

Адреса відправника	Адреса одержувача	Тип пакету	Дані/повідомлення	CRC
--------------------	-------------------	------------	-------------------	-----

Рисунок 6.6 – Базова схема пакета повідомлень

Протоколи різних рівнів нашаровуються один на іншій. Кожен протокол створює свою оболонку-конверт, починаючи із середнього і кінчаючи нижнім рівнем. Можна позначити кожен оболонку як заголовок і закінчення (приклад для мережі з трьома рівнями протоколів на рисунку 6.7). При прийомі повідомлень на кожному рівні зовнішня оболонка видаляється програмним забезпеченням, після чого проводиться передача повідомлення на більш високий рівень.

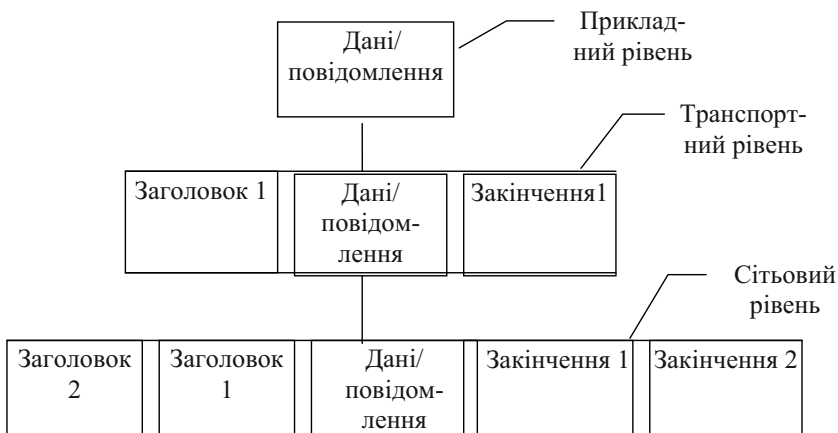


Рисунок 6.7 – Кілька рівнів пакетів повідомлень

Існує два типи міжкомп'ютерного обміну даними

- *датаграми i*
- *сеанси.*

**Датаграми** – це повідомлення, що не вимагають підтвердження про прийом від приймаючої сторони. Якщо таке підтвердження необхідне, то адресат повинен сам послати спеціальне повідомлення. Кожна датаграма є самостійним повідомленням, і при наявності декількох датаграм у ЛКМ порядок їхньої доставки не гарантується.



У деяких випадках максимальний розмір датаграм набагато менше, ніж довжина повідомлень у сеансах. Зате в більшості ЛКМ швидкостей передачі датаграм набагато вище, ніж повідомлень у сеансах.

На противагу датаграмам, у *сеансі* передбачається створення логічного зв'язку для обміну повідомленнями між робочими станціями і гарантується одержання повідомлень. У той час як датаграми можуть передаватися в довільні моменти часу, у сеансі для передачі повідомлень спочатку необхідно виконати деяку підготовчу роботу: сеанс повинен бути спочатку встановлений, після цього відбувається обмін повідомленнями і, нарешті, після закінчення обміну даними сеанс повинен бути закритий.

Фірма Novell у мережній ОС NetWare застосовує протокол IPX для обміну датаграмами і протокол SPX для обміну в сеансах.

Якщо користувач хоче використовувати ЛКМ із найбільшою ефективністю, то йому знадобляться мережні прикладні програми, що реалізують переваги, надані ЛКМ. Застосування багатокористувальницьких версій прикладних програм дозволить різко збільшити продуктивність.

#### 6.1.4 Захист даних у мережі

При роботі ЛКМ файли всіх користувачів знаходяться в одному великому сховищі. Якщо не розпочати спеціальних мір безпеки, то хто завгодно зможе переглядати і модифікувати будь-які файли, що містять дані й документи будь-якого іншого користувача, включаючи президента компанії. Крім того, навіть мимовільна помилка одного з користувачів може привести до втрати інформації у мережі (команда del замість dir). Збій у мережі живлення або несправності дискового накопичувача можуть мати аналогічні наслідки. Тому питанням захисту інформації у мережі віддається особливе значення.

Розроблено цілий ряд методів рішення даної проблеми. Це:

- 1) використання паролів. Кожному користувачеві в ЛКМ привласнюється пароль, що періодично міняється й тримається в секреті;
- 2) обмеження доступу до визначених директорій або визначеним серверам. Наприклад, в ОС NetWare користувач може відкривати й читати директорії, але модифікувати їх він не може.

- 3) резервне копіювання (на дискети, на стример, оптичні диски) випадковим, серйозним (за допомогою програм типу BACKUP) або професійним (схема “син-батько-дід”) способами.
- 4) апаратурна надмірність у сполученні з резервним копіюванням. Краще мати 2 середніх розмірів сервера, чим один великий. На другому сервері повинні знаходитися резервні копії даних. Виробниками також пропонуються сервера з дисковими масивами – системами накопичувачів, у яких інформація дзеркально дубльована на різних дисководах. У випадку якщо один накопичувач вийде з ладу, його негайно замінить інший без збою системи.
- 5) використання джерел безперебійного живлення (UPS) або більш дешеву систему чергового живлення (SPS) для захисту файлів-серверів від збою живлення. Після провалля напруги в мережі батареї UPS забезпечать працездатність сервера в наступних 10 хвилин, що досить для автоматичного завершення роботи й збереження робочих файлів.

## **6.2 Глобальні мережі. INTERNET**

### **6.2.1 Загальні принципи побудови глобальних мереж**

Інтернаціоналізація наукових, ділових, культурних зв'язків і просото особистих контактів у сучасному світі багато в чому можлива без фізичного переміщення людини завдяки розвитку різних засобів комунікації.

Усе більші можливості в цьому плані надають глобальні мережі.

Глобальні мережі поєднують окремі локальні мережі. Уся робота в глобальних мережах будується на базі трьох основних компонентів:

- 1) технічній основі;
- 2) використанні протоколів;
- 3) використанні мережних додатків.

## Технічна основа Інтернет

Технічною основою, що з'єднує всю розмаїтість комп'ютерів у єдину мережу, є лінії зв'язку, від яких залежить швидкість і якість передачі інформації у мережі, вартість підключення до мережі (через вартість установки (оренди) і експлуатації ліній зв'язку).

Робота по з'єднанню двох ЛКМ буде простіше, якщо вони мають однакові топології й однакові мережні операційні системи, задовольняють широко розповсюдженим стандартам і протоколам. Якщо це не так, то прийдеться шукати системи, що спеціалізуються на з'єднанні ЛКМ із потрібними топологіями або такими мережними ОС. Для з'єднання різних ЛКМ можуть знадобитися мости, шлюзи, звичайні й гібридні маршрутизатори.

Мости призначені для з'єднання мережних сегментів (окремих ЛКМ), що мають різні фізичні середовища, наприклад, для з'єднання сегмента з оптоволоконним кабелем і сегмента з коаксіальним кабелем. Мости також можуть використовуватися для зв'язку сегментів, що мають різні протоколи низького рівня, тобто можна з'єднувати окремі ЛМ, що працюють по принципах дозволу колізій і передачі маркера.

Мости аналізують, фільтрують, направляють повідомлення, прагнучи понизити трафік (потік даних) сегмента, до якого вони підключені. Цим пояснюється їхня важлива роль в об'єднанні мереж. При виявленні перевантаженості трафіку фізичного сегмента мережі, можна розділити його на 2 фізичних сегменти за допомогою моста, що обмежить трафік кожного з них, не завантажуючи сегмент даними адресованими іншому.

Мости часто виявляються повільними пристроями, тому що їм приходится витратити багато часу на визначення й аналіз адрес повідомлень, на перевірку пакетів і напрямку їх по заданих адресах. Але вони мають комунікаційні якості, що роблять їх дуже корисними в комп'ютерних мережах із змішаними протоколами.

Маршрутизатори не мають таку здатність до аналізу повідомлень, як мости, але зате можуть приймати рішення про вибір оптимального шляху для даних між двома мережними сегментами. Для роботи маршрутизаторів потрібен той самий протокол у всіх сегментах, із якими він зв'язаний.

Гібридні маршрутизатори – це гібрид моста і звичайного маршрутизатора. Вони здатні приймати рішення про те, чи можливо маршрутизувати пакет із даним протоколом. Після цього вони роблять маршрути-

зацію тих повідомлень, для яких це можливо, а для інших служать мостом.

Гібридні маршрутизатори – це складні, дорогі і важко установлені прилади, але для складних неоднорідних мереж вони представляють найкраще рішення.

Шлюзи – найбільш розвитий метод приєднання мережних сегментів і комп'ютерних мереж до центральних ЕОМ. Необхідність у мережних шлюзах виникає при об'єднанні систем, що мають різну архітектуру. У цих випадках потрібно переводити увесь потік даних, що проходять між двома системами.

Прикладами глобальних мереж є: Internet, європейська мережа EUNET, частиною якої є мережа RelCom, інші глобальні мережі – CompuServe, BITNET, UUNET і т.д.

В подальшому викладенні основні особливості глобальних мереж будуть розглянуті на прикладі мережі Internet.

Мережа Internet є найбільшою всесвітньою комп'ютерною мережею. У 1969р. Управління перспективних досліджень ARPA (Advanced Research Project Agency), один з підрозділів Міністерства оборони США, почало роботу над проектом, що повинен був, з одного боку, привести до створення каналів зв'язку, що практично не піддаються руйнуванню, а, з іншого боку – полегшити співробітництво між розкиданими по всіх штатах дослідницькими організаціями оборонної промисловості. Так виникла мережа ARPANet.

Мережа ARPANET поступово росла від жменьки комп'ютерів у 1971 р. до 1000 у 1984 р. Паралельно з'явилися й інші комп'ютерні мережі. Стало ясно, що при спілкуванні користувачів цих мереж був би величезний виграш.

У сімдесятих роках за підтримкою ARPA були розроблені протоколи (правила) пересилання даних між різними комп'ютерними мережами, що уможливило розробку всесвітньої Мережі. І от у 1986 р. Національний науковий фонд США з метою з'єднання в мережу великого числа науково – дослідницьких установ і розвитку міжнародної кооперації заснував проект NSFNET, результатом якого і з'явилося створення Internet. Було створено 5 дуже дорогих суперкомп'ютерних центрів, доступних для використання в будь-яких наукових установах, потім кількість центрів збільшилася до 13. У кожній частині країни зацікавлені установи повинні були з'єднатися зі своїми найближчими сусідами. Ланцюжки, що вийшли, з'єднувалися до суперкомп'ютера в одній зі своїх точок. Суперкомп'ютерні центри через такі ланцюжки були з'єднані ра-

зом. У такій топології будь-який комп'ютер міг зв'язатися з будь-яким іншим, передаючи повідомлення через сусідів. У той же час розробили спосіб використання мережі для проведення електронних конференцій.

Світ одержав мережу з цілком добровільною участю, керовану чимось на зразок ради старійшин.

Вища влада в Internet визнається за суспільством із добровільним членством – ISOC (Internet Society). Воно призначає «раду старійшин», що відповідає за технічну політику, підтримку й керування Internet, і являє собою групу запрошених добровольців, називану IAB (Рада по архітектурі Internet). IAB відповідає за стандарти мережі, стежить за правильним їх дотриманням.

Наприклад, кожен комп'ютер у мережі повинен мати унікальний IP адрес (пояснення далі). Сам IAB не присвоює адрес, але саме він розробляє правила їхнього присвоєння.

Інша добровільна організація IETF (Оперативний інженерний загін Internet) збирається регулярно, щоб обговорити поточні експлуатаційні і назріваючі технічні проблеми. Робочі групи добровольців працюють у різних напрямках – розробка нових стандартів або доробка існуючих, стратегічні дослідження, вироблення стратегії при виникненні проблем, випуск документації. Результат їхньої роботи – доповідь.

Некомерційна Інтернет-організація ICANN координує політику реєстрації доменних імен, але особливо не втручається в те, як розпоряджаються своїми доменами окремі реєстратори і країни.

За Internet ніхто централізовано не платить. Представники мереж збираються разом і вирішують, як їм з'єднатися й утримувати ці взаємозв'язки. Користувач платить за підключення до деякої мережі, яка у свою чергу платить власникові більш високого масштабу.

Приєднані до мережі комп'ютери одержали назву хостів мережі. Такий хост може бути частиною вашої власної ЛКМ або належати комерційному постачальнику послуг доступу до мережі (**провайдеру**). З'єднання ЛКМ із Internet здійснюється за допомогою мостів, маршрутизаторів або шлюзів через виділені волоконооптичні, супутникові або телефонні лінії. Швидкість передачі даних залежить від застосовуваних ліній зв'язку і коливається в діапазоні від 28,8 Кбіт/с (модем на звичайних телефонних лініях – ледве менш 2-х заповнених екранів) до 45 Мбіт/с (більш 2800 повних екранів) і більш. Швидкість передачі по оптоволоконних лініях опорних магістралей Інтернету досягає одного гигабіта в секунду.

## Використання протоколів

Кожне повідомлення, що посилається в Internet, проходить крізь протоколи, щонайменше, 3-х рівнів. Це наступні протоколи: міжмережний (IP), на якому відслідковується доставка повідомлень від одного місця до іншого; транспортний, на якому відслідковується цілісність переданих повідомлень (TCP); і прикладний рівень, на якому комп'ютерні формати повідомлень перетворюються у повідомлення, які може читати людина, і навпаки. Задача перетворення повідомлень у формати протоколів і назад – важка і нудотна – виконується мережними хостами, що рятує користувача від роботи з адресації у мережі.

Співвідношення протоколів еталонної моделі (OSI), що рекомендуються Міжнародною організацією по стандартизації, і протоколів, що працюють у мережній архітектурі TCP/IP, показано в таблиці 6.1 [7].

Припустимо, потрібно передати інформацію з одного комп'ютера, підключеного до Інтернету, на інший комп'ютер. Протокол TCP одержує інформацію з прикладної програми, розбиває її на пакети і нумерує всі пакети, щоб при одержанні можна було правильно зібрати інформацію.

Передані в мережі пакети досить малі – близько 1500 байтів і менш. Далі за допомогою протоколів IP усі пакети передаються одержувачеві, де за допомогою протоколу TCP перевіряється, чи всі частини отримані. Тому що окремі частини можуть подорожувати по Інтернету самими різними шляхами, то порядок приходу частин може бути порушений. Після одержання всіх частин TCP розташовує їх у потрібному порядку і збирає в єдине ціле.

Таблиця 6.1. – Рівні еталонної моделі й протоколи TCP/IP

Модель OSI	TCP/IP
Рівень додатків	Рівень додатків
Рівень представлення	
Рівень мережного інтерфейсу користувача	
Транспортний рівень	Транспортний рівень
Мережний рівень	Internet
Канальний рівень	Мережний інтерфейс
Фізичний рівень	Фізичний рівень

При передачі датаграм використовується протокол датаграм користувача (UDP). Цей протокол особливо зручний при передачі малих повідомлень, що уміщаються в одному пакеті. Він також використовується

ся при передачі аудіо й відео інформації, де втрата одного пакета не надто позначається на якості даних. При необхідності використовують відразу декілька протоколів, для розв'язання різних задач в одному сеансі.

Протоколи TCP/IP широко використовуються не тільки в Internet, але й у локальних мережах.

Щоб міжмережні протоколи IP могли виконувати свою роботу, необхідна система ідентифікації вузлів або комп'ютерів, між якими відбувається обмін повідомленнями. Основними форматами адрес комп'ютерів у Internet є: IP-адреса, DNS-адреса, адреса E-mail, адреса телеконференції USENET.

IP-адрес. При розробці Internet однією з найбільш важливих задач було створення мережі, яку практично неможливо було б вивести з ладу. Тому передача даних між двома довільними точками не повинна була проходити через який-небудь центр. З цієї причини була створена гнучка система, завдяки якій Internet передає дані адресатам самостійно.

Основний формат Internet – адреси, як і вся інформація, оброблювана комп'ютером, є числовим і виражається у двійковій системі. Інформація, виражена у двійковій системі, – це число, що складається винятково з цифр 0 і 1. Internet-адреса має 32 біта. Двійкову адресу розділяють на чотири блоки по 8 біт, а потім число, що міститься в блоці, записують у десятковій системі. Отримані числа відокремлюють крапками. Кожен блок може містити число від 0 до 255. Праве крайнє число – адреса комп'ютера, інші – адреси мереж, у які входить цей комп'ютер.

Вкладеність адрес із правого краю на лівий.

Приклад синтаксису IP-адреси: 123.45.67.89 – комп'ютер 89 входить у мережу 67, що у свою чергу приєднана до мережі 45, що є частиною мережі 123.

IP-адрес, навіть записаний у десятковій системі, не так-те легко запам'ятати. Тому разом з ним була розроблена Система імен доменів, що включає службу каталогізації й набір угод по найменуванню комп'ютерів мережі (хостів).

*Домен* – це поійменована група хостів мережі. Повні імена доменів у такій системі складаються із серії простих імен, які розділяються крапками. Ім'я комп'ютера стоїть крайнім ліворуч, правим крайнім у DNS-адресі стоїть домен найвищого (першого) рівня – географічний чи організаційний.

Принципи DNS-адресації можна пояснити на прикладах:

**donduet.edu.ua** – крайнє лїве поле (donduet) є ім'ям хосту, що призначене комп'ютеру його адміністратором. Після імені хосту йде ім'я домену, членом якого є цей хост. Комп'ютер з ім'ям donduet є членом домену edu (освітні установи), що знаходиться в Україні (ua).

**gopher.th-darmstadt.de.** Gopher – це ім'я комп'ютера в мережі th-damstadt домену de, що служить для позначення Німеччини. Зазначена адреса зчитується праворуч лїворуч. Насамперед, Internet установлює зв'язок з вищестоящим маршрутизатором, що керує всіма адресами домену de. Якщо адреса необхідної зони знайдена, то може бути встановлений зв'язок із відповідною мережею (у даному прикладі з ЛВМ Технічного інституту м. Дармштадт), де знаходиться каталог усіх локально приєднаних комп'ютерів. Таким чином, дані можуть бути передані потрібному комп'ютеру.

Приведені приклади ілюструють географічний принцип побудови доменних імен. Кожній країні, мережі якої підключилися до Internet, виділено і закріплено стандартом відповідне ім'я домену (таблиця 6.2).

Таблиця 6.2. – Деякі домени верхнього географічного рівня

Ім'я домену	Країна	Ім'я домену	Країна
au	Австралія	is	Ісландія
at	Австрія	it	Італія
ca	Канада	jp	Японія
cl	Чилі	kr	Корея
dk	Данія	nz	Нова Зеландія
ec	Еквадор	ru	Росія
es	Іспанія	se	Швеція
fi	Фінляндія	su	Колишній СРСР
fr	Франція	tw	Тайвань
de	Німеччина	uk	Англія/Ірландія
us	Сполучені Штати Америки	ua	Україна

Приклади доменних імен, сформованих по організаційній ознаці:

**accent6.com** – комп'ютер з ім'ям accent6 є членом домену використуваного комерційними організаціями (com).

<http://www.vu.org> – віртуальний університет, що відноситься до групи некомерційних організацій;



<http://www.fbi.gov> – штаб-квартира ФБР (група урядових закладів США);

Міжнародний стандарт закріплює домени, побудовані по організаційному принципу (таблиця 6.3).

Таблиця 6.3. – Домени верхнього “організаційного” рівня

Ім'я домену	Тип
com	Комерційні організації
edu	Освітні установи
gov	Урядові заклади США
int	Міжнародні організації
mil	Організації збройних сил США
net	Системи опорних мереж і інформаційні центри
ogr	Некомерційні організації

Усі DNS – імена і IP – адреси зібрані в розподілену базу даних, розміщену по адміністративних вузлах Інтернету. DNS – система автоматично установлює відповідність символічних і цифрових адрес.

При роботі в Інтернеті найчастіше використовуються не просто доменні адреси, а **універсальні покажчики ресурсів**, називані **URL - Universal Resource Locator**.

URL – це адреса будь-якого ресурсу в Інтернеті разом з указівкою того, за допомогою якого протоколу слід до нього звертатися. У покажчику крім власне адреси є відомості про те, яким протоколом варто звертатися до даного ресурсу, яку програму для цього варто запустити на сервері і до якого конкретного файлу варто звернутися на сервері.

Прикладом покажчика може бути **<http://www.microsoft.com/ie>**. Назва протоколу **http** на початку вказує, що далі йде адреса Web-сторінки, а назва **ie** наприкінці покажчика описує каталог з ім'ям **ie** на сервері **www.microsoft.com**.

Покажчик **<ftp://www.mycompany.ru/business/index.html>** описує, що до файлу **index.html**, розташованому в каталозі **business** на сервері **www.mycompany.ru** варто звернутися по протоколу передачі файлів FTP.

Зверніть увагу на те, що розширення файлу складається з чотирьох букв, що допускається правилами утворення імен і розширень файлів у мережних операційних системах.

Після доменної адреси може знаходитися номер порту, тобто номер програми, яку треба завантажити на сервері. Крім того, замість доменних імен у URL можна використовувати цифрові адреси. Таким, образом, користувач може побачити досить складну адресу, типу:

**ftp://123.45.6.78:9535/workarea/common/main/text.doc**

Такі складні покажчики на практиці користувачу не придеться конструювати. У переважній більшості випадків можна обмежитися доменною адресою й іменами каталогів і файлів, що розташовуються за адресою і розділяються косими рисами. Порти в переважній більшості випадків описувати не треба, тому що вони визначаються за замовчуванням. Типи протоколів також самостійно визначаються й підставляються на адресу сучасними програмами. Тому, замість `http://www.name.ru/users` прийнято говорити `www.name.ru/users`. Часто URL називають адресою ресурсу. Якщо далі буде говоритися про адресу, то мова йде саме про URL, а не про доменне ім'я або цифрову адресу.

Інші формати адрес будуть розглянуті далі при описі відповідних видів послуг мережі.

### **Використання мережних додатків**

Додатки, використовувані в Internet, відрізняються по своїй загальній структурі від автономних (немережевих) додатків. Мережні додатки побудовані відповідно до моделі “клієнт – сервер”. Інструментальні засоби, засновані на цій архітектурі, розподіляють роботу одного додатка між двома програмами, так називаною програмою-клієнтом і програмою-сервером, що вступають у діалог один з одним по мережі.

Фактично всі додатки, застосовувані для доступу до вилучених ресурсів Internet, використовують модель “клієнт – сервер”. Це дозволяє забезпечити доступ до всілякої інформації – від сільськогосподарської до астрофізичної будь-якому користувачу, що має доступ до Internet.

Програма – клієнт, що запускається на комп'ютері користувача, уточнює, що саме йому потрібно, і з'єднується через мережу з програмою – сервером, що керує необхідною інформацією. Програма – сервер знаходить необхідний ресурс або інформацію і посилає відповідь назад клієнту (через відповідні протоколи).

Програма типу “клієнт” завжди виконується асинхронно. Користувач запускає її в міру необхідності, а програма – сервер звичайно, працює безупинно. Коли активних клієнтів нема, вона продовжує працювати, очікуючи надходження нових запитів.

Таким чином, ми коротко познайомилися з трьома “китами”, на яких побудована мережа Internet.

## 6.2.2 Підключення до INTERNET

Способи підключення до Internet постійно міняються з розвитком мережного технічного й програмного забезпечення. Основним параметром для порівняння різних способів підключення до Internet і зіставлення якості послуг, наданих провайдерами, є швидкість передачі повідомлень між комп'ютерами в мережі.

Для оцінки пропускної здатності мережі можна орієнтуватися на наступні дані [8]. Припустиме чекання появи на екрані комп'ютера текстової сторінки з невибагливою графікою 4 – 5 секунд. Обсяг такої сторінки з допоміжною мережною інформацією не перевищує 30 Кбайт (240 Кбіт). Для надходження такого обсягу інформації на комп'ютер за 5 секунд потрібна швидкість потоку цифрових сигналів порядку 50 Кбит/с. Для перегляду відеофільму в реальному режимі часу потрібна швидкість у 20 разів більша.

В даний час використовуються чотири різних варіанти підключення до Internet:

- постійне підключення, щодня, двадцять чотири години в добу;
- робота винятково за допомогою електронної пошти;
- з'єднання, що комутирується, за допомогою емуляції термінала;
- IP-з'єднання, що комутирується.

Коротко розглянемо ці варіанти.

Постійне підключення дозволяє користувачу працювати з Інтернетом у будь-який час і часто використовується для підключення локальної мережі до Інтернету, після чого всі користувачі локальної мережі можуть одночасно працювати в Інтернеті.

Постійне підключення здійснюється за допомогою виділеної лінії зв'язку, що забезпечує високу швидкість передачі інформації. Тому що цей тип підключення досить дорогий, то його не використовують приватні особи і невеликі організації. В основному, він використовується середніми й великими фірмами. Крім того, при таких підключенні легко можна організувати доступ будь-якого користувача Інтернету до інформації, розташованої на комп'ютері фірми. Таким чином, і створюються численні вузли в Інтернеті, що може відвідувати користувач, який підключився до Інтернету.

Робота винятково за допомогою електронної пошти забезпечує користувачу можливість тільки посилати й приймати електронні листи за допомогою Інтернет. Цей спосіб називається підключенням по протоколу UUCP. Достоїнством даного способу є відносна дешевина і трохи більша надійність на поганих каналах зв'язку.

Як канали зв'язки використовуються звичайні телефонні лінії. Хоча користувач може одержати практично будь-яку інформацію за допомогою електронної пошти, але робота при безпосереднім з'єднанні з мережею значно зручніше.

Одним із найпоширеніших і дешевих способів Інтернет-з'єднань є з'єднання за допомогою телефонних ліній через відповідний модем. Для роботи з Internet по телефону існують 2 способи: термінальний доступ по каналу, що комутирується, і доступ, що комутирується, до інтернетовського протоколу.

З'єднання, що комутирується, за допомогою емуляції термінала. При термінальному доступі користувач установлює діалоговий сеанс зв'язку з комп'ютером-хостом Internet за допомогою свого комп'ютера, модему і комунікаційної програми. Установивши термінальне з'єднання з інтернетовським комп'ютером, у відповідь на його запит користувач повинен вказати реєстраційне ім'я для входу в систему й пароль. Це дозволить вузлу упевнитися у тім, чи має користувач право працювати з системою й установити з'єднання.

Після того як модеми встановили з'єднання, спеціальне програмне забезпечення перетворює комп'ютер користувача у вилучений термінал комп'ютера постачальника. Звідси і йде назва «емуляція термінала». Тепер комп'ютер користувача не виконує власні обчислення, а просто, як і призначено терміналу, посилає усе, що набирає користувач на своїй клавіатурі, віддаленому комп'ютеру.

Віддалений комп'ютер розпізнає команди і виконує відповідні їм дії, результати яких передаються назад і виводяться на екран монітора користувача. Тобто практично користувач як би працює за комп'ютером постачальника послуг Інтернету, хоча сидить дома за своїм комп'ютером. Для такої роботи користувачу треба знати, яка система встановлена в постачальника, і як з нею працювати. Це не дуже зручно, тому останнім часом такий доступ практично не використовується, хоча кілька років назад значна більшість користувачів Інтернету в усьому світі користалися саме таким способом з'єднання. Сьогодні ж цей спосіб використовують в основному професіонали, щоб домогтися деяких нестандартних результатів.

IP-з'єднання, що комутирується – основний спосіб підключення до Інтернету. За допомогою цього з'єднання модем користувача через звичайну телефонну лінію зв'язується з модемом, встановленим

на комп'ютері постачальника мережних послуг. Після установки зв'язку комп'ютер користувача стає повноправним користувачем Інтернету, якому доступні всі послуги без винятку. При цьому комп'ютер користувача сам забезпечує роботу інтернетівських протоколів обміну даними. Обмін інформацією між комп'ютерами при такім з'єднанні йде за допомогою протоколів TCP/IP.

Протоколи TCP/IP споконвічно були розроблені для комп'ютерів, що знаходяться в постійному з'єднанні один з одним. Щоб ці протоколи працювали при з'єднанні, що комутирується, використовуються додаткові протоколи: SLIP або PPP.

SLIP (Serial Line Internet Protocol) – Інтернет-протокол для послідовного каналу.

PPP (Point-to-Point Protocol) – протокол крапка – крапка, також називаний протоколом взаємодії між вузлами. Даний протокол з'явився пізніше ніж SLIP, і в ньому враховані недоліки протоколу SLIP. Наприклад, він містить більше можливостей для корекції помилок.

Протокол PPP поступово витісняє SLIP.

Якщо постачальник послуг пропонує на вибір підключення по SLIP чи по PPP, то краще вибрати PPP як більш досконалий.

Для домашніх комп'ютерів, «домашніх» мереж, невеликих офісів і фірм, коли потрібна висока швидкість потоків даних, можуть використовуватися інші способи підключення.

Підключення по виділеній телефонній лінії – коли спеціально підготовлена і відтестована лінія використовується тільки для виходу в Інтернет. Підтримувана швидкість – від 64 до 256 Кбіт/с.

Технологія підключення ADSL – забезпечує високошвидкісну передачу даних по звичайних телефонних лініях зв'язку (швидкість до користувача 1,5 – 8 Мбіт/с, від користувача – від 640 Кбіт/с до 1,5 Мбіт/с). Для такого з'єднання необхідні спеціальний ADSL-модем і спеціальне устаткування на телефонній станції. У лініях зв'язку виділяють три діапазони по частоті – для голосових повідомлень (телефон), для комп'ютерних даних від комп'ютера користувача і до нього. У результаті можна говорити по телефону й обмінюватися даними по мережі одночасно.

Передача даних по мережах кабельного телебачення. При наявності мережі кабельного телебачення і відповідного модему можна підк-

лючитися до Інтернет і одержувати дані зі швидкістю до 10 Мбіт/с, а відправляти запити зі швидкістю до 2 Мбіт/с. Цей вид з'єднання активно розвивається, швидкість передачі підвищується, ціни за 1Мбайт інформації падають.

Безпроводний доступ по радіоканалу на надвисоких частотах. Радіоканалом зв'язують мережі, що знаходяться на великій відстані, але в межах видимості антен. Зв'язок з комп'ютером здійснюється через радіомодем. Швидкість передачі – від 64 Кбіт/с до 2 Мбіт/с.

Технологія доступу в Інтернет через мобільні телефони. Створено протоколи безпроводних систем зв'язку (WAP, GPRS), що дозволяють безпосередньо зв'язуватися з Інтернет через мобільні телефони. Безумовно, що одержувати великі обсяги даних на маленький екран незручно, але використовувати для зв'язку, одержання довідкової інформації і навіть виконувати певні економічні операції (оплата товарів, послуг...) цілком зручно.

Вибір того чи іншого способу підключення залежить від потреб і фінансових можливостей користувача.

### **6.2.3 Інформаційні послуги Internet**

Інформаційні послуги (або сервіси) мережі Internet- це прикладні програми, розроблені з метою одержання доступу до інформації визначеного типу або з метою обміну даними. До основних послуг Internet відносяться – E-MAIL, USENET NEWS, WWW, FTP, ICQ.

#### **Електронна пошта**

Електронна пошта є найбільш популярним доповненням до локальних обчислювальних мереж і широко використовується в глобальних мережах.

Електронна пошта, або E-mail, є однієї з основних прикладних програм у Internet. У першу чергу, вона служить для передачі текстових повідомлень у межах глобального співтовариства Internet. У цій якості вона практично не відрізняється від звичайної пошти, хіба що функціонує значно швидше (можна сказати миттєво), а оплата за її використання здійснюється по місцевому тарифу, незалежно від того, у яким місці земної кулі знаходиться ваш кореспондент.

Як і завжди при передачі даних у Internet, користувач оплачує тільки передачу інформації до вузла свого провайдера. Витрати по подальшому пересиланню, незалежно від географічної далекості кореспондента, на користувача не покладаються.

Повідомлення, що надходять по електронній пошті, зберігаються в спеціальному поштовому комп'ютері в особистій "поштової скриньці" одержувача, відкіля він може їх у будь-який момент вивантажити й переглянути.

Електронна пошта – це багатозначний термін, використовуваний для визначення процесу передачі повідомлень між комп'ютерами.

Повідомлення можуть бути і простими записками, що пересилаються в сусідню кімнату, і мультимедійними відеопрезентаціями, переданими на інший континент.

З погляду користувача електронна пошта дуже схожа на звичайну пошту, однак вона не зводиться тільки до пересилання повідомлень по електронних каналах зв'язку. Електронна пошта – це зовсім нове комунікаційне середовище зі своїми особливостями, проблемами і навіть етикою.

З історії електронної пошти. Попередником електронної пошти вважається телекс. Це була друкарська машинка, з'єднана з телефоном. Бізнесмен із Каліфорнії диктував лист секретарю, що набирив текст для передачі по телексу. Через кілька митей повідомлення з'являлося на приймаючому телексі в Нью-Йорку. Телекс був дорогий, але надійний, простий у використанні і швидко передавав повідомлення по телефонних лініях.

Потім з'явився факс. Можна розглядати факс як пристрій передачі фотокопій. Робота з пристроєм дуже проста. Потрібно тільки набрати телефонний номер факсу одержувача і помістити у свій факс переданий документ. Однак, процеси, що відбуваються під час такої передачі, не такі вже і прості. Факс-машина розбиває текст документа на мільйони чорних і білих крапок, стискає інформацію і посилає її по телефонній лінії, де факс-машина, що приймає, відновлює вміст документа і друкує його.

Наступним нововведенням у сфері комунікацій стала електронна пошта. На початку 1960 років електронна пошта спочатку була названа CBMS – системою передачі повідомлень за допомогою комп'ютерів (Computer Bissed Messaging System). Фірма Western Union уперше зареєструвала торгову марку Electronic Mail (Електронна пошта) у 1974 ро-

ці. Ця ідея була реалізована вченими й програмістами, що працювали за терміналами, саме вони і розробили прості програми обміну повідомленнями. У багатьох випадках передача повідомлень була зручніша, ніж розмови по телефону або особисті контакти. Спочатку це були повідомлення, що негайно з'являлися на екрані терміналу, але згодом з'явилася можливість залишати повідомлення співробітникам, відсутнім за терміналом у момент передачі. Можна було написати й послати коротку записку одержувачу, поза залежністю чи зайнятий він у цей час іншими справами чи виїхав в інше місто.

Перші розроблювачі електронної пошти стали розвивати ідею обміну повідомленнями, намагаючись моделювати функції звичайної пошти. Користувач міг підключитися до терміналу, перевірити наявність пошти на своє ім'я, а також прочитати або послати пошту. В міру росту числа терміналів і повідомлень, що пересилаються, стало незручно користатися однією загальною електронною поштовою скринькою. Було розроблено програмне забезпечення, що дозволяє користувачу мати власну поштову скриньку. Далі з'явилися поштові системи із засобами забезпечення конфіденційності. Тепер підключитися до поштової системи можна було, тільки набравши своє ім'я й пароль.

Для організації електронної пошти створений ряд пакетів програм (поштові клієнти): Pegasus Mail, cc:Mail, Microsoft Mail, Da Vinci eMail, Higgins Mail, Eudora, The Bat, Outlook Express, Outlook 2000.

Одним із безкоштовно розповсюджуваних пакетів електронної пошти є Pegasus Mail для локальних мереж NetWare. Цей пакет легко встановлюється, надійно працює і простий у звертанні. Пакет можна конфігурувати по своєму смаку, наприклад, можна складати персональні списки розсилання, перенаправляти пошту іншим співробітникам, архівувати стару кореспонденцію.

Недоліком цього пакета є відсутність засобів для обміну кореспонденцією із зовнішніми мережами. Існують версії пакета під DOS і Windows. Установка пакета Pegasus Mail полягає в копіюванні декількох файлів у загальну директорію ЛКМ. Після цього кожної з користувачів ЛКМ може запустити пакет Pegasus Mail.

Фірма Microsoft також активно займається розробкою програм електронної пошти, включаючи їх складовою частиною в пакет Microsoft office. Це програми Microsoft Mail і Schedule + для Windows – 95 (office-95), Outlook Express, Outlook для більш старших версій.



Електронна пошта дозволяє обмінюватися не тільки діловими документами, але і неформальною інформацією, для передачі якої раніш був потрібний телефон. Електронну пошту можна використовувати для:

- швидкої й зручної посилки факсів;
- обміну файлами (даними, електронними таблицями, документами, створеними за допомогою текстових процесорів, графікою, відео кліпами, музикою, мовними повідомленнями, а також будь-якою іншою інформацією, що може зберігатися в пам'яті комп'ютера);
- спілкування зі своїми родичами, друзями, що живуть у різних (навіть самих віддалених) регіонах світу;
- пересилання листів і т.д.

#### Компоненти електронної пошти.

По суті, системи електронної пошти, що використовують файлові сервери, мають просту структуру. Центральна база даних містить імена й адреси всіх користувачів пошти. Інша база даних зберігає індивідуальні поштові повідомлення і зв'язані з ними файли.

Усі програмні пакети, призначені для реалізації електронної пошти, складаються з двох елементів: власне поштової служби і користувальницького інтерфейсу.

Компонента поштової служби одержує повідомлення, декодує адреси і відправляє ці повідомлення по призначенню. У залежності від адреси повідомлення може бути передане по ЛКМ у межах офісу, через комп'ютерні системи офісу в іншу ЛКМ або ж через комп'ютерні шлюзи чи мости на інший кінець країни. Програми поштової служби повинні бути надійними й швидкими, мати простий користувальницький інтерфейс.

Якість користувальницького інтерфейсу є найважливішим чинником, що визначає, стане чи ні електронна пошта повсякденним засобом спілкування. Основною роботою при підготовці до пересилання повідомлення є підготовка останнього. Текстовий редактор, убудований у систему електронної пошти, з одного боку, повинен бути простим і зручним в застосуванні навіть для новачків, а, з іншого боку, досить могутнім і швидким для тих, хто використовує електронну пошту постійно. Крім того, наявність адресної книги; здатність кодувати й декодувати вкладені файли; можливість перевірки орфографії є важливими параметрами при виборі поштового клієнта.

Відповідне програмне забезпечення повинен установлювати мережний адміністратор, який також заповнює списки користувачів і укажує всі підключені міжмережні мости (якщо вони маютьс'я).

Поштова хост-система – це комп'ютер, зв'язаний з мережею Internet, що одержує поштові повідомлення і направляє їх у відповідні поштові скриньки. У більшості випадків це окремий комп'ютер, зв'язаний з тією ж локальною мережею, що і персональний комп'ютер користувача. В іншому випадку він розташовується в компанії, що надає послуги Internet і тоді зв'язок із хост-системою здійснюється за допомогою модему.

Для обміну листами використовується особлива система адресації. Адреси електронної пошти записуються за допомогою букв латинського алфавіту й цифр. Кожна поштова адреса в мережі Internet складається з двох частин: імені поштової скриньки і поштового хосту – ЕОМ, на якій розташована ваша поштова скринька. Вони розділяються знаком @, наприклад:

user15h@trade4.donbass.com ,

де user15h – ім'я, на яке зареєстрована поштова скринька; trade4.donbass.com – ім'я поштового сервера.

Провайдер призначає адресу електронної пошти, що буде особистим ідентифікатором користувача в Internet. Кожна електронна поштова адреса унікальна.

Приклади e-mail адреси:

dieh@tcc-online.com

Education@imail.ru

frodo@bag\_end.morder.com

Існує три способи роботи з поштою: offline, online і disconnected. У першому випадку (offline) поштовий клієнт зчитує пошту із сервера, зберігаючи її на локальному комп'ютері. Далі можна дивитися пошту, видаляти або групувати листи локально, при цьому немає необхідності в зв'язку із сервером. У другому випадку (online) пошта зберігається на сервері, і всі операції з нею здійснюються безпосередньо на ньому ж. В останньому випадку (disconnected) пошта залишається на сервері, але при цьому на комп'ютері користувача створюється резервна копія, із якою і працює клієнт.

У Internet ряд провайдерів пропонує безкоштовні поштові системи (mail.ru, yandex.ru, rambler.ru, aportal.ru і ін.), створення поштової скринь-

ки в яких супроводжується гарною підказкою і не представляє проблеми (рисунок 6.8).

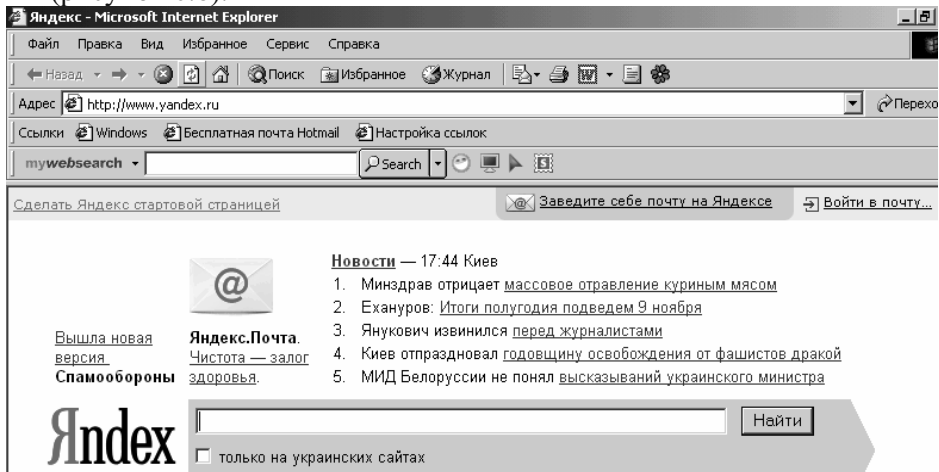


Рисунок 6.8 – Вікно Яндекс із запрошенням створити поштову скриньку

Електронну пошту можна також використовувати для відправлення коротких повідомлень на мобільні телефони. Звичайно, так можна слати не більш 160 знаків (інше буде урізане) англійськими буквами простим текстом. Електронна адреса мобільного телефону виглядає так: 38067номер@sms.kyivstar.net (для абонентів Київстар), 38050номер@sms.umc.com.ua (для абонентів UMC) і т.п.

Аналогічно можна посилати повідомлення на ICQ (даний сервіс буде розглянутий нижче) через електронну пошту: UIN@pager.icq.com, де UIN – номер ICQ. Ще можна одержувати у свою скриньку файли (тільки не дуже великі): треба послати на адресу www4mail@web.bellanet.org лист-запит з порожньою темою, у тілі якого варто написати один рядок: TSOURCE URL, де URL – адреса сторінки чи файлу, що ви хочете одержати.

## Техніка безпеки при роботі з електронною поштою

Працюючи з електронною поштою варто пам'ятати, що легко можна заразити комп'ютер різними вірусами. Як правило, такий вірус розсилає себе без вашого відома усім вашим знайомим, електронні адреси яких зазначені в адресній книзі.

Щоб уникнути цього, можна використовувати різні способи.

1. Мати постійно працюючу програму-монітор (сторож), що перевіряє усе, що надходить в комп'ютер. Комп'ютер може злегка пригальмовувати, зате безпечно.
2. Ніколи не відкривайте файли, прикладені до листів, якщо немає впевненості в їхньому походженні й призначенні. Такі віруси не страшні, поки користувач сам їх не запустить.

Щоб убезпечити себе від вірусів, можна відключити вікно попереднього перегляду листа в поштовій програмі. Для того щоб прочитати лист, потрібно буде двічі клацнути по ньому в списку.

Тепер, одержавши підозрілий лист, перш ніж відкрити його подвійним клацанням, варто ознайомитися з його вхідним текстом. Для цього з контекстного меню треба вибрати команду Свойства (**Properties**), відкрити закладку **Подробности (Details)** і натиснути внизу кнопку **Message Source**. Відкриється вікно з вхідним текстом повідомлення. Отож, якщо в цьому тексті зустрінеться слово IFRAME, відразу видаліть це повідомлення, тому що майже напевно, в ньому "сидить" вірус, що запуститься, автоматично при відкритті цього листа. Ще варто шукати файли з розширенням EXE, COM, VBS, PIF – це теж майже напевно віруси. Щоб видалити заражене повідомлення, користайтесь сполученням клавіш Shift+Del – тоді воно не потрапить у папку "Удаленное", а буде вилучено без сліду.

З погляду безпеки категорично не рекомендується робити свій електронний адрес надбанням широкої громадськості. Він вам потрібний для переписування з визначеними людьми – от нехай тільки ці люди його і знають. Уникайте залишати свою адресу в Мережі, тому що це чревате не тільки вірусами, але і тим, що незабаром він потрапить у руки спамерів. Спам – це нав'язлива реклама. Щоб не витратити дорогоцінний час на відсортовування потрібної інформації, краще пам'ятати: ваша електронна адреса – конфіденційна інформація і подумайте тричі, перш ніж де-небудь її залишити. Крім того, є смисл завести два адреси: один для вузького кола обраних, інший для тих випадків, коли вам усе-таки необхідно повідомити адресу сторонньому, наприклад, при одержанні реєстраційного ключа до програми.

## Списки розсилання

Під **списком розсилання** розуміють програмне забезпечення, що дозволяє розіслати поштове повідомлення деякій групі передплатників. Користувач передає своє повідомлення по спеціальній адресі, а програмне забезпечення списків розсилання потім пересилає його всім учасникам. Якщо при цьому повідомлення проглядається перед розсиланням уповноваженою особою (модератором), то список називається регульованим, у протилежному випадку – нерегульованим. Крім того, модератор виконує адміністративні функції – архівування списків розсилання, виключення зі списків застарілих адрес і ін.

Розсилання можуть бути публічними й закритими. Закриті розсилання – це інструмент загальної роботи, звичайно вони створюються й використовуються усередині якоїсь компанії. Для публічних розсилок, як правило, визначені правила підписки й відмовлення від неї (відписки), а також спілкування в них.

Існують сайти, що пропонують відповідний Web-сервіс – на таких сайтах хто-небудь може створити власне розсилання. Звичайно, розсилання на цих сайтах працює тільки в режимі віщання.

Списки розсилання припускають, що при кожній взаємодії із сервером буде завантажуватися уся електронна пошта, а користувач самостійно буде відсівати усі вже отримані раніше повідомлення.

Проте, списки розсилання залишаються одним із способів входження в Інтернет для тих, хто може підключитися до мережі тільки через електронну пошту.

## Групи новин USENET NEWS

У Internet дуже популярна служба **USENET NEWS** (USENET або News), що іноді називають **телеконференцією**. Ця служба працює приблизно так само, як і електронна пошта, але листи, що одержуються, доступні для загального огляду. Функціонування USENET-системи забезпечується зв'язаними між собою News-серверами (серверами новин). Вони організують циркуляцію повідомлень, зібраних у визначені тематичні групи. News-сервери можуть підтримувати одночасно тисячі телеконференцій, причому набори дискусійних груп відрізняються.

Провайдери можуть установити на своїй базі локальні копії популярних News-серверів і пропонувати своїм клієнтам дану послугу, для цього він надає адресу свого сервера новин. Маються також News-сервера, підключитися до яких можна безкоштовно (msnews, microsoft.com...).

На сьогодні розповсюджується більше десяти тисяч груп новин, що охоплюють широкий спектр тем. Ім'я групи новин, як правило, складається з декількох частин, розділених крапками. При читанні імені зліва направо його частини поступово звужують коло розглянутих питань у цій групі – теми, підтеми, предметні розділи. У групах мережі UserNet крайня ліва частина імені позначає тему, до якої відноситься дана група. Приклади деяких основних категорій груп:

comp – обговорення питань комп'ютерної тематики;

misc – обговорення спеціальних тим;

news – обговорення проблем, зв'язаних із самими групами новин;

rec – кіно, спорт і інші захоплення;

sci – теми, зв'язані з наукою;

soc – соціальні й культурологічні питання;

talk – розмови на загальні теми;

biz – обговорення комерційних питань.

При звертанні до сервера новин фірми Microsoft (адреса msnews.microsoft.com) пропонується цілий набір груп, серед яких є наступні імена: microsoft.public.ru.russisn.outlookexpress; microsoft.public.ru.design.gallery.

Щоб одержати доступ до груп новин, необхідно мати спеціальну програму. **Програма читання груп новин** – це програма, за допомогою якої користувач підписується на необхідні групи новин, переглядає повідомлення, послані іншими учасниками телеконференції, відповідає їм або починає нову дискусію, відправивши своє питання або нове повідомлення. Зокрема, як таку програму зручно використовувати поштовою програмою Outlook Express:

завантажити програму → м. **Сервис** → к. **Учетные записи** → вкладка **Новости** → кнопка **Добавить** → пункт **Новости** → слідувати вказівкам майстра

підключення до Інтернету.

Підключитися до деяких серверів новин можна також через браузер (*броузер*) – спеціальну програму, про яку мова йтиме нижче.

Після підключення до сервера новин, можна починати завантаження списку всіх доступних груп новин, використовуючи відповідну команду. Завантаження списку груп новин займе кілька хвилин, по закінченні завантаження програма читання покаже всі доступні групи новин. Знайшовши в списку необхідну групу новин, на неї варто підписати. Підписка означає внесення цікавих для вас груп новин у спеціальний список, завдяки чому доступ до потрібних конференцій значно полегшується. Користувач буде одержувати на свій комп'ютер постійно обновлюваний список заголовків статей і їхніх авторів, кожен статтю можна розкрити й прочитати.

## World Wide Web

**WWW (World Wide Web чи просто Web)** являє собою одну з найбільш могутніх комунікаційних середовищ нашого часу, через яку користувач може одержати безкоштовний доступ до мультимедіа – документів, що зберігаються на WWW-серверах (Web-site) усього світу. Наприклад, можна відвідувати віртуальні художні виставки, читати міжнародні щоденні газети й журнали, дізнаватися результати спортивних змагань або за допомогою інтерактивних географічних карт подорожувати по світу, клацаючи "мишею" на тій точці земної кулі, у яку хочуть потрапити.

WWW цілком виправдує свою назву: по-англійському web – павутина. WWW можна представити у виді мережі (наприклад, рибацької), що охоплює усі Internet-комп'ютери земної кулі. У кожному вузлі мережі знаходиться комп'ютер, а ділянки між вузлами відповідають лініям комунікацій, через які здійснюється передача даних.

У WWW відсутній центральний орган керування, який визначає, хто і з якою інформацією може виходити в мережу, тому можна сказати, що тут користувач знайде інформацію на будь-яку тему, що його цікавить.

Основна ідея WWW – серверів – використання гіпертексту.

У 70-х роках Тед Нельсон уперше запропонував використовувати гіпертекст. У березні 1989 р. Тім Бернерс-Ли, співробітник Європейської лабораторії фізики елементарних часток (CERN) у Швейцарії поширив пропозицію розробити «гіпертекстову систему» для можливості легкого обміну між різними групами фізиків.

У 90-х м.м. був розроблений новий інформаційний сервіс, що повинен був, по-перше, полегшити доступ до ресурсів Internet, а по-друге, здійснити революцію у відображенні інформації шляхом прямого зв'язування мультимедіа-файлів: зображень, аудіо файлів, відео файлів. У результаті цього проекту і виник World Wide Web.

Завдяки засобам WWW у мережі Internet стало можливим використання технології мультимедіа. Крім того, завдяки програмі WWW-Client, називаною програмою перегляду або "браузером" (від англ. browse -переглядати), можна користатися всіма можливостями Internet з однієї єдиної оболонки користувача.

Представлення інформації у World Wide Web засновано на можливостях так званих гіпертекстових посилань (hiperlinks). У гіпертексті деякі слова виділяються графічно, наприклад, офарблюються в інший колір і/або підкреслюються. Виділення слова свідчить про наявність зв'язку з іншими документом, у якому більш докладно розглядається тема, зв'язана з виділеним словом, що у свою чергу також може містити гіпертекстові посилання на інші джерела інформації. Перехід на зв'язані документи здійснюється простим клацанням по посиланню.

У WWW гіпертекстове посилання не є єдиною можливістю зв'язування різномірної інформації. Оскільки поряд із текстовими файлами можна зв'язувати і бінарні файли, то на такі файли можуть існувати гіпертекстові посилання у вигляді мультимедіа-елементів.

Наприклад, на інтерактивній карті Німеччини кожен населений пункт може бути активізований клацанням на його назві. Після цього користувач одержує доступ до локальних ресурсів Internet даного населеного пункту. У такий же спосіб, виконавши клацання мишею, користувач може програвати аудіо- файли або переглядати відео-файли. Тому в цьому зв'язку вживають узагальнюючий термін "гіпермедіа".

Для реалізації можливостей гіпертексту була розроблена спеціальна мова програмування – HTML, за допомогою якого звичайний текст розмічається особливим образом, після чого документ стає гіпертекстовим. Конструкції цієї мови – теги – дозволяють керувати шрифтом, кольором тексту і фону, визначати посилання, уставляти графіку, аудіо й відео і т.п. Самі теги при прочитанні сторінки не видні, а видні тільки результати їхньої дії.

Одиницею гіпертекстових даних WWW є Web-сторінка – мінімальний фрагмент гіпертексту, який можна завантажити й прочитати за один раз.



Для ідентифікації гіпертекстової сторінки в мережі використовуються так називаний URL (Universal Resource Locator) – Універсальний Локатор Ресурсів. У ньому міститься інформація про спосіб передачі даних між клієнтом і сервером (протокол), імені сервера, на якому знаходиться сторінка, шляхи до неї у дереві каталогів сервера, імені сторінки. Адреса будь-якого web-сервера починається з аббревіатури http, що позначає вид протоколу переданих даних (у даному випадку це HyperText Transmission Protocol – протокол передачі гіпертексту).

Приклад: <http://www.donduet.edu.ua/uk/index.shtml> – повна адреса головної сторінки нашого університету.

На сервері або вузлі WWW може міститися кілька сторінок гіпертексту. Існують наступні способи організації гіпертекстових сторінок – ієрархічний, лінійний і у виді павутини.

- Ієрархічна організація припускає наявність декількох рівнів сторінок – основна, потім посилання на сторінки другого рівня, із другого рівня посилання на сторінки третього рівня і т.д.
- Лінійна організація припускає наявність рівноправних сторінок одного рівня, зв'язаних між собою посиланнями.
- Організація у виді павутини припускає безліч сторінок, що зв'язані різноманітними гіперпосиланнями, що утворюють складну структуру.

Для прочитання гіпертекстових документів розроблені спеціальні програми – браузері.

Браузер (browser) – програми, використовувані для приєднання до вузлів Web, перегляду Web – сторінок і одержання доступу до інших ресурсів. Графічні зображення, текстова інформація, певним чином оформлена й розташована, виводяться на екран за допомогою браузера, що інтерпретує спеціальні знаки форматування, впроваджені в Web – сторінку.

Поряд зі своїми власними додатками, WWW пропонує просту у звертанні оболонку користувача для виходу в найбільший інформаційний потік світу – Internet.

### **Програми Telnet, FTP, Gopher**

Програма **Telnet** була розроблена для забезпечення дистанційного доступу до віддаленого комп'ютера в INTERNET. При цьому комп'ютер

користувача виступає як термінал, підключений до великого комп'ютера (mainframe).

На відміну від комп'ютерів, термінали не мають власних обчислювальних можливостей. Завдяки наявності монітора і клавіатури, вони тільки представляють можливість доступу до якого-небудь комп'ютера.

Системи mainframe/terminal використовуються, наприклад, у туристичних бюро, де з їхньою допомогою здійснюється запит наявності вільних місць на авіарейси й бронювання авіаквитків безпосередньо в аеропортах.

Telnet емулює цю ситуацію. Це означає, що при Telnet-зв'язку з Internet-комп'ютером усе відбувається таким чином, начебто користувач сидить в іншій кімнаті за терміналом, безпосередньо зв'язаним із Internet-комп'ютером. Telnet не надає власних команд для керування іншим комп'ютером, але забезпечує прямий доступ до нього.

Одним із найцікавіших сервісів Internet є програма FTP, що пересилає файли відповідно до протоколу **FTP**. FTP – це також спеціальний сервер, керуючий файловим FTP-архівом, назва методу доступу в по-кажчику URL.

(**File Transfer Protocol** – протокол передачі файлів).

FTP був розроблений для обміну файлами між комп'ютерами в Internet. Комп'ютери, на яких маютьяся файли для загального користування, називають FTP-серверами.

За допомогою численних FTP – серверів, розкиданих по усьому світі, Ви можете одержати доступ більш ніж до 10000 Гб безкоштовного програмного забезпечення для всіх операційних систем. Особливість у тім, що розглянутий сервіс не має обмежень на розмір переданої інформації (файлу).

За допомогою однієї з таких програм можна встановити зв'язок із яким-небудь FTP-сервером у Internet і "поритися" у каталогах, відкритих для загального доступу, щоб знайти цікавлячі файли і скопіювати їх у свій комп'ютер, подібно тому, як копіюються дані з одного жорсткого диска на іншій у тому самому комп'ютері. При використанні типової FTP-програми користувачу надаються можливості керування двома ідентичними частинами: своїм персональним комп'ютером і обраним FTP-сервером.

Система **Gopher** була розроблена для роботи з численними інформаційними ресурсами INTERNET в університеті штату Міннесота в

1991 році. Робота з нею нагадує керування даними за допомогою утиліт Windows Explorer або File Manager.

Особливість системи Gopher полягає в тому, що за допомогою ієрархічної системи меню вибирається шлях пошуку інформації або файлу. Необхідні протоколи INTERNET вже інтегровані в систему Gopher, тому всі технічні процеси відбуваються на задньому плані. Це означає, що Ви вибираєте одну із записів у меню, а Gopher виконує все інше, незалежно від того, що потрібно зробити: установити Telnet-зв'язок з іншим комп'ютером або провести FTP-сеанс. Переходячи до нового елемента меню, Ви встановлюєте зв'язок із віддаленим комп'ютером на підставі отриманої від попереднього комп'ютера інформації про Internet-адреси зазначених ресурсів.

Існує більш 750 Gopher-серверів. Цей інформаційний сервіс став уже настільки популярним, що ресурси Internet, керовані Gopher-сервером, називають Gopher Space (простір Gopher).

### **Безпосереднє спілкування в Internet**

В Інтернеті маються цікаві технологічні рішення, що реалізують можливість спілкування в реальному часі.

**IRC** (Internet Relay Chat – бесіда через Internet) – служба (сленговий chat або чат), що схожа на UseNet, але обмін повідомленнями в ній ведеться без затримок.

Служба з'явилася у Фінляндії в 1988 р. І швидко поширилася по Інтернет. Спілкування в цьому середовищі підтримується IRC-серверами, зв'язаними в однойменні мережі. Сервери забезпечують взаємодію багатьох користувачів, що групуються по визначеним рубрикам (каналам) – по тематиці, проблемам, мові спілкування і т.п. Кожен канал обслуговується конкретною IRC- мережею, відповідні мережі не перетинаються між собою. Приклади популярних каналів: Dalnet, Undernet server, IRCnet. Великий список чатів представлений на сайті [www.dir.ru](http://www.dir.ru) [енцикл.].

Щоб з'єднатися з найближчим IRC-сервером і поспілкуватися на обраному каналі, необхідно установити програму – клієнта служби, наприклад, програму mIRC ([www.mirc.co.uk](http://www.mirc.co.uk)) або Mirc.

Програми легко встановлюються і настроюються. Програми пропонують список IRC-серверів або можна увести відомий користувачу IP-адрес сервера.

При реєстрації користувач повинен вибрати псевдонім («ник»). Тому що «ник» зобов'язаний бути унікальним, користувачеві пропонується відразу зарезервувати альтернативний варіант (Alternative). Він буде використовуватися у випадку конфліктів з основним ником. Також, при вході, можна вказати Invisible mode, режим невидимості, що збереже появу користувача в чаті в таємниці і, як наслідок, убереже від наполегливої реклами

Мережа IRC некомерційна, і тому, більшість серверів існує на суспільних початках. Природно, це не може не позначитися на якості роботи.

У 1996 р. в Інтернет з'явилася служба, що швидко завоювала величезну популярність. Мова йде про ICQ, що у народі ласкаво називають Аської.

**ICQ** (I seek you – я шукаю тебе) – служба, що дає можливість користувачам мережі обмінюватися повідомленнями в реальному часі, організувати чат (chat – бесіда), пересилати файли і багато чого іншого. Підтримують ICQ-систему сервери Mirabilis, що забезпечують з'єднання по особливому ефективному протоколу. Про підключення до конкретного сервера не треба піклуватися, програма – клієнт (ICQ) сама усе зробить.

Програма – клієнт ICQ розповсюджується безкоштовно, легко встановлюється й налаштовується. При установці програми користувач вибирає собі псевдонім, а ICQ привласнює йому унікальний реєстраційний номер (UIN).

Чим гарна ICQ. Вона істотно випереджає пошту по оперативності, і вона набагато зручніше IRC у спілкуванні хоча б тому, що не вимагає додаткових знань (куди і як підключатися).

При спілкуванні в ICQ не має значення, із якого комп'ютера користувач входить в Інтернет, головне, щоб на цьому комп'ютері була програма – клієнт ICQ, і він був зареєстрований.

На кожен реєстраційний номер (UIN) автоматично заводиться персональний web-сайт, потрапити на який можна з браузера за адресою [http://www.icq.com/особистий\\_UIN](http://www.icq.com/особистий_UIN). Можна також відсилати повідомлення через поштову програму, див. вище.

Для безпечної роботи в ICQ при налаштуванні програми рекомендується установити прапорець Do not allow other to see my IP address (не показувати іншим мою IP адресу).

До речі, ICQ уже має аналоги – Instant Messenger від Netscape, Yahoo Messenger від Yahoo, Odigo (програма відрізняється зручним інтерфейсом і широкими можливостями, англomовна – <http://www.odigo.com>).

В Інтернеті свої правила спілкування. Бути знайомим з нетикетом потрібно кожному жителю віртуального світу. Тут існує безліч писаних, а ще більше не писаних правил поведінки. Загалом, більшість жителів не вважають спамерів (спам – нав'язлива реклама), грубіянів і хуліганів.

Так уже трапилося, що спілкування в чатах, форумах, через ICQ і пошту відбувається за допомогою набору тексту на клавіатурі. Не кожен це вміє робити швидко, а комусь хочеться висловити свої емоції не тільки словами. А ще бувають моменти, коли під час розмови хочеться дуже швидко відповісти. Напевно, у силу цих і ще інших причин, з'явився ряд загальноприйнятих скорочень, перелік основних із них показаний у додатку А [<http://uaportal.com/Guide/Begin/Address/index1.html>].

### **Обмін інформацією «мобільний телефон-Internet»**

Відкритий протокол і технічний стандарт Wireless Application Protocol (WAP) розроблений з ініціативи фірми Unwired Planet (Phone.com). Його широке застосування привело до того, що можна обмінюватися інформацією через Всесвітню мережу безпосередньо з мобільних телефонів, без посередництва комп'ютера. WAP збагачує набір таких послуг мобільного зв'язку, як телефонія і короткі текстові повідомлення (SMS) [6].

Треба відзначити, що по протоколу WAP на мобільний телефон зміст Web-ресурсів безпосередньо не передається. У Internet інформація представлена у вигляді HTML-сторінок, робота з якими припускає швидкі комунікації, могутні процесори, великі екрани й обсяги пам'яті комп'ютерів. Те, чим не володіють звичайні мобільні телефони. Тому для WAP використовується спеціалізована мова розмітки – Wireless Markup Language (WML), більш проста й строга, чим HTML.

Нині в Україні вже існують сотні WAP-сайтів. Найбільш популярними є [wap.gala.net](http://wap.gala.net), [wap.kyevstar.net](http://wap.kyevstar.net), [wap.prostir.com.00](http://wap.prostir.com.00)

Для перебування інформації у «мобільному» секторі Internet виникла потреба в пошукових системах. WAPJAG ([www.wapjag.com](http://www.wapjag.com)) на сьогодні найбільший каталог по WAP-ресурсах і також найбільш попу-

лярна пошукова машина. Поки доступна лише англійська версія цього порталу, однак розроблювачі незабаром обіцяють русифікований варіант.

Кращим серед росіян «Wap-розвідувачем» тепер вважається *Яндекс* (wap.yandex.ru).

В українському секторі можна відзначити портал wap.uaport.net, розроблений компанією ElVisti. Один із самих насичених інформацією українських WAP-ресурсів уже сьогодні доступний російською й англійською мовами.

Найбільш корисні й зручні послуги WAP, зв'язані з доступом до електронної пошти. Завдяки їм користувач може в будь-який момент переглянути свіжу кореспонденцію на дисплеї мобільного телефону.

Останнім часом така послуга з'явилася в найбільших безкоштовних серверів (wap.mail.ru, wap.imail.ru, wap.newmail.ru).

## **6.2.4 Організація комплексної роботи користувача в Internet**

З усього перерахованого, яке охоплює далеко не усі види послуг Internet, видно, що користувачеві варто освоїти досить велику кількість програм. Але й у цій області про користувача подбали.

Для організації комплексної роботи користувача у Internet найбільш популярні програми Netscape Navigator, Internet Explorer. Це інтегровані пакети, що забезпечують роботу браузера, електронну пошту, програми групової роботи в Internet і ін. функції.

Обидва пакети функціонально схожі, і між їхніми розроблювачами йде жорстка конкурентна боротьба.

Обидва пакети розповсюджуються безкоштовно.

Коротка характеристика програми Internet Explorer.

Існує кілька способів одержання дистрибутивного набору Internet Explorer. Його можна одержати з Internet або придбати на компакт-диску.

Останній варіант вимагає менше часу, чим переписування файлів по мережі. Microsoft розповсюджує дистрибутивний набір Internet Explorer, що містить усі компоненти пакета Internet Explorer. Деякі провайдери послуг Internet (ISP, Internet Service Provider, – організація, що надає зв'язок із Internet) і продавці програмного забезпечення також розповсюджують комплекти Internet Explorer, якими можна скористатися.

Хоча установка з компакт-диску вимагає менше часу, мережний варіант також не позбавлений достоїнств. Головне з них полягає в тім, що, одержавши Internet Explorer із Web-сервера Microsoft, користувач стає власником самої свіжої версії. Правда, одержання усього Internet Explorer може зайняти кілька годин у залежності від можливостей модему, але ж це робиться лише один раз!

Компанія Microsoft передбачила прекрасний спосіб поповнення уже встановленого пакета Internet Explorer новими компонентами або відновлення тих із них, для яких з'явилися більш свіжі версії. Потрібно перейти за адресою

<http://www.microsoft.com/downloads/search.aspx?displaylang=ru>  
і вибрати необхідну програму для завантаження і відновлення.

Компоненти Internet Explorer.

- **Основні компоненти Microsoft Internet Explorer.** Сюди входять компоненти робочого столу, призначені для інтеграції з Web, компоненти підтримки Java і майстер підключення до Internet. Інтегрований з Web робочий стіл дозволяє використовувати Web-сторінки в якості "шпалери" робочого столу. Засоби підтримки Java містять у собі виконуючу систему Java Virtual Machine і компілятор Java Just-in-Time і призначені для виконання аплетів Java у середовищі Internet Explorer. Майстер підключення до Internet полегшує з'єднання з ISP.
- **Засоби комунікації.** Цю групу утворюють Microsoft NetMeeting, Outlook Express і Chat. NetMeeting – це засіб для інтерактивного обміну по Internet аудіо- і відеоінформацією, а також додатками. Outlook Express є програмою обробки електронної пошти й новин. Chat дозволяє спілкуватися по мережі, вводячи й читаючи повідомлення.
- **Засоби мультимедіа.** Сюди входять програми NetShow, VRML Viewer, Multimedia Controls, DirectShow і Direct. NetShow дозволяє розповсюджувати по Internet відео – і аудіопередачі. VRML Viewer відкриває перед вами світи віртуальної реальності. До складу Multimedia Controls входять засоби створення динамічних мультимедійних ефектів. DirectShow являє собою програму підтримки цифрового відео. Direct –

це набір інтерфейсів прикладного програмування для створення програм взаємодії зі спеціальним периферійним устаткуванням.

- **Компоненти для розробки Web-сторінок.** У цю групу входять FrontPage Express – графічний редактор Web-сторінок; Видавець Web – майстер, призначений для розміщення Web-сторінки на сервері; служба Windows для створення особистого Web-сервера, що дозволяє розповсюджувати Web-сторінки по Internet безпосередньо з вашого комп'ютера; допоміжне програмне забезпечення, що надає доступ до баз даних через Web-вузли.

- **Додаткові компоненти.** Це змішана група. У неї входять наступні сервісні програми. Microsoft Wallet дозволяє зберігати на комп'ютері в захищеній формі дані кредитної картки і свою адресу, що надає можливість швидко виконувати в Internet фінансові операції. Пакет Supplemental Web Fonts пропонує два додаткових сімейства шрифтів TrueType, розроблених для Web. IE Sound Pack – це набір використовуваних у IE звукових ефектів. Диспетчер для складання розкладу виконання задач Internet Explorer.

- **Засоби мовної підтримки.** У цю категорію входять компоненти, що підтримують додаткові мови: японський, корейський, китайський (традиційний і спрощений), а також мови пан'європейської групи (центрально-європейські мови, кирилиця, грецька, турецька і мови прибалтійських народів).

При установці Internet Explorer користувач може вибрати мінімальний, звичайний або повний варіант конфігурації у залежності від своїх можливостей і потреб.

Вхід у програму – через ярлик на робочому столі або кнопка **Пуск** → к. **Програми** → **Internet Explorer** (рисунок 6.9).

Настроювання пакета здійснюється : м.**Сервіс** → к. **Окно Обозревателя**.

Пакет Internet Explorer постачений прекрасною довідкою російською мовою, що дає досить докладні вказівки по використанню різних сервісів.



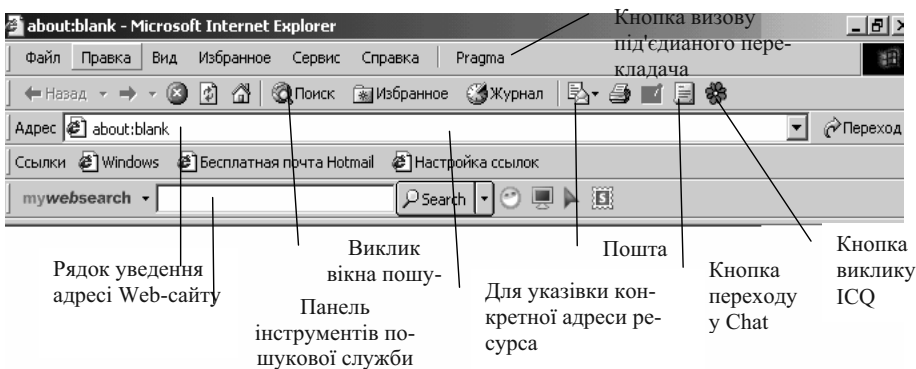


Рисунок 6.9 – Вікно Internet Explorer перекладач

## 6.2.5 Інформаційно-пошукові системи INTERNET, технологія пошуку

У Internet міститься інформація практично про усе і доступ до неї забезпечений цілодобово. Єдина проблема полягає в тім, як її витягти. І отут на допомогу приходять пошукові служби.

У WWW існують десятки пошукових систем (search engines), що можуть надати мільйони й мільйони потрібних вам посилань на сторінки WWW. Є глобальні, регіональні, орієнтовані на якусь мову або спеціалізовані по визначених типах об'єктів (музика, живопис, реферати і т.п.). Приклади найбільш відомих пошукових служб: AltaVista, Northern Light, HotBot, Excite, Infoseek, Lycos і Yahoo!, Google, Rambler, АПОРТ!, Yandex, Uaport, TopPing.

На серверах пошукових служб виконується спеціальна програма, називана роботом (robot). Іноді такі програми називають павуками (spider) або гусеницями (crawler). Робот збирає інформацію про вміст Web, повертаючи на свій сервер усі виявлені сторінки. По них складається докладний опис доступних ресурсів – індексна база даних. Індекссування виконується таким чином, щоб усі слова (за винятком союзів) могли використовуватися при обробці запиту від потенційного користувача. При кожному новому обході мережі індексна база обновляється.

Накопичена роботами інформація проглядається редакторами, що відкидають свідомо недостовірні, маревні відомості і складають каталоги ресурсів – бази даних з адресами ресурсів.

Наприклад, найбільший пошуковий каталог Yahoo! містить більш мільйона записів, і формують його біля півтори сотень редакторів. Каталоги мають ієрархічну структуру і дозволяють досить близько підійти до шуканого матеріалу.

Слід зазначити, що пошук у Internet досить творчий процес, що вимагає від користувача деякої гнучкості мислення. Справа в тім, що комп'ютери в мережі не вміють нічого, крім як одержувати й передавати інформацію. Задача користувача – чітко задати їм, що конкретно він хоче одержати.

Існують два основних методи пошуку:

- по ключових словах (key words) і
- логічному дереву (subject tree).

За допомогою ключових слів завжди легше знайти щось конкретне, і важче – щось загальне й невизначене. А за допомогою каталогів зручно шукати інформацію на загальні теми (природа, мистецтво, комп'ютери, медицина і т.д.). Любий каталог має рубрикатор. Таким чином, інформація деякою мірою відсортована. Кожна адреса поставляється анотацією представлених матеріалів. При рішенні досить стандартної пошукової задачі саме каталог, а не пошукова машина виявляються більш прийнятними для початку пошуку.

При звертанні до пошукової служби користувач попадає на початкову сторінку, що дозволяє формулювати критерій пошуку і задавати оператори, що визначають параметри пошуку.

Пошук ведеться шляхом знаходження відповідності між заданими користувачем ключовими словами й умістом Web-сторінок, повідомлень груп новин, файлів FTP-серверів і інших ресурсів Internet. Пошукова служба повертає список вузлів (звичайно порціями по 10 або 20 вузлів), причому першими будуть повернуті вузли з найвищою ймовірністю відповідності сформульованому критерію.

При індексації документів пошукові машини вираховують так називану «вага» слова на сторінці – співвідношення кількості повторів на сторінці заданого слова до загальної кількості слів на сторінці документа. Якщо заданий запит, що складається з декількох слів, то більш релевантними будуть документи, у яких сукупна вага слів буде максимальною. Однак, при підрахунку ваги не враховується, поруч або окремо стоять дані слова, і тому немає гарантій, що в перших документах міститься мак-

симальна кількість повторень словосполучень. Цілком можливо, що такого словосполучення там узагалі не буде.

Звернутися до пошукових служб можна безпосередньо з вікна браузера Internet Explorer за допомогою убудованої панелі пошукової служби або через вікно пошуку (кнопка **Поиск**), прямо задавши адресу пошукової служби. На рисунку 6.9 показано вікно з убудованою панеллю інструментів пошукової служби MSN search, що набудована як мегапошукова система (рисунок 6.10).

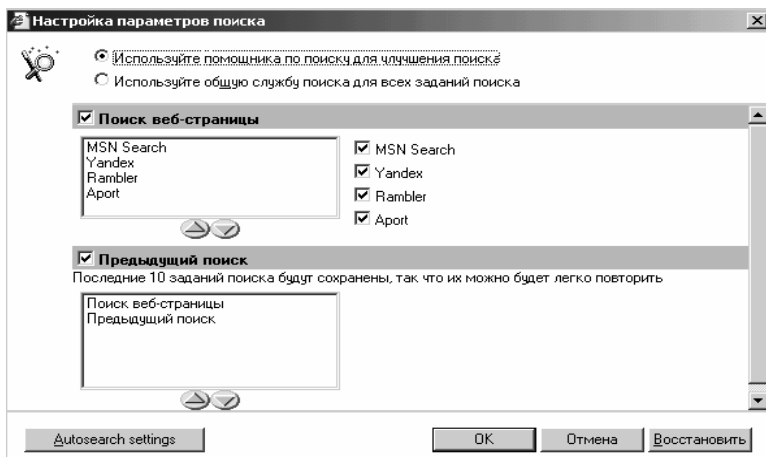


Рисунок 6.10 – Вікно налаштування параметрів пошуку

Якщо користувачу необхідна інша пошукова служба, то в таблиці 6.4 показані адреси найбільш популярних служб.

Практично всі пошукові служби використовують для створення запитів той самий підхід. За допомогою знаків "плюс" (+) і "мінус" (-) можна визначити наявність або відсутність визначеного слова на шуканій Web-сторінці або в статті. Наприклад, заданий запис у полі запиту **+chocolate -cake** означає, що буде знайдена інформація, у якій міститься слово *chocolate* і є відсутнім слово *cake*.

Таблиця 6.4. – Адреси пошукових служб

Назва	URL	Назва	URL
Загальний пошук по WWW		Пошук програм на FTP	
Google	www.google.com	Rus-soft	Ru-soft.unet.ru
AltaVista	www.altavista.com	Freeware	www.freeware.ru
HotBot	www.hotbot.com	Download	www.download.ru
Lycos	www.lvcos.com	Archie	http://www.ftpsearch.ntnu.no
Yahoo!	www.vahoo.com	FileSearch	http://www.ftpsearch.com
Excite	www.excite.com	FileSearch.ru	http://www.filez.com
Яндекс	www.vandex.ru		http://www.ftpsearch.city.ru
Апорт	www.aport.ru	Спеціалізований пошук	
Rambler	www.rambler.ru	Пошук людей –	www.dubna.ru/eros/  www.hi.ry/email/
ПОЗНАЧКА	meta-ukraine.com	пошукова сис-	
Uaport	uaport.net	тема Э-Росс	
TopPing	www.topping.com.ua	Сервер Ні	

Фраза в лапках («Фізичні явища») буде оброблена як єдине ціле. Більшість пошукових служб також підтримує логічні вираження, що визначають комбінацію слів при завданні критерію – це И, ИЛИ, НЕ (вони ж AND, OR, NOT або & (+), |, –). Наприклад, Jazz AND (Stockton OR Malone) означає, що в результатах пошуку повинне обов'язково згадуватися слово *Jazz* і бути присутнім слово *Stockton* або *Malgne*. Два останніх слова не можуть бути присутнім одночасно.

При пошуку зручно використовувати функції «розширений пошук», «пошук у знайденому», за допомогою яких можна деталізувати умови пошуку (рисунк 6.11).

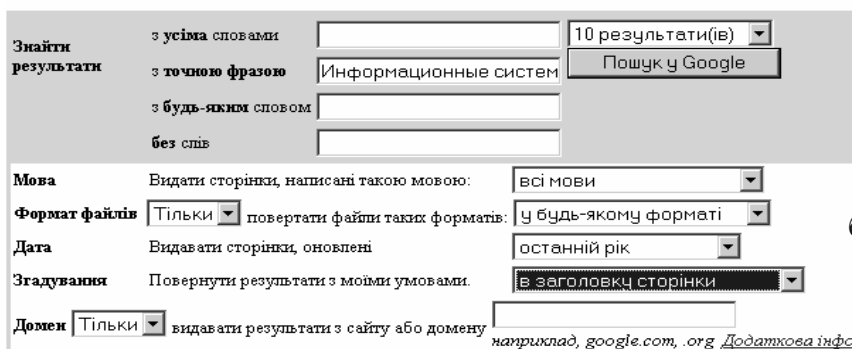
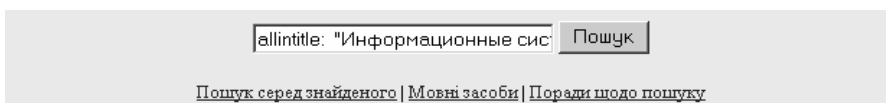
Пошук безкоштовної програмної продукції можна виконувати на FTP – серверах. Легше це зробити за допомогою російських сховищ, що дають опис програм і коментарі до них. У таблиці 6.4 дані деякі адреси.



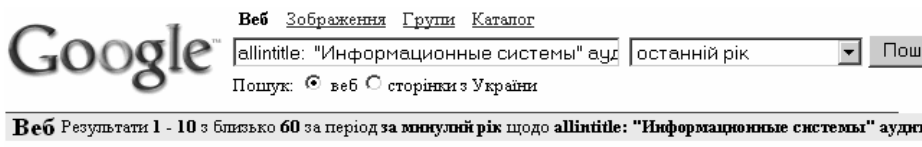
**Компьютерные информационные системы**

Бюджетирование как инструмент управления. КИС:Бюджетирование – программа для бюджетного планирования и контроля. Методические рекомендации, как составить ...  
 www.cis2000.ru/ - 22к - 10 Лис 2005 - Збережено на сервері - [Подібні сторінки](#)

a)



b)



b)

Рисунок 6.11 – Вікна пошукової системи Google при звичайному, розширеному і «у знайденому»

### Питання і завдання для самостійного виконання

1. Що означає поняття «комп'ютерна мережа»?
2. Для чого необхідна мережна карта і її основні функції?

3. Яке програмне забезпечення необхідне для роботи локальної комп'ютерної мережі?
4. Що означає поняття «комунікаційний протокол» для комп'ютерних мереж?
5. Чим обумовлені проблеми мережної інформаційної безпеки і як вони вирішуються для локальних мереж?
6. Чим відрізняються локальні комп'ютерні мережі від глобальних?
7. Які технічні і програмні засоби необхідні для підключення локальних мереж до регіональних і глобальних?
8. Способи підключення комп'ютера користувача до Internet?
9. Що таке «сервіси Internet» і які сервіси Ви знаєте?
10. Засоби групового й індивідуального спілкування в Internet?
11. Типи адрес у Internet? Що таке URL у Internet?
12. Можливості електронної пошти?
13. Які засоби забезпечують користувачу можливість пошуку в Internet?
14. Способи пошуку інформації в комп'ютерній мережі?
15. Створіть свою поштову скриньку на одному з поштових серверів, наприклад, на Yandex, Rambler або Mail. Перевірте працездатність скриньки, пославши повідомлення самому собі або кому-небудь із друзів.
16. Виконайте пошук інформації з теми «Нові засоби обміну інформацією в комп'ютерних мережах» за допомогою будь-яких двох пошукових систем і з використанням їхніх можливостей по раціональному пошуку.

## Література

1. Войтюшенко Н.М., Остапець А.І. Комп'ютерна техніка і програмування. Частина 1. Основи підготовки користувача ПК: Навчальний посібник по базовій підготовці для студентів денної і заочної форм навчання. – Донецьк, ДонДует, 2001 – 150 с.
2. Внутренний мир World Wide Web/ Пер. с англ. – К.: Издательство «Диал-Софт», 1997. – 544 с.
3. Вульф К. Технологии Microsoft: проблемы и их решение. Универсальный справочник системного администратора/ Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 736 с.
4. Информатика: Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів/ За ред. О.І. Пушкаря. – К.: Видавничий центр «Академія», 2003. – 704 с.
5. Компьютеры, сети, Интернет. Энциклопедия./Новиков Ю и др. – СПб.: Питер, 2002. – 928 с.
6. Лозікова Г.М. Комп'ютерні мережі: Навчально-методичний посібник. – К.: Центр навчальної літератури, 2004. – 128 с.
7. Основы современных компьютерных технологий: Учебное пособие/ под ред. проф. Хомоненко А.Д. – СПб.: КОРОНА принт, 1998 – 448с

## Розділ 7. Основи алгоритмізації й програмування (Інструментальні засоби програмування)

Сьогодні персональний комп'ютер стає звичайним засобом на робочому місці фахівця, значить і сам фахівець повинен володіти звичайним для комп'ютера наповненням. У це поняття входять не тільки користування пакетами прикладних програм, але й уміння поставити задачу, настроїти програму або змінити її деякі функції.

Багато професійних пакетів прикладних програм містять засоби програмування, що дозволяють актуалізувати додатки.

Наприклад, у пакеті автоматизації діловодства Microsoft Office, комплексне рішення практичних задач можна забезпечити при використанні мови програмування Visual Basic. Тобто грамотний користувач повинен бути знайомим з питаннями алгоритмізації й програмування.

Велика частина спеціалізованих пакетів має або свою, спеціально розроблену мову програмування для настроювання й актуалізації, або використовує стандартні мови програмування.

Наприклад, пакет для інженерів, будівельників, архітекторів Autocad, дозволяє розширювати й змінювати потрібні функції за допомогою мови програмування Lisp, більшість популярних бухгалтерських пакетів під Windows («1С: Бухгалтерія», «Парус-предприятие», «Акцент-бухгалтерія») дозволяють виконувати настроювання на нові форми документів, зміни в планах рахунків і ін. за допомогою убудованих мов програмування. Без цих засобів при сьогоденній динаміці господарського життя придбані пакети працювали не більш би півроку.

Сьогодні широко використовуються об'єктно – орієнтовані мови програмування, які значно полегшують створення програм із зручним інтерфейсом. Крім того, практичне спостереження таке: якщо людина розібралася з однією з мов програмування, то освоєння інших мов іде значно простіше.

Тому в курсах базової підготовки фахівців вивчення мови необхідно й обов'язково. Тоді можна буде сказати, що наші фахівці не просто вміють працювати на ПК, а володіють інтелектуальним інструментом.

## 7.1 Загальна характеристика етапів рішення задач на ОМ

Електронна обчислювальна техніка покликана полегшити розв'язання різних задач для людини. Безпосередньому рішенню задачі на ЕОМ, у т.ч. і ПК, передує тривалий період підготовки цього рішення.

Рішення всякої задачі проходить наступні етапи:

1. **Постановка задачі** – установлюються цілі розв'язуваної задачі, умови й вимоги до рішення.

Описується результатна й вхідна інформація про досліджуваний об'єкт, явище, ситуації.

Візьмемо самий простий приклад: «У батька з матір'ю синів на 2 більше, ніж дочок, а всього чоловіків у будинку вдвічі більше, ніж жінок. Скільки в батька з матір'ю синів і скільки дочок?»

Звичайно, задачі, розв'язувані на робочих місцях різних фахівців, значно об'ємніше і складніше, для їхнього грамотного опису вимагаються відповідні знання. Тому постановку задачі звичайно виконує фахівець, що працює на даному робочому місці. Хто ж краще нього може знати весь обсяг і нюанси роботи?

### 2. Математичне формулювання задачі.

На цьому етапі встановлюються в остаточному виді ті формули, математичні залежності чи рівняння (модель), що підлягають рішенню. Математична постановка задачі визначається фахівцем з даної проблеми разом із математиком.

Продовжимо наш жартівливий приклад. Позначимо число синів через  $Y$ , а число дочок – через  $Z$ . Різниця цих величин, за умовою, дорівнює 2.

$$Y - Z = 2$$

Виразимо далі співвідношення між числом чоловіків ( $Y + 1$ ) і числом жінок ( $Z + 1$ ) у родині:

$$Y + 1 = 2 * (Z + 1)$$

або:

$$Y - 2 * Z = 1$$



Задача, таким чином, звелася до системи двох лінійних рівнянь. Не важко вивести, що в родині три сини й одна дочка.

При математичному формулюванні багатьох проблем виникають подібні, але більш складні системи, вирішити які не так просто, як нашу задачу.

### 3. *Вибір чисельного методу.*

Чисельні математичні методи дозволяють звести рішення самих різних задач до послідовного виконання елементарних операцій, для виконання яких пристосований той чи інший ПК.

У більшості випадків та сама задача може бути вирішена декількома чисельними методами. На даному етапі вибирається такий метод, що щонайкраще забезпечує виконання поставленої задачі (забезпечує необхідну точність, швидкість рішення ...).

4. *Алгоритмізація.* Це перший етап програмування, у процесі якого розробляється алгоритм рішення задачі. За допомогою алгоритму записується хід рішення задачі.

5. *Програмування.* Здійснюється переклад алгоритму на мову конкретної машини чи мову, зрозумілу машині (алгоритмічну).

6. *Налагодження програм.* Задача даного етапу полягає в тому, щоб шляхом апробування на машині розробленої програми, виявити помилки, допущені на всіх попередніх етапах. Налагодження програм виконується на контрольному прикладі, що повинен охоплювати всі можливі напрямки рішення задачі, – приклад з уже відомими результатами. Якщо розрахунок по програмі збігся з результатами контрольного приклада, то задача вирішена правильно.

Якщо синтаксичні помилки й помилки в написанні команд програми може допомогти знайти комп'ютер, то логічні помилки виявляються тільки за допомогою контрольного приклада

7. *Рішення задачі на ЕОМ.* Після налагодження програми складаються інструкції з роботи з програмою й підготовці вхідних даних. По цих інструкціях обчислення на комп'ютері може виконувати будь-як користувач, знайомий із правилами роботи на даній машині.

Частина із зазначених етапів, а саме 1 – 4, не вимагають наявності комп'ютера, 6 і 7 етапи виконуються безпосередньо на конкретній ЕОМ. 5 – й етап, у залежності від застосовуваного класу машин, може визначатися конкретним видом комп'ютера чи ні.

У процесі перерахування етапів рішення задачі зустрілися поняття (алгоритм, програма), що вимагають пояснення, а етапи 4 – 6

вимагають більш детального розгляду у виду їхньої важливості для роботи на комп'ютері.

## 7.2 Поняття алгоритму. Основні властивості й методи запису алгоритмів

Самі того не помічаючи, ми повсякденно зіштовхуємося з поняттям алгоритму.

Наприклад, розраховуючи за покупку, ми складаємо або множимо числа по цілком визначених правилах, у необхідній послідовності і, незалежно від касира, приходимо до результату.

Або, виходячи з будинку, ви думкою вже склали план дій – куди ви йдете, яким шляхом.

Знайомлячись з побутовими приборами, більш складною технікою, ми керуємося інструкцією, що цілком однозначно і детально визначає наші дії по експлуатації й збереженню прибору.

Ще один приклад: необхідно обчислити  $Y$  по формулі:

$$Y = (d * b + c) / d$$

Обчислення ми звичайно проводимо в наступній послідовності:

1. Множимо  $d$  на  $b$ , одержуємо проміжний результат обчислень, рівний  $S$ .
2. Складаємо отриманий результат  $S$  із  $c$ , одержуємо проміжний результат обчислень, рівний  $K$ .
3. Поділяємо отриманий результат  $K$  на  $d$  і одержуємо остаточний результат обчислень, рівний  $Y$ .

У наведеному вище прикладі ми приходимо до одержання результату в результаті виконання у визначеній послідовності ряду операцій над вхідними даними.

**Ось цей строгий і однозначний опис послідовності операцій, необхідних для реалізації обраного методу рішення задачі і являє собою алгоритм.**

Академік Марков визначає це поняття таким чином:

**«Під алгоритмом варто розуміти точне приписання, що визначає обчислювальний процес, що веде від вхідних даних, що варіюються, до шуканого результату».**

У більш широкому значенні, під терміном «алгоритм», розуміють приписання, що дозволяє єдиним і формальним способом вирішувати будь яку задачу, незалежно від її характеру й походження.

Сам термін «алгоритм» утворився від імені узбецького математика Аль-Хорезмі, який ще в 13-м столітті дав алгоритми додавання, вирахування, множення й ділення чисел.

Розглянуті приклади алгоритмів дають можливість сформулювати деякі **загальні властивості**, притаманні будь-якому алгоритмові:

1. Властивість **детермінованості (визначеності)**.

Алгоритм повинен забезпечувати однозначність результату рішення задачі при заданих вхідних даних. Правила і їхня послідовність повинні бути чіткі і строго визначені, що не допускають різночитань.

2. Властивість **дискретності** – розчленованість алгоритму на окремі елементарні кроки, можливість виконання яких людиною або машиною не викликає сумніву.

3. Властивість **масовості** – алгоритм повинен будуватися таким чином, щоб була можливість його використання для різних вихідних даних. Тобто, алгоритм повинен забезпечувати рішення будь-якої задачі з класу однотипних задач. Наприклад, у задачі обчислення  $Y$ , кожен може задати свої конкретні значення для  $d$ ,  $b$  і  $c$  ( $d$  – не дорівнює нулеві) і задача буде вирішена.

Численність і трудомісткість перерахованих вище етапів підготовки й реалізації задачі, робить безглуздим однократне її рішення на комп'ютері. Тільки багаторазове рішення багатьма користувачами задачі робить корисним і вигідним створення для неї програми.

4. Властивість **результативності** – при будь-якій вхідній інформації і строгому дотриманні приписань, що визначають обчислювальний процес, алгоритм повинен приводити до одержання шуканого результату.

5. Властивість **ефективності** – досягнення результату найбільш раціональним шляхом (за мінімальне число кроків).

Розглянуті загальні властивості алгоритмів є емпіричними. Вони сформульовані на підставі досвіду, аналізу всіх існуючих алгоритмів.

Очевидно, що для процесу алгоритмізації досить важливо, щоб алгоритм можна було записати в компактній, наочній і легко зрозумілій формі.

Існують різні **способи запису алгоритмів**. Це – *словесний, формульно-словесний, операторний, блок-схемний (або графічний)* способи запису і запис *за допомогою алгоритмічних мов*.

Приклад, розглянутий вище, являє собою словесний опис алгоритму.

Приклад *словесно-формульного* опису алгоритму для обчислення вираження:

$$S = \sum_{i=1}^n f(i)$$

- M1.  $i = 1, S = 0$
- M2. Якщо  $i > n$ , перейти на M5
- M3.  $S = S + f(i)$
- M4.  $i = i + 1$
- M5. Вихід.

Розповсюдженим методом запису алгоритмів є *блок-схемний метод*.

**Блок-схема є графічним зображенням структури алгоритму.**

Кожен етап алгоритмічного процесу зображується у вигляді різних геометричних фігур: прямокутників, ромбів, кружків і т.д. У середині цих фігур описується зміст даного етапу обчислень у довільній формі. Причому, якщо опис етапу обчислень виявляється громіздким, то усередині блоку ставиться номер етапу, а опис дається в додатку до блок-схеми.

Між окремими, графічно зображеними етапами алгоритмічного процесу, установлюються зв'язки у вигляді ліній потоку, що вказують для кожного етапу його можливих спадкоємців.


Нами будуть використовуватися графічні позначення, що відповідно до ГОСТом 19.003 – 80 зображуються в такий спосіб (таблиця 7.1).

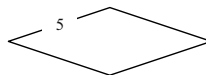
*Лінії потоку* — , ← , ↑ , | – застосовують для вказівки напрямку алгоритму рішення (без стрілки, якщо лінія спрямована зліва направо і зверху вниз, із стрілкою – в інших випадках).

Усі блоки блок-схеми нумеруються вгорі ліворуч у розриві лінії (крім блоків "пуску – останова", з'єднувача й переходу на іншу сторінку).

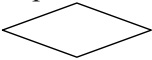
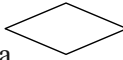
Таблиця 7.1. – Витримка з ГОСТ 19.003 – 80

Найменування	Позначення	Функція
1	2	3
1. Процес		Виконання операції або групи операцій, у результаті яких змінюється значення, форма представлення або розташування даних.
2. Рішення		Вибір напрямку виконання алгоритму або програми, у залежності від деяких умов.
3. Уведення – вивід		Перетворення даних у форму, придатну для обробки (уведення) або відображення результатів обробки (вивід)
4. Пуск – останов		Початок, кінець, переривання обробки даних або виконання програми.
5. З'єднувач		Указівка зв'язку між перерваними лініями потоку блок-схеми ( у кружку вказується номер блоку ).
6. Перехід на іншу сторінку		Посилання на іншу сторінку при переносі блок-схеми.

Наприклад: 



Як використовуються конкретні позначення операцій у блок-схемах, розглянемо на прикладах у наступному питанні.

Усі блоки, крім , мають по одному виходу, а  – має тільки два виходи (спадкоємця).

Метод запису алгоритмів за допомогою *алгоритмічних мов* застосовується при рішенні простих задач, коли необхідна послідовність операцій очевидна. Якщо ж задача складна або громіздка, то спочатку складають блок-схему рішення, а потім програму для комп'ютера.

## 7.3 Алгоритми основних видів обчислювальних процесів

Аналіз задач, які розв'язуються людиною, показав, що алгоритм будь-якої задачі може бути представлений як комбінація базових алгоритмів (обчислювальних процесів). До таких процесів відносяться:

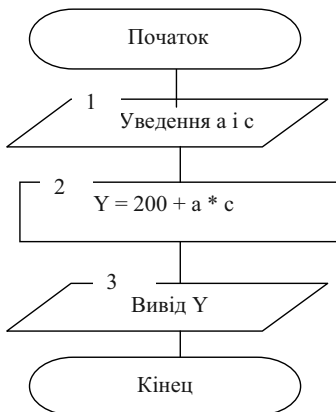
1. *Прості (лінійні) обчислювальні процеси.*
2. *Розгалужені обчислювальні процеси.*
3. *Циклічні обчислювальні процеси.*

### 7.3.1 Простий не розгалужений обчислювальний процес

Алгоритми простих не розгалужених процесів не містять жодного етапу, що має більш одного спадкоємця, тобто алгоритми реалізуються простою послідовністю операцій.

Приклад 7.1.\*- Записати алгоритм обчислення значення  $Y$  по формулі:

$$Y = 200 + a * c$$



\* – Далі на цих же прикладах будуть показані прийоми програмування на Visual Basic.

### 7.3.2 Розгалужені обчислювальні процеси

Алгоритми обчислювальних процесів, що розгалужуються, містять хоча б один етап, що має більш одного спадкоємця, який включається в обчислювальний процес у залежності від виконання деякої умови. Інакше можна сказати, що задача має кілька шляхів рішення, обраних у залежності від виконання тієї або іншої умови. Таким чином, алгоритм повинен не тільки описувати, але і вибирати потрібний хід рішення в залежності від вхідних даних або проміжних результатів. Напрямки, по яких може йти обчислювальний процес, називаються його гілками.

Прикладами такого процесу є: нарахування заробітної плати, якщо використовуються різні форми оплати праці, добір і розрахунок вузлів і механізмів із різних деталей і т.п.

Приклад 7.2. – Записати у вигляді блок – схеми алгоритм обчислення:

$$y = \begin{cases} a * x + b & , \text{якщо } x \geq 0 \\ a * x - b & , \text{якщо } x < 0 \end{cases}$$

При записі алгоритмів обчислювальних процесів, що розгалужуються, необхідно дотримувати *наступні* вимоги:

1. У різних гілках можна використовувати ті самі позначення перемінних.
2. Обчислення або процеси, що повторюються в різних гілках блок – схеми, виносяться за межі розгалуження (у нашому прикладі це – обчислення  $a * x$  і друк результату).
3. Складні умови обчислень розбиваються на ряд простих.

З урахуванням перерахованих вимог, рішення задачі буде виглядати наступним образом (рисунок 7.1):

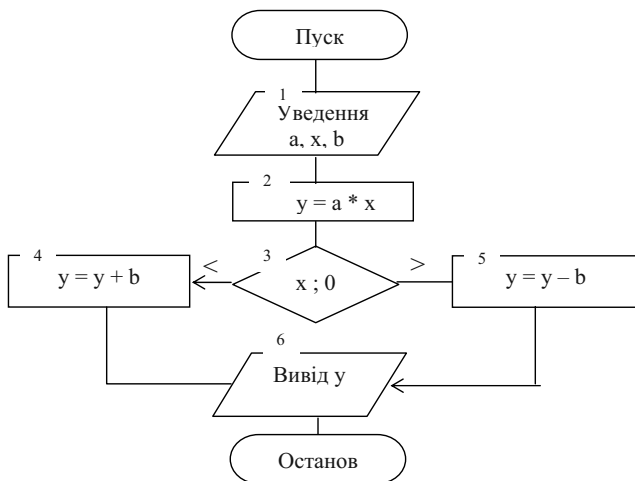


Рисунок 7.1 – Блок-схема простого розгалуженого процесу

Розглянута задача мала 2 гілки, але кількість гілок у задачі може бути 2, 3, 4 і більш.

Приклад 7.3. – Записати у вигляді блок-схеми алгоритм обчислення:

$$y = \begin{cases} x / a - x / b & \text{при } a \leq 5 \text{ і } x > 6, \\ (a + b) / x & \text{при } a \leq 5 \text{ і } x = 6, \\ a^3 + b * e^{2.5} & \text{в інших випадках} \end{cases}$$

У наведеному прикладі в кожній гілці мається складна умова, що повинна бути розбита на ряд простих. Крім того, у першій гілці додається ще 2 умови –  $a \neq 0$ ,  $b \neq 0$ .  $e^{2.5}$  – це експотенціальна функція. Блок-схема приклада представлена на рисунку 7.2.

### 7.3.3 Циклічні обчислювальні процеси

Більшість практичних задач (економічних, технічних, наукових) вирішуються на ЕОМ за допомогою циклічних алгоритмів. В алгоритмах рішення таких задач зустрічаються ділянки обчислень, що представляють собою багаторазове повторення однієї і тієї ж послідовності операцій.



Наприклад, при обчисленні  $y = a^x$ , операція множення повторюється  $x$  раз. При  $x = 5$ , операція множення повторюється 5 раз:  $y = 1 * a * a * a * a * a$ , а при  $x = 121$  – множення повторюється 121 раз. Чому з'явилася множення на 1, буде видно при освоєнні методики пошуку робочих формул.

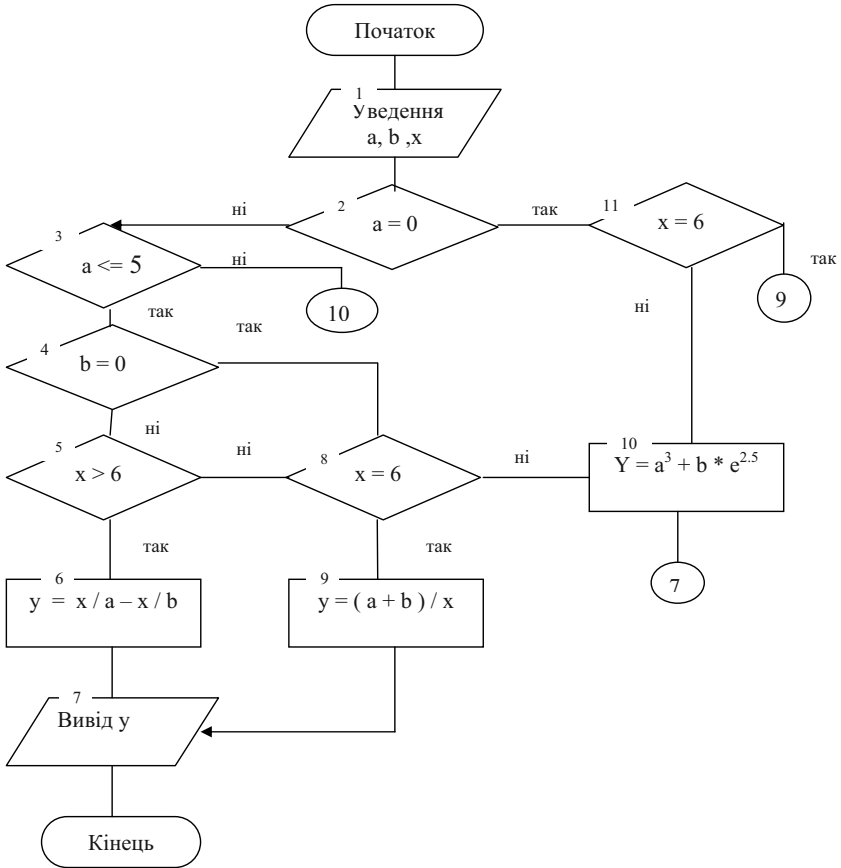


Рисунок 7.2 – Блок-схема до приклада 7.3

При обробці багато рядкових економічних документів процес обробки рядка документа повторюється стільки раз, скільки рядків у даному документі.

Таким чином, **циклічний обчислювальний процес** – це процес, що містить  $n$  раз повторювані етапи обробки інформації.

Цикл повинен повторюватися доти, поки не виконається деяка умова виходу з циклу. Узагальнена блок – схема циклічного обчислювального процесу представлена на рисунку 7.3.

При уважному погляді на блок-схему видно, що основою її робочої частини є робочі формули. Робочі формули і дозволяють підраховувати результат при будь-якій заданій кількості циклів. Виходить, перш ніж малювати блок-схему, необхідно скласти робочі формули. Пошук робочих формул по всіх типах циклів виконується за єдиною методикою, що буде розглянута нижче на конкретному прикладі.

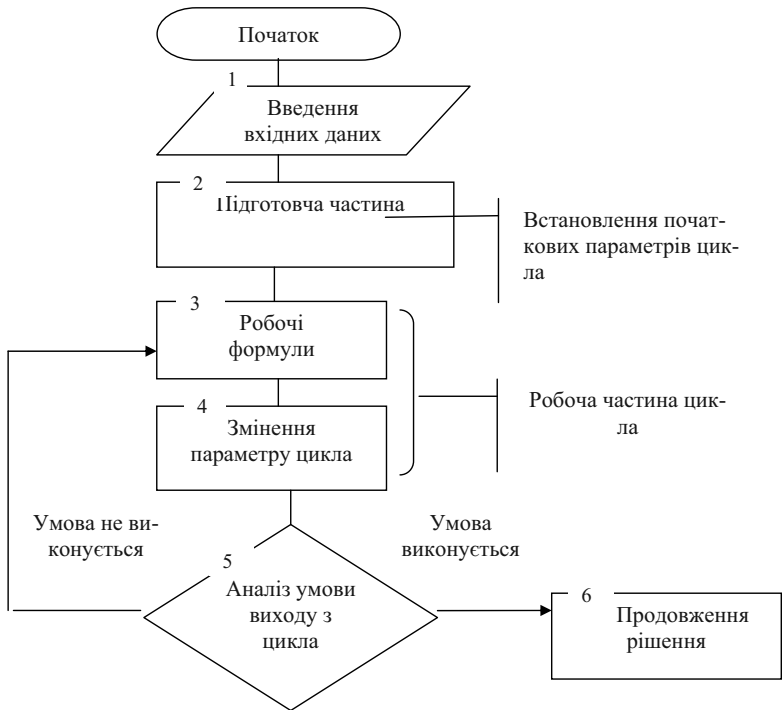


Рисунок 7.3 – Узагальнена блок-схема циклічного обчислювального процесу

Зміст робочих формул дозволяє визначити, що встановлювати в підготовчій частині, як змінювати перемінну (параметр) циклу.

Крім того, варто звернути увагу, що в першому блоці вводяться поодинокі перемінні, а кілька однотипних операцій по уведенню або виводу даних уже повинні входити в цикл, як багаторазово повторювані.

**Циклічні процеси** можна класифікувати в залежності від кількості й складності вхідних циклів на *прості (один цикл)* і *складні (цикл у циклі або розгалуження в циклі)*. У свою чергу **прості цикли** підрозділяються в залежності від способу виходу з циклу на *арифметичні* й *ітераційні*.

### Арифметичні цикли

Циклічний процес, у якому відомо заздалегідь кількість повторень циклу, називається арифметичним циклом.

Наприклад, обчислення величини  $y = a^x$ , при заданому значенні  $x$ , або  $y = n!$ , при заданому значенні  $n$  реалізується за допомогою арифметичних циклічних процесів. Кількість повторень циклу в них задають  $x$  і  $n$  відповідно.

Для підрахунку кількості повторень циклу в арифметичних циклах уводять спеціальні перемінні, які називають *лічильником циклів*. Лічильник являє собою цілу змінну величину. У підготовчій частині блок – схеми необхідно задати початкове значення для лічильника. Якщо лічильник працює на збільшення, тобто початкове значення його 0 чи 1, і після кожного виконання циклу лічильник збільшується на одиницю, то такий лічильник називається *прямим*. Якщо лічильникові у якості початкового значення привласнюється максимальне число повторень циклу, і після кожного виконання циклу він зменшується на одиницю, то такий лічильник називається *зворотним*.

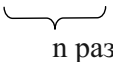
Приклад 7.4. – Скласти блок-схему обчислення  $y = n!$ .

Розгорнуто формулу  $y = n!$  можна записати:  $y = 1 * 2 * 3 * \dots * n$

Пошук робочої формули:

**1 крок** – записати й простежити, як росте формула при збільшенні кількості операцій. Позначимо кількість виконуваних операцій через  $i$ .

$$\begin{array}{ll}
 1) & i = 1 \quad y = 1; \\
 2) & i = 2 \quad y = 1 * 2; \\
 3) & i = 3 \quad y = 1 * 2 * 3; \\
 \dots & \dots \\
 4) & i = n \quad y = 1 * 2 * \dots * n
 \end{array}$$



**2 крок** – позначити кожен наступний крок через попередній, тобто другий через перший, третій через другий і т.д.

$$\begin{array}{l}
 1) i = 1 \quad y = 1 \\
 2) i = 2 \quad y = 1 * 2 = y * 2 \\
 3) i = 3 \quad y = 1 * 2 * 3 = y * 3 \\
 \dots \\
 4) i = n \quad y = 1 * 2 * \dots * n = y * n
 \end{array}$$

Елементи однакові і в першому кроці позначені через у, тому вводимо це позначення на другому кроці

Аналогічно на 2) кроці  $1*2$  позначено через у, що і відображуємо на 3) кроці і т.д.

**3 крок** – із усіх записаних формул вибрати однакові елементи, що стоять ліворуч від першого знака дорівнює (для розглянутого приклада – це  $y =$ ), а потім усі однакові елементи праворуч від останнього знака дорівнює (це  $y *$ ). Останній співмножник – змінна величина і просто записати його ми не можемо. Варто подивитися на записи в кроках  $1 \dots n$  і знайти перемінну, котра також змінювалася б, як і другий співмножник. У нашому прикладі це  $i$ . Тоді робоча формула може бути записана в такий спосіб:

$$y = y * i$$

Формула позначає, що попередній добуток буде братися з чарунки у, збільшуватися на наступний співмножник і міститися знову в ту ж саму чарунку з ім'ям у. Саме такий прийом і дозволяє нарощувати результат потрібну кількість разів.

З робочої формули видно, які перемінні необхідно установити в підготовчій частині (блок 2 рисунок 7.4).

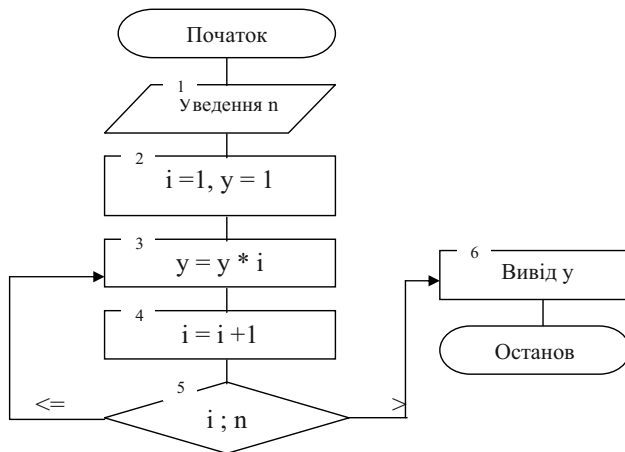


Рисунок 7.4 – Блок-схема обчислення  $y = n!$

Щоб формула працювала, у правій частині повинні бути відомі значення перемінних  $y$  і  $i$ . Фактично  $i$  – це лічильник (або параметр циклу) і початкове значення його можна установити  $1$ , яке йде з кроку  $1$ . Значення  $y$  також треба підібрати таке, щоб уже на першому кроці ( $1$ )  $y = 1$ ) результат вийшов правильний. Очевидно, що для множення це  $1$ .

Умовою виходу з циклу буде перевищення значення лічильника  $i$  над необхідною кількістю повторень циклу  $n$ .

### Ітераційні цикли

Не у всіх задачах заздалегідь відома кількість повторень циклу. Прикладами таких задач є задачі обчислення квадратного кореня з цілих позитивних чисел, обчислення тригонометричних функцій за допомогою рядів, інтегралів наближеними методами, уведення набору даних перемінної довжини. В економіці багато задач планування, проектування вирішуються ітераційними методами.

В ітераційних циклах задається точність шуканого результату, що дозволяє визначити момент виходу з циклу, або яка-небудь інша умова.

Приклад 7.5. Скласти блок-схему обчислення квадратного кореня для позитивного числа  $x$  (рисунок 7.5):

$$y_k = y_{k-1} + \left( \frac{x}{y_{k-1}} - y_{k-1} \right) / 2 = \left( y_{k-1} + \frac{x}{y_{k-1}} \right) / 2$$

Де:  $y_{k-1}$  – попереднє значення кореня  
 $y_k$  – наступне обчислене значення

У якості початкового наближеного значення можна задати будь-яке число, але бажано ближче до  $x$ .

Точність обчислення визначається по  $\left| y_k^2 - x \right| \leq \xi$   
 формулі:

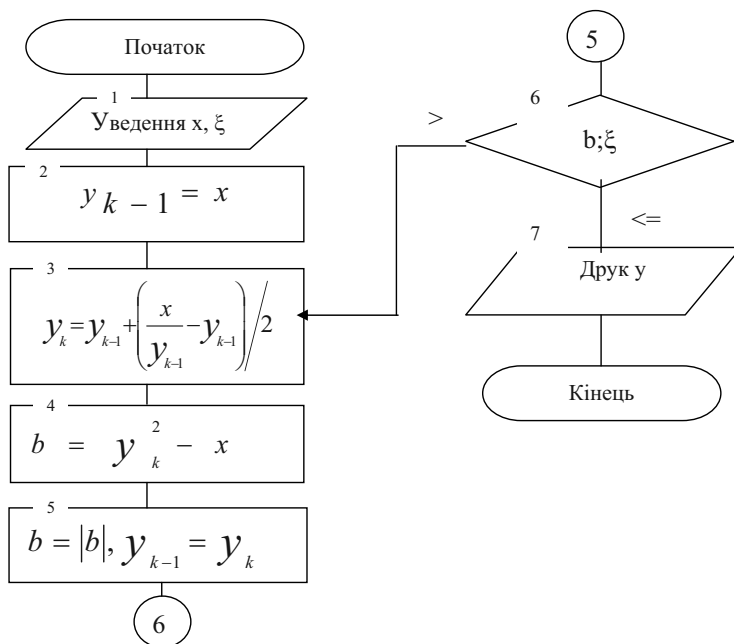


Рисунок 7.5 – Блок-схема до приклада 7.5

### Циклічні алгоритми, що обробляють масиви

В усіх задачах, що розглядалися раніше, використовувалися перемінні, котрі мали різні імена і характеризували різні сторони об'єкта. Але в практичних задачах, особливо при обробці експериментальних і економічних даних, часто приходится мати справу з даними, що характеризують ту саму властивість об'єкта, явища, даними, логічно взаємозалежними. Сукупність перемінних, як уже визначалося раніше, називається масивом, а окремі перемінні – елементами масиву. Звичайно для позначення елементів одного масиву використовуються букви з індексом (підрядковим номером).

Наприклад. Мається масив сум по найменуваннях сировини, що надійшли в комору підприємства суспільного харчування –  $x$

$$X = x_1, x_2, x_3, \dots, x_i, \dots, x_n, \quad i = 1 \div n$$

де:  $x_i$  – сума приходу  $i$ -го найменування сировини в комору;

$i$  – номер, привласнений найменуванню сировини.

При рішенні задач із масивами варто пам'ятати, що введення даних в ОЗП комп'ютера здійснюється по одному елементу масиву і фактично являє собою циклічний процес.

Приклад 7.6. Необхідно скласти блок-схему обчислення суми надходження сировини в комору підприємства суспільного харчування по формулі:

$$y = \sum_{i=1}^n x_i$$

Визначення робочої формули:

1.  $i = 1 \quad y = x_1$

2.  $i = 2 \quad y = x_1 + x_2 = y + x_2$

3.  $i = 3 \quad y = x_1 + x_2 + x_3 = y + x_3$

.....

n.  $i = n \quad y = x_1 + x_2 + x_3 + \dots + x_n = y + x_n$

Робоча формула:

$y = y + x_i$
---------------

Блок – схема задачі представлена на рисунку 7.6.

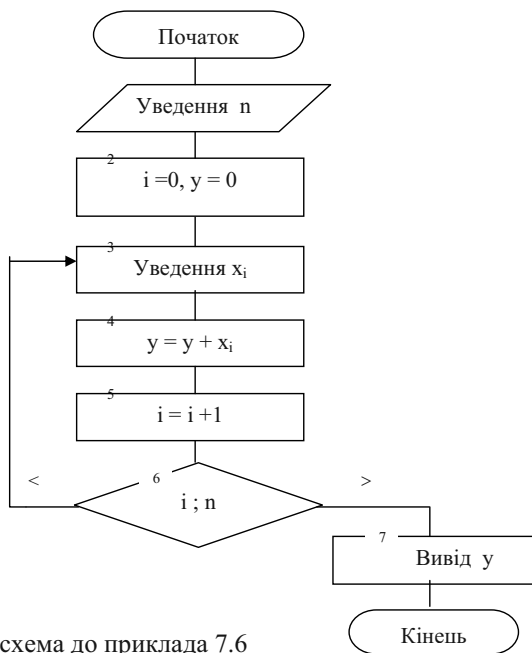


Рисунок 7.6 – Блок-схема до прикладу 7.6

Цикли, що обробляють масиви, можуть бути арифметичними (з умови задачі відома кількість повторень) як у приведеному прикладі, і ітераційними.

Приклад 7.7. – Необхідно скласти алгоритм уведення масиву чисел невизначеної довжини, але відомо, що наприкінці масиву стоїть нуль (0). Це може бути той же масив про надходження сировини:

$$X = x_1, x_2, x_3, \dots, x_i, \dots, 0$$

Блок – схема задачі представлена на рисунку 7.7

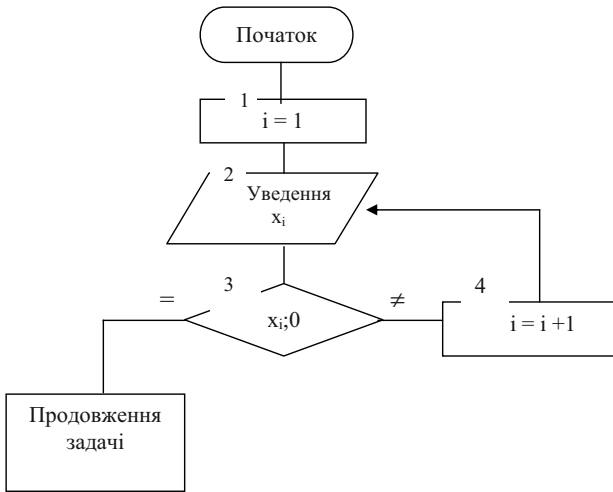


Рисунок 7.7 – Блок-схема до приклада 7.7.

### Складні цикли

Складні цикли, це цикли, що складаються з двох і більш циклів, вкладених друг у друга або з циклів, що містять розгалуження. Приклади таких циклів – сума всіх елементів матриці, підрахунок товарообігу підприємства за місяць, якщо є дані продажі по окремих найменуваннях товарів щодня, обробка пачки однотипних документів і т.д. Розглянемо приклад циклу в циклі.

Приклад 7.8. – Скласти блок-схему алгоритму підрахунку товарообігу підприємства за місяць (S), якщо є дані продажі по окремих найменуваннях товарів щодня (x<sub>ij</sub>),



$j = 1 \div n$ ,  $n$  – кількість найменувань товарів,  
 $i = 1 \div m$ ,  $m$  – кількість робочих днів у місяці.

$$S = \sum_{i=1}^m \sum_{j=1}^n x_{ij} \quad \text{– вхідна формула.}$$

Пошук робочої формули:

$i = 1$

1)  $j = 1 \quad S = x_{11}$

2)  $j = 2 \quad S = x_{11} + x_{12} = S + x_{12}$

3)  $j = 3 \quad S = x_{11} + x_{12} + x_{13} = S + x_{13}$

.....

n)  $j = n \quad S = x_{11} + x_{12} + \dots + = S + x_{1n}$

$i = 2$

1)  $j = 1 \quad S = x_{21}$

2)  $j = 2 \quad S = x_{21} + x_{22} = S + x_{22}$

3)  $j = 3 \quad S = x_{21} + x_{22} + x_{23} = S + x_{23}$

.....

n)  $j = n \quad S = x_{21} + x_{22} + \dots + x_{2n} = S + x_{2n}$

.....

$i = m$

1)  $j = 1 \quad S = x_{m1}$

2)  $j = 2 \quad S = x_{m1} + x_{m2} = S + x_{m2}$

3)  $j = 3 \quad S = x_{m1} + x_{m2} + x_{m3} = S + x_{m3}$

.....

n)  $j = n \quad S = x_{m1} + x_{m2} + \dots + x_{mn} = S + x_{mn}$

Робоча формула –  $S = S + x_{ij}$

Початкове значення суми  $S = 0$ . Зовнішнім є цикл по днях, а внутрішнім – по найменуваннях товарів, тобто правий індекс змінюється швидше. Пройти у внутрішній цикл можна тільки через блок підготовки зовнішнього. Аналіз умови виходу з циклу варто перевіряти по кожному індексі. Блок-схема задачі показана на рисунку 7.8.

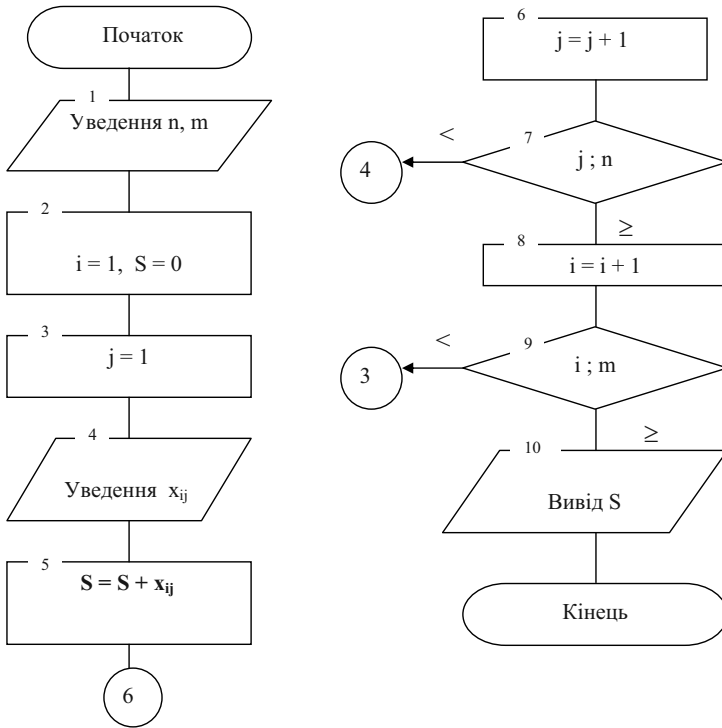


Рисунок 7.8 – Блок-схема до прикладу 7.8.

## 7.4 Алгоритмічні мови: поняття і класифікація

Маючи алгоритм рішення задачі, людина повинна представити його мовою «зрозумілій» машині – записати у вигляді програми.

ПК безпосередньо виконує програми **машинною мовою** програмування даної ЕОМ. При цьому програма являє собою набір окремих команд комп'ютера у двійкових кодах. Ці команди є досить «простими», наприклад, додавання, множення, порівняння або пересилання окремих даних. Кожна команда містить у собі відомості про те, яка операція повинна бути виконана (код операції), із якими операндами (адреси даних або безпосередньо самі дані) виконуються обчислення і куди (адреса) повинен бути поміщений результат.

Машинні мови були першими мовами програмування – **нульовий рівень** алгоритмічної мови. Якщо згадати, як росте у двійковому коді зображення чисел із їхнім збільшенням, то можна представити наскільки об'ємними були у минулому перші програми (одній машинній команді відповідала 1 команда мовою програмування – співвідношення 1:1) . Для програмування необхідно було знати особливості конкретної ЕОМ і велику кількість реалізованих нею операцій (команд), а пошук помилок серед моря нулів і одиниць виростав у велику проблему. Тому фахівці активно займалися розробкою нових, більш зручних і могутніх алгоритмічних мов.

У своєму розвитку алгоритмічні мови пройшли цілий ряд рівнів. Рівень мови програмування визначається семантичною (значеннєвою) ємністю його конструкцій і ступенем його орієнтації на програміста. Чим більше мова орієнтована на людину, тим більше співвідношення машинних команд і команд мови програмування, тим вище її рівень.

**Перший рівень** – мнемокоди, мови символічного кодування (*Асемблери*). Мова Асемблера дозволяє програмістові користатися мнемонічними кодами операцій, привласнювати замість двійкових кодів зручні імена чарункам і областям пам'яті, а також задавати найбільш зручні схеми адресації, хоча залишається співвідношення 1:1 або 1: до декількох.

**Другий рівень** – мова *Макроасемблера* є розширенням мови Асемблера шляхом включення в нього макрозасобів – засобів визначення й використання нових, більш могутніх команд як послідовності базових інструкцій, що трохи підвищує його рівень.

Мови Асемблера й Макроасемблера застосовуються системними програмістами професіоналами з метою використання всіх можливостей устаткування ЕОМ і одержання ефективної за часом виконання і по необхідному обсязі пам'яті програми. На цих мовах звичайно розробляються відносно невеликі програми, що входять до складу системного програмного забезпечення: драйвери, утиліти й інші.

Мови нульового – другого рівня машинно-орієнтовані, тому що вимагають знань архітектури конкретних моделей комп'ютерів. Це не зручно для масового користувача, тому наступним кроком було створення проблемно-орієнтованих мов, що не вимагають детального знання ЕОМ.

**Третій рівень** – *проблемно-орієнтовані мови*, наприклад, Паскаль, Си, Бейсік, Фортран, PL/1. Мови АЛГОЛ, ФОРТРАН, ПАСКАЛЬ орієнтовані на задачі інженерного, обчислювального характеру, у них добре був розвинений апарат перетворення складних формул, мова КОБОЛ, PL/1 – мали розвинену систему представлення складної структури даних і були орієнтовані на рішення економічних задач, мова Си – мова для системних програмістів і т.д. Співвідношення в таких мовах уже було 1: до багатьох.

Для перекладу програми, написаної мовою високого рівня, у відповідну машинну програму використовуються *мовні процесори (транслятори)*. Розрізняють два види трансляторів: інтерпретатори й компілятори.

*Інтерпретатор* – це програма, що одержує вхідну програму і в міру розпізнавання конструкцій вхідної мови реалізує дії, описувані цими конструкціями.

*Компілятор* – це програма, що приймає вхідну програму і породжує на своєму виході програму, записувану на об'єктній (проміжній) мові програмування (об'єктну програму). В окремому випадку об'єктною може служити машинна мова, і в цьому випадку отриману на виході транслятора програму можна відразу ж виконати на ЕОМ.

**Четвертий рівень** – *об'єктно – орієнтовані* алгоритмічні мови ( Visual Basic, Delphi, Java, C++Builder і Visual C++ і т.д.).

В основі об'єктно орієнтованих мов програмування лежить поняття *об'єкта*, а суть його виражається формулою: «об'єкт = дані + процедури». Кожен об'єкт інтегрує в собі деяку структуру даних і доступні тільки йому процедури обробки цих даних, називані *методами*. Об'єднання даних і процедур в одному об'єкті називається інкапсуляцією.

Для опису об'єктів служать класи. *Клас* визначає властивості й методи об'єкта, що належить цьому класу. Відповідно, будь-який об'єкт можна визначити як екземпляр класу.

Програмування розглянутого стилю полягає у виборі наявних або створенні нових об'єктів і організації взаємодії між ними.

Алгоритмічні мови всіх зазначених рівнів відносяться до *процедурних мов* або *мов системного рівня* – усі вони в тім або іншому ступені складаються з послідовності операторів (інструкцій), що задають процедуру рішення задачі.

До непроцедурних мов програмування можна віднести функціональні або аплікативні (ЛИСП) і логічні або реляційні (ПРОЛОГ) мови.

Роль основної конструкції у функціональних мовах грає вираження, дією є виклик функції, єдиним способом розчленування програми на частині є введення імені для функції. Це характеризує аплікативні мови як мови програмування дуже високого рівня.

Мови логічного програмування, особливо Пролог, широко використовуються в системах штучного інтелекту.

Центральним поняттям у логічному програмуванні є *відношення*. Програма являє собою сукупність визначень відносин між об'єктами (у термінах умов або обмежень) і мети (запиту). У реляційному програмуванні потрібно тільки специфікувати факти, на яких алгоритм ґрунтується, а не визначати послідовність кроків, що потрібно виконати. Логічні програми, у принципі, мають невелику швидкодію, тому що обчислення здійснюються методом проб і помилок, пошуком із поверненнями до попередніх кроків.

Звичайно кожен додаток, створений за допомогою зазначених мов, являє собою монолітну сутність, розроблену із самого початку і до самого кінця, але ситуація в області ПЗ змінилася.

Інфраструктура сучасного бізнесу складається з незліченного числа додатків, пристроїв, протоколів, джерел даних і оболонок. Щоб організація могла працювати ефективно, усі ці ресурси необхідно координувати, а, крім того, повинен існувати спосіб, що дозволяє легко створювати нові додатки, здатні на новому рівні використовувати можливості існуючих ресурсів. Більш того, щоб компанія могла реалізувати нові технології, необхідно мати можливість інтегрувати їх з існуючими системами. Сьогодні усе більше й більше додатків створюються за рахунок інтеграції наявних ресурсів.

Для зв'язування окремих компонентів програмного забезпечення використовуються **мови сценаріїв**, що грають роль свого роду клею. Це дозволяє вивести програмування на більш високий рівень і прискорити процес розробки програм.

Мови опису сценаріїв, або **скриптові** мови програмування, використовуються практично скрізь. Звичайно, левина частка застосування цих технологій приходить на веб-додатки. Але і на ізольованих від Інтернету і узагалі від якої-би, то не було мережі комп'ютерах теж активно використовуються скрипти. Навіть найпростіша опера-

ційна система вже на початку свого завантаження використовує спеціальні текстові файли для налаштування й оптимізації. А це ні що інше, як найпростіші сценарії.

Усе вище сказане надихнуло розроблювачів на створення **скриптових** мов. Коли власне "програма" створюється у вигляді звичайного текстового файлу спеціального формату (сценарію) і передається для обробки на задалегідь встановлений у системі клієнта інтерпретатор. Так з'явилися **HTML, PHP, Perl, JavaScript, JScript, VBScript, Python, Rexx, Tcl** і ще багато різних чудових універсальних і вузькоспеціальних **мов опису сценаріїв**.

Завдяки сполученню роботи мов системного рівня і мов сценаріїв величезні поклади програмного забезпечення можна замінити на невеликі й прості програмні компоненти, що роблять тільки те, що нам необхідно і їх можна поєднувати один з одним.

## Питання й завдання для самостійного виконання

1. Назвіть етапи підготовки рішення задач на ПК, що виконуються без допомоги комп'ютера.
2. Виконати математичне формулювання наступної задачі:

Визначити кількість порцій відбивних котлет, якщо відомо, що в їдальню надійшла тушу яловичини першої категорії вагою 183 кг. Котлетне м'ясо складає 43% від усєї туші. Норма закладки брутто на готовий виріб вагою 56 г складає 97 г.

Вага порції в грамах			
Кількість порцій			

3. Дати визначення алгоритму.
4. Яка властивість алгоритму ілюструє наступний нижче приклад?

Приклад. – Один бізнесмен придбав комп'ютер і підключився до інформаційної мережі. У процесі роботи потрібна була довідка про клієнтів його фірми, що проживають у Донецькій і Дніпропетровській областях. З клавіатури в ПК був уведений запит «Клієнти проживаючі в Донецькій і Дніпропетровській областях?». Після хвилинного міркування машина видала відповідь «Не значаться», хоча бізнесмен точно знав, що в цих областях клієнти в нього є. У чому помилка?

5. Пояснить властивість алгоритму «масовість».
6. Дайте математичне формулювання задачі:

Необхідно розрахувати вартість реалізованих товарів покупцю, якщо відомі його покупна ціна, кількість і сума надбавки. Причому сума торгової надбавки

залежить від приналежності товару до визначеної товарної групи: для одягу це 20%, для взуття – 15%, для канцтоварів – 10%, для інших товарів – 8%.

7. Запишіть алгоритм рішення задач із п.2 і 6 у блок-схемному вигляді.

8. Знайдіть робочу формулу для рішення задачі:

Визначити суму витрат на електроенергію для виробничих потреб за рік, якщо відомі:

1) кількість енергоустановок на підприємстві –  $n$ ;

1) масив потужностей кожної енергоустановки –  $M_i$  кВт/година;

2) масив часу роботи  $t_i$  кожної енергоустановки (у годину);

3) тариф за 1 кВт/годину електроенергії –  $R_i$ .

9. Представити графічно алгоритм рішення наступних задач:

$$a) \quad y = \begin{cases} 0 & \text{при } x = 0, \\ e^x + 13,29 * x & \text{при } x > 0, \\ a * (x - 8,3)^2 & \text{при } x < 0 \end{cases}$$

$$б) \quad S = \begin{cases} a + b * x & \text{при } x \leq 9,3 \\ a * x^2 + b + a * b * x & \text{при } 9,3 < x \leq 12 \\ k & \text{при } x > 12 \end{cases}$$

$$в) \quad y = \begin{cases} 5 * a + 5 - 4 * x & \text{при } a = 1, \\ 0,5 * e^x - (x - 1)^2 & \text{при } a = 2, \\ x^4 - 7 * x^2 + 12 & \text{в інших випадках.} \end{cases}$$

г) Визначити рентабельність  $R$  у цілому по фірмі, якщо відомий прибуток  $P_i$  і товарообіг  $T_i$  по кожній торговій точці ( $i = 1, 2, 3, \dots, n$ ):

$$R = \frac{\sum_{i=1}^n P_i}{\sum_{i=1}^n T_i} * 100$$

10. Для чого використовуються алгоритмічні мови? Які мови Вам відомі?

11. До яких рівнів відносяться сучасні алгоритмічні мови? Приведіть приклади.

## Література

1. Войтюшенко Н.Н., Остапец А.И. Компьютерная техника и программирование. Часть 2. Основы Алгоритмизации и программирования на Visual Basic: Учебное пособие по базовой подготовке для студентов дневной и заочной форм обучения. – Донецк: ДонГУЭТ, 2002. – 150 с.
2. Інформатика: Комп'ютерна техніка. Комп'ютерні технології. Посібник. – К.: Видавничий центр “Академія”, 2001 – 696 с.
3. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів/За ред. О.І.Пушкаря.-К.: Видавничий центр “Академія”, 2003. – 704 с.
4. Основы современных компьютерных технологий: Учебное пособие/Под ред. Проф.. Хомоненко А.Д. – СПб.: КОРОНА принт, 1998 – 448 с.



## Розділ 8. Загальні відомості й основні поняття мови програмування Visual Basic

### 8.1 Користувальницька оболонка пакета

З появою операційної системи **Windows** знадобилася алгоритмічна мова, що враховує особливості цієї системи і досить проста й звична для освоєння. Такою мовою став **Visual Basic**, що базується на основних конструкціях і особливостях звичайного Бейсіка і враховує властивості й форми представлення в середовищі **Windows**. **Visual Basic** є основою для створення макросів і програмних зв'язків у програмах пакета Microsoft Office – мови **Visual Basic for Application (VBA)**.

Завантаження Visual Basic (VB):

Кнопка **Пуск** → **Програми** → **Microsoft Visual Basic** → **Visual Basic** → у діалоговому вікні **New Project** вкладка **New** виділити **Standard.exe** → кнопка **Открыть**.

Після завантаження програми на екрані може бути присутнім ряд вікон інтегрованого середовища розробки (рисунок 8.1):

- 1) **вікно проектування «Visual Basic [design]»**, що містить заголовний рядок вікна і звичайні для неї кнопки системного меню, кнопки згорнути в значок, розгорнути/відновити, закрити; рядок меню (перелік і призначення команд приведені в додатку А) і рядок інструментальних кнопок;
- 2) **вікно конструктора форм «Form1»- екранна форма** для створення власного прикладного вікна користувача. У ньому присутні обов'язкові для усіх вікон Windows назва вікна й кнопки системного меню.
- 3) **вікно інструментів пакета VB**, що складається з двох стовпчиків кнопок інструментів;
- 4) **вікно провідника проекту** із заголовком **«Project1»**, у якому перелічуються усі компоненти створюваного додатка – екранні форми і програмні модулі;
- 5) **вікно властивостей об'єктів «Properties»**, що будуть розглянуті далі.

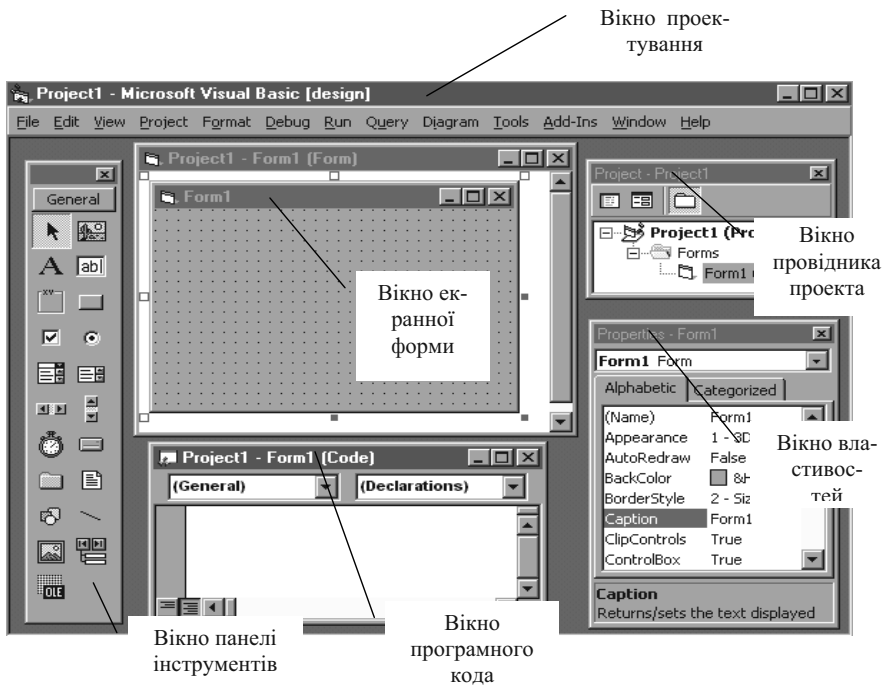


Рисунок 8.1 – Вікно проектування Visial Basic

б) **вікно програмного коду «Code»**. Призначено для створення в ньому програмних модулів (тобто самих програм, що виконують розрахунки й інші перетворення й дії).

Усі вікна мають кнопку системного меню і можуть бути закриті. Якщо на екрані немає потрібного Вам вікна, його можна відкрити за допомогою команд пункту меню **View**:

- **Object** – вікно екрана «Form1»;
- **Toolbox** – вікно інструментів пакета VB;
- **Properties Window** – вікно властивостей об'єктів «Properties»;
- **Code** – вікно програмного коду «Code»;
- **Project Explorer** – вікно провідника проекту.

У пакеті є гарна довідкова система мовою пакета. Виклик її йде через клавішу [F1] або пункт меню **Help**.

Крім зазначених, у вікні проектування можна викликати:

- вікно редактора меню (**Menu Editor**) – для створення й редагування рядка меню для форми,

- вікно макета форми ( **Form Layout** ) – для перегляду форми на екрані (аналогічно режиму «Попередній перегляд» у Word),
- вікно перегляду об'єктів ( **Object Browser** ) – для перегляду всіх елементів проекту,
- вікно Локальні ( **Locals** ) – для перегляду значень локальних перемінних,
- вікно Спостереження ( **Watches** ),
- вікно Безпосереднє виконання ( **Immediate** ) – для ручного введення й виконання команд.

**Вихід із пакета** можна виконати через системне меню вікна проектування (Microsoft Visual Basic [design]) або через:

пункт меню **File** → команда **Exit**.

## 8.2 Основні принципи розробки програм (проектів) у середовищі Visual Basic

Алгоритмічна мова Visual Basic (**VB**) – мова об'єктно – орієнтованого проектування. Це значить, що в ньому можна маніпулювати готувими об'єктами й методами їхньої обробки на рівні операторів алгоритмічної мови.

В об'єктно – орієнтованій мові використовуються такі поняття як:

- ◆ **об'єкти,**
- ◆ **властивості,**
- ◆ **події,**
- ◆ **методи.**

**Об'єкти** – пристрої або загальні елементи Windows-додатків, використовувани більшістю створюваних програм. При об'єктно- орієнтованому програмуванні практично усі компоненти комп'ютерної системи є об'єктами. У Visual Basic виділяються такі групи об'єктів:

- **глобальні (global objects):** clipboard (буфер обміну), debug (відладник), printer (принтер), screen (екран), app (додатки);
- **інтерфейсні або об'єкти взаємодії:** form (екранна форма), controls (керуючі елементи, наприклад ті, котрі присутні на панелі інструментів);
- **об'єкти бази даних.**

У першу чергу будуть використовуватися об'єкти вікна панелі інструментів (рисунок 8.2).










Показчик  
 Напис (мітка)  
 Рамка  
 Прапорець  
 Комбінований список  
 Горизонтальна лінійка про-  
 кручування  
 Таймер  
 Вікно папок  
 Фігури  
 Кадр  
 Об'єкт OLE



Картинка  
 Текстове вікно  
 Командна кнопка  
 Селекторна кнопка  
 Список  
 Вертикальна лінійка  
 прокручування  
 Кнопка устрою  
 Вікно файлів  
 Лінія  
 Вікно даних

Рисунок 8.2 – Панель інструментів Visual Basic

### Призначення деяких елементів панелі інструментів:

-  – етикетка або поле напису (**Label**). Призначена для створення підпису (напису) різних об'єктів, що дозволяє користувачеві визначити, що міститься або повинне міститися в полі, а також для виводу різних повідомлень.
-  - текстове вікно, використовуване для введення-виводу даних. (**TextBox**)
-  - командна кнопка. (**CommandButton**). Призначена для виклику необхідної процедури обробки.
-  - горизонтальна лінійка прокручування (**HScrollBar**)
-  - малюнок (**PictureBox**). Призначений для створення поля, у яке може бути поміщена ілюстрація (малюнок, картинка, формула і т.д.)
-  - група (**Frame**)
-  - перемикач (**OptionButton**)
-  - Прапорець (**CheckBox**)
-  - Об'єкт **OLE**. Призначений для розміщення у формі формул, як ілюстрації, і інших об'єктів .

Щоб ідентифікувати будь-який значок елемента на панелі інструментів Visual Basic, можна установити на нього покажчик миші, при цьому з'явиться спливаюча підказка з назвою цього елемента.

При складанні своєї програми кожен користувач визначається, з якими конкретно об'єктами він буде працювати – із якими формами вікон, із якими пристроями (результат виводити на екран, на друк), які керуючі елементи повинні містити його вікна.

Усі вимоги до обраних об'єктів користувач фіксує у виді **властивостей** у вікні **Properties**. Кожному об'єктові відповідає свій заздалегідь заданий набір властивостей.

**Властивість об'єкта** – це якість або характеристика об'єкта, вона визначає, як об'єкт виглядає або як він поводить. Наприклад, властивостями об'єктів є їхні імена, напис об'єкта, колір фону, символів. Список властивостей для кожного об'єкта визначений, а користувач може встановлювати конкретні значення потрібних властивостей: колір – блакитний, тип шрифту – Times New Roman, конкретний текст у напису і т.д.

Для установки значення властивості конкретного об'єкта необхідно клацнути по цьому об'єкту, тобто зробити його активним, і вивести клацанням на передній план вікно **Properties** (малюнок 8.3). У вікні відображається список властивостей активного в даний момент об'єкта й значення їх, привласнені за замовчуванням.

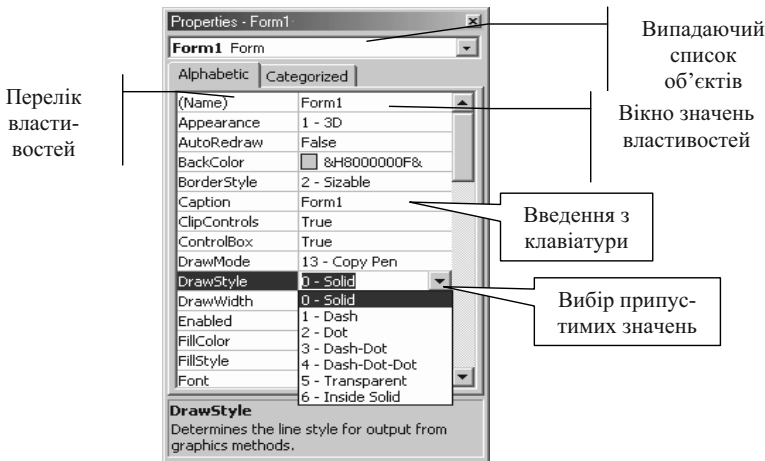


Рисунок 8.3 – Вікно властивостей Properties

Під рядком заголовка вікна розташований рядок (поле) з списком, що випадає, який утримує перелік усіх об'єктів створюваного проекту. У цьому рядку повинен бути зазначений (обраний) цікавлячий Вас об'єкт (його ім'я).

Виклик необхідної властивості для установки здійснюється по його назві в *першому стовпчику* вікна **Properties**. Щоб змінити значення властивості, потрібно клацнути в *другому стовпчику* вікна на кнопці праворуч наприкінці рядка властивості → у списку з припустимими значеннями, що з'явиться, вибрати клацанням необхідне або активізувати рядок і ввести дані за допомогою клавіатури.

Приклади деяких властивостей:

**Name** – встановлює ідентифікатор (ім'я) доступу до об'єкта (текстовому полю, полю написів, полю меню, що керують кнопкам і т.д.) з програми; властивість недоступна при виконанні програми. Наприклад, Name – для форми встановлює ім'я форми, використовуване в програмі. Ім'я починається з букви, складається з букв, цифр і підкреслення, має не більш 40 символів. Спочатку задається ім'я елемента, а потім уводиться для нього код програми.

**Alignment** – вирівнювання тексту (**0** – уліво, **1** – вправо, **2** – центрування);

**AutoSize** – автоматичне керування розміром поля напису або текстового поля (**true** – автоматична зміна відповідно до розміру вмісту, **false** – розміри незмінні й зайвий уміст відтинається);

**BackColor, ForeColor** – установка кольору фону і переднього плану об'єкта;

**BorderStyle** – задає тип рамки для об'єкта, причому для *form* і *textbox* тільки для читання під час виконання. Припустимі значення:

**0** – немає контуру,

**1** – фіксований одиночний верхній контур (рядка назви і меню),

**2** – за замовчуванням змінюваний контур,

**3** – фіксований подвійний.

**Caption** – текст, відображуваний у заголовку (для форми), усередині або поруч з елементом управління.

**FontName** – встановлює або повертає шрифт, використовуваний для відображення тексту в елементах управління, а також при виконанні операцій малювання й друку. Рекомендується використовувати шрифт **Times New Roman Cyr**.

**Height, Width** – зовнішня висота й ширина об'єкта. Задаються у твіпах (1 см = 567 твіпів).

**Text** – текст у текстовому або комбінованому полі, у полі списку. Для останнього – тільки для читання під час виконання.

Visual Basic є мовою, орієнтованою на обробку подій. Це означає, що окремі частини програми виконуються у відповідь на певні **події**. Кожен об'єкт повинен реагувати на певні події (дії), що відбуваються:

- на екрані (клацання миші по управляючих елементах),
- на клавіатурі (натискання різних клавіш),
- у програмі під впливом конкретних даних.

Тобто програма повинна мати архітектуру, керовану подіями.

Приклади подій:

**Change** – відбувається при зміні стану елемента управління при наявності цілого ряду можливих подій (масиву подій).

**Click** – відбувається при одноразовому клацанні миші над об'єктом.

**DbClick** – відбувається при подвійному клацанні миші на об'єкті.

**KeyPress** – відбувається при натисканні клавіші на клавіатурі.

**Load** – відбувається при виклику додатка за допомогою оператора *Load* у програмі або в результаті неявного завантаження.

**Unload** – відбувається при вивантаженні форми з оперативної пам'яті відповідно до дій користувача (за допомогою меню або кнопок) або оператором *Unload* із програми.

З кожною подією в Visual Basic зв'язана **процедура (підпрограма)**, що виконується при здійсненні події. У загальному виді синтаксис процедури наступний:

**Sub** <ім'я об'єкта>\_<ім'я події> (оголошення параметрів)

.....

**оператори**

.....

**End Sub**

Для кожного об'єкта зафіксований перелік можливих процедур, задача користувача – вибрати потрібну і написати відповідний код програми. Виконується це у вікні коду (рисунок 8.4).

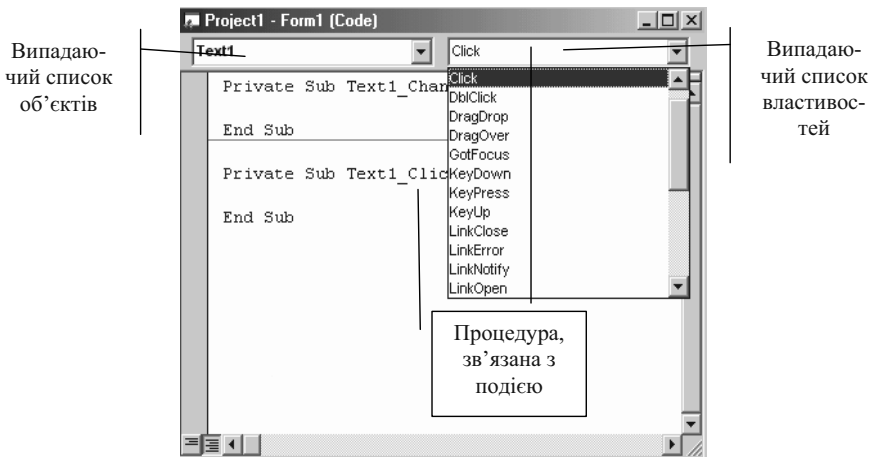


Рисунок 8.4 – Вікно програмного коду

При роботі з об'єктами виникає потреба в цілому ряді стандартних операцій, зв'язаних із відповідними вікнами, кнопками, полями. Visual Basic звільняє користувача від програмування таких операцій шляхом надання **набору методів**.

Наприклад, можна використовувати такі методи:

- Clear** – очищає вміст буфера обміну (Clipboard);
- Cls** – очищає форму або поле ілюстрацій;
- EndDoc** – припиняє посилку документа на принтер;
- GetData** – повертає зображення з Clipboard;
- GetText** – повертає текстовий рядок із буфера обміну;
- Hide** – ховає форму, але не вивантажує її з оперативної пам'яті;
- Move** – переміщає форму або об'єкт;
- Print** – друкує текстовий рядок на об'єкті;
- SetData** – поміщає ілюстрацію в заданому форматі в Clipboard;
- SetText** – поміщає текстовий рядок у буфер обміну;
- Show** – відображає форму на екрані.

Синтаксис запису методу:

**<ім'я об'єкта>.<ім'я методу> [параметри]**



Приклади: 1) Form1.Hide; 2) Form1.Show; 3) Printer.Print "y= ";y.Text

Об'єкти, що мають загальні властивості і поведінку (події й методи), поєднуються в *класи*. Ім'я класу вказує на тип об'єктів, що він містить. Класи можна розглядати як основу для створення інших об'єктів цього ж типу. Тому всі об'єкти одного класу, наприклад **TextBox**, будуть діяти однаково. За замовчуванням екземпляри того самого класу мають ім'я, що складається з імені батьківського класу і порядкового номера – **Text1**, **Text2**, **Text3**. При необхідності користувач може створювати свої класи об'єктів.

У Visual Basic також широко використовується поняття колекції об'єктів. *Колекція* – це набір об'єктів, об'єднаних загальним ім'ям, причому не обов'язково це об'єкти одного класу. Прикладом колекцій можуть служити убудовані в Visual Basic 6 колекції, у т.ч. **Forms** і **Controls**. **Forms** – це безліч завантажених форм додатка, **Controls** – це безліч усіх елементів управління у формі. Користувач може створювати свої власні колекції, поповнювати й актуалізувати їх.

**Розробка нової програми** або, по термінології VB, проекту складається з двох етапів:

- 1) **Етап візуального програмування** – *розробка форми або набору форм*, що будуть видаватися користувачеві під час роботи з програмою. Узгодження переліку й змісту форм із замовником може значно заощадити кошти на розробку проекту.
- 2) **Етап програмування у вхідному коді** – *розробка власне програми обробки інформації*, що відповідає різним діям користувача.

### 8.3 Реалізація першого етапу проектування

На етапі візуального програмування необхідно визначити:

1. Скільки екранних форм буде в задачі, для чого вони будуть використовуватися, що на цих формах буде розташовано (текстові поля, кнопки, лінійки прокручування, прапорці і т.д.).
2. Створити розроблені форми.

3. Визначити властивості об'єктів (самих форм і розташованих на них елементів управління) – зовнішній вигляд і особливості їхньої роботи в програмі.

Потрібні елементи вікна малюються на екранній формі за допомогою відповідних кнопок панелі інструментів. Для цього потрібно клацнути на кнопці з необхідним інструментом, далі при натиснутій лівій кнопці миші визначити розмір і місце розташування елемента у формі. Розміри самого вікна і розташованих на ньому елементів, коли вони активні (виділені), змінюються аналогічно всім мальованим об'єктам. Щоб зробити об'єкт активним (виділити), необхідно клацнути на ньому мишкою. При цьому довкола нього з'явиться рамка виділення, що містить вісім маленьких квадратиків. При розташуванні покажчика миші усередині рамки виділення можна переміщати об'єкт, а на одному з квадратиків рамки – змінювати його розміри.

*Для видалення* непотрібного об'єкта необхідно зробити його активним, а потім натиснути клавішу **Delete**

Приклад 8.1. – Спроекувати форму для виконання розрахунків по формулі:

$$y = 200 + a * c$$

Виконання: На екранній формі повинні розташовуватися текстові вікна для сприйняття значень **a** і **c**, а також вікно для виводу результату **c**. Щоб знати, де яке вікно знаходиться, їх необхідно підписати, створивши до них поля написів (етикетки, мітки). Розрахунок повинен виконуватися при клацанні по кнопці **Розрахунок**, а по кнопці **Вихід** – здійснюється завершення роботи програми.

1. Намалюйте на папері або уявіть собі думкою вид такого вікна (рисунок 8.5), після цього можна приступити до створення екранної форми на комп'ютері.

2. Завантажте Visual Basic.

3. У відкритому вікні Visual Basic з'явиться порожня екранна форма **Form1**. Якщо її розміри не задовольняють користувача, то варто клацнути по ній (виділити) і змінити розмір, як у звичайному малюнку.

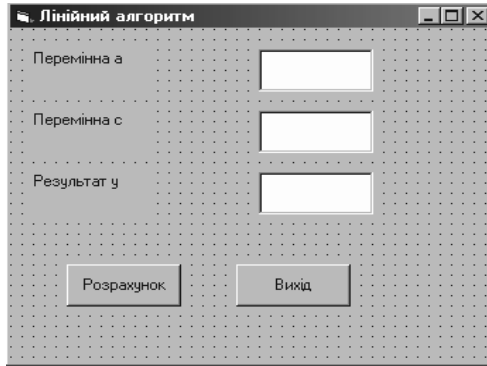





Рисунок 8.5 – Вид екранної форми для приклада:  $y = 200 + a * c$

Усі наступні дії по розробці форми програми представлені в таблиці 8.1.

Таблиця 8.1. – Послідовність дій по створенню форми для приклада  $y = 200 + a * c$

№ п/п	Операція	Виконання
1	2	3
1	Дати ім'я електронній формі.	1). Відкрити вікно <b>Properties</b> , якщо воно відсутнє на екрані: пункт меню <b>View</b> → <b>Properties Windows</b> ; 2). Вибрати властивість <b>Caption</b> , у правому стовпчику стерти <b>Form1</b> і написати <b>Лінійний алгоритм</b> .
2.	Занести напису вихідних даних і результату.	1). На панелі інструментів виконати подвійне клацання по кнопці  <b>Label</b> – напис), у центрі форми з'явиться прямокутник з написом <b>Label1</b> . Перетягніть його на відведене для нього місце в створюваній формі (рисунок 8.5). Ще двічі повторіть цю послідовність дій, одержавши <b>Label2</b> , <b>Label3</b> . 2). Активізувати клацанням миші вікно <b>Properties</b> ; у ньому у верхньому списку, що випадає, вибрати <b>Label1</b> ; у списку властивостей виділити <b>Caption</b> , у другому стовпчику стерти <b>Label1</b> написати: <b>Перемінна a</b> . 3). У верхньому списку вікна <b>Properties</b> , що випадає, вибрати <b>Label2</b> і у властивості <b>Caption</b> написати: <b>Перемінна c</b> .

Продовження таблиці 8.1.

1	2	3
3.	Створення текстових вікон для введення і виводу даних.	<p>4). У верхньому списку вікна <b>Properties</b>, що випадає, вибрати <b>Label3</b> і у властивості <b>Caption</b> написати: <b>Результат у</b>.</p> <p>1). На панелі інструментів виконати подвійне клацання по кнопці  (<b>TextBox</b> – текстове вікно) для створення текстового вікна <b>Text1</b>. Перетягніть його на відведене для нього місце в створюваній формі (малюнок 8.5). Ще двічі повторіть цю послідовність дій, одержавши <b>Text2</b> і <b>Text3</b>.</p> <p>2). Активізувати клацанням миші вікно <b>Properties</b>; у ньому у верхньому списку, що випадає, вибрати <b>Text1</b>; у списку властивостей виділити <b>Name</b>, стерти <b>Text1</b> і написати <b>a</b> (<u>англійською мовою</u>).</p> <p>3). Потім у списку властивостей вибрати властивість <b>Text</b> і стерти в іншому стовпчику <b>Text1</b>. (У полі повинно бути порожньо).</p> <p>Аналогічним образом дати елементу <b>Text2</b> ім'я <b>z</b>, а <b>Text3</b> – <b>y</b> (<u>англійською мовою</u>)* і у властивості <b>Text</b> стерти <b>Text2</b> і <b>Text3</b>.</p>
4.	Створення екранних кнопок	<p>1). На панелі інструментів виконати подвійне клацання по кнопці  (<b>CommandButton</b> – командна кнопка) і перетягнути отриману кнопку на відведене для неї місце у формі, повторити зазначені дії для одержання другої кнопки.</p> <p>2). Активізувати клацанням миші вікно <b>Properties</b>; у ньому у верхньому списку, що випадає, вибрати <b>CommandButton1</b>; у списку властивостей виділити <b>Caption</b>, стерти <b>CommandButton1</b> і написати <b>Розрахунок</b>; потім виділити <b>Name</b> і замість <b>CommandButton1</b> написати <b>Rachet</b>.</p> <p>3). У верхньому списку вікна <b>Properties</b>, що випадає, вибрати <b>CommandButton2</b>; у <b>Caption</b> написати <b>Вихід</b>, а в <b>Name</b> написати <b>Vixod</b>.</p>

\* – імена об'єктів можна задавати на інших мовах, що встановлені на комп'ютері, але далі при введення програми на англійській буде не дуже зручно переключатися з однієї мови на іншу. Тому назви на інших мовах (не англійській) зручно давати об'єктам, до яких не

буде звертання в тексті підпрограм. Такі приклади будуть приведені далі.

У результаті виконаних дій створена форма потрібного виду, яку необхідно зберегти.

**Збереження проекту складається з двох кроків.**

- **На першому кроці** виконується **збереження форми** – файл із розширенням **.frm**.
- **На другому кроці** виконується власне **збереження проекту** з розширенням **.vbp**.

**Схема збереження проекту:**

Пункт меню **File** → команда **Save Project** → дати ім'я формі (**Лінійний алгоритм**) і вказати потрібну папку → **ОК** → дати ім'я проекту (**Лінійний алгоритм**) і вказати папку **ОК**.

## 8.4 Реалізація другого етапу проектування

При роботі користувача з формою після уведення вхідних даних клацання мишкою по кнопці **Розрахунок** повинне забезпечити одержання результату. Така реакція на подію «клацання» (Click) забезпечується на другому етапі проектування – *програмування подійних процедур*.

Програмування подійної процедури виконується у вікні програми (**Code Window**), що викликається подвійним клацанням миші по потрібному об'єкту. При цьому використовуються як методи, так і звичні для алгоритмічних мов команди. Тому коротко розглянемо основні елементи мови Visual Basic.

### 8.4.1 Елементи мови

Алфавіт мови складається з латинських букв, цифр, службових символів.

Наприклад, *одиначні лапки* позначають *початок коментарю*, що при виконанні програми ігнорується й несе тільки довідкову інформацію для читання програми; знак підкреслення наприкінці рядка оператора означає наявність його продовження на наступному рядку.

У програмах використовуються:

- **арифметичні операції** у порядку пріоритетів їхнього виконання – ^, \*, /, \ – ділення з ігноруванням дробової частини результату, +, – ;
- **операції відносин** – <, >, =, >=, <=, <> ;
- **логічні операції** – **And, Or, Not, Xor** і ін.

**Строкові вираження** – текст ув'язнений у подвійні лапки. Для строкових виражень використовується операція конкатенації (+), що дозволяє поєднувати останні і будувати з них нові фрази.

Наприклад:

“Сьогодні” + “гарна” + “погода” = Сьогодні гарна погода.

Деякі типи даних Visual-Basic приведені в таблиці 8.2.

Таблиця 8.2. – Типи даних

№ n/n	Назва	По-знач.	Характеристика типу даних
1	Integer	%	цілі числа від -32768 до +32768, 2 байти
2	Single precision	!	число з плаваючою крапкою одинарної точності, 4 байти (Використовується за замовчуванням)
3	Long Integer	&	довге ціле, 4 байти
4	Double precision	#	число з плаваючою крапкою подвійної точності, 8 байтів
5	String	\$	рядок символів до 65535 сим. або оголошеної довжини
6	Currency	@	число з фіксованою крапкою (15 зн. ліворуч і 4 праворуч від крапки, 8 байт)
7	Variant		дані, для яких тип не відомий. Займають більше місця, чим явно оголошені перемінні.

Усі дані, що використовує програма, можуть бути **постійними** (**константами**) або **перемінними**. Константи і перемінні оголошуються на

початку будь-якої підпрограми або модуля:

- **константи** – за допомогою оператора **Const**,
- **перемінні** – за допомогою оператора **Dim**.

Формат оператора **Const** наступний:

[Global] Const <ім'я> = вираження [, <ім'я> = вираження]...<sup>1</sup>

Приклади: const Pi = 3.1415

const Max% = 250

const Clovo\$ = “відмінно”

Типи констант і виражень, що привласнюються їм, повинні збігатися. В арифметичних вираженнях заборонено використовувати операцію зведення в ступінь і деякі функції.

Формат оператора **Dim** наступний:

**Dim** <ім'я перемінної>[(**[**опис масиву перемінних**]**)] [**As**[**New** type]**]**, <ім'я перемінної>[(**[**опис масиву**]**)] [**As**[**New** type]**]**...

Де:

**As type** – визначає тип даних (Integer, Long,...) чи об'єкта.

Наприклад:

Dim a As Integer,

Dim Temper As Single,

Dim d, b, rakurc – за замовчуванням тип Variant,

Dim FIO As String – будь-яка довжина символної перемін-

ний,

Dim Street As String\*75 – явна вказівка довжини строкового типу (75 – кількість символів у назві вулиць).

**New** – використовується для створення нових характеристик спеціальних типів об'єктів, таких як Form1.

Опис масиву перемінних включає вказівку їхньої нижньої й верхньої границь:

Dim A(8,3);

Dim A(0 To 8, 0 To 3)

Dim A(8,0 To 3)

Dim A(-4 To 10)

Максимальне число масивів у даному операторі – 60.

Секція оголошень повинна бути на початку кожної процедури й процедури – функції. Оголошення перемінних і масивів у подійної процедури дійсно тільки в цій процедурі.

---

<sup>1</sup> У записах формату операторів будуть використані наступні позначення: {} – вибір одного з перерахованих значень, [] – елемент може бути або не бути, <> – треба написати конкретне значення.

Для оголошення *глобальних* перемінних використовується глобальний модуль (**General**), що автоматично включається до складу будь-якого проекту і розміщається перед першою процедурою.

Наприклад:

General

Option Explicit

Dim X As Integer

Dim I, Y

Якщо тип даних для перемінної не визначений, то за замовчуванням вона одержує тип **Variant**. Це значить, що тип перемінної визначається в залежності від типу даних, що вперше будуть у неї поміщені.

Найпростішим оператором мови є *оператор присвоювання*.

Формат оператора:

**<перемінна> = вираження**

Наприклад:

Y = 0

Z = (x + 7) / 2

Значення вираження правої частини при-  
власнюється перемінній лівої частини

#### 8.4.2 Уведення – вивід даних

**Уведення даних** здійснюється через керуючі елементи форми – текстові поля, табличні поля (масиви), комбіновані списки (файли). Уведення здійснюється набором на клавіатурі потрібних символів і закінчуються натисканням клавіші **Enter**; клацанням миші по потрібному імені файлу, видам характеристик, керуючий кнопці **OK** і т.п. При цьому названим подіям можуть відповідати підпрограми (процедури), що виконують рішення задачі.

**Вивід даних** можна здійснювати на екран, на друк, у файл. Вивід на екран виконується, наприклад, присвоєнням визначених значень текстовим полям, табличним полям.

Для виводу на друк необхідно використовувати метод **Print**.

Синтаксис:

**[об'єкт]Print [{Spc(n),Tab(n)}][вираження][;I,}] ,**

де:

**об'єкт** – це в даному випадку принтер (Printer);



**Spс(n)** і **Tab(n)** – функції вставки **n** пробілів і табуляції у визначений стовпчик друкуючої голівки принтера;

**вираження** – те, що видається на друк;

**;** **I**, – немає пропуску між виведеними на друк даними або табуляція через 14 колонок.

Наприклад:

```
Printer.Print "y = ";y.Text
```

Для виводу даних у визначеній формі необхідно використовувати фор-матований вивід – функцію **Format**.

Синтаксис:

**Format[\$] (expression [,fmt]),**

де:

**\$** – ознака строкового типу,

**expression** – арифметичне або строкове вираження,

**frm** – задається формат виведеного вираження:

**0** – задає друк цифри або нуля, наприклад, 000000 – буде надруковане число не більш 6 знаків, якщо воно менше 6 знаків, то друкуються головні нулі.

**#** – задає друк цифри або нічого.

**.** – задає друк місця десяткової крапки, наприклад, #####.##.

**%** – виводить число в процентному виді, тобто вираження збільшується на 100%.

**,** – множить вираження на 1000.

**E** – науковий формат у виді числа з плаваючою комою.

**\$** – строковий формат “ ”.

**:** – формат часу.

**/** – формат дати й ін.

При введенні й виводі даних може виникнути необхідність перетворення з однієї форми представлення даних в іншу. Наприклад, введення й вивід у текстове поле здійснюється строковим типом, а обчислення необхідно виконувати з числовими даними.

**Перетворення строкових даних у числові** здійснюється за допомогою функції **Val**.

Синтаксис:

**Val (строкове вираження) – числове значення.**

Наприклад:

У текстове поле **a** введений рядок 5 (символ 5 сприймається як текст). Для участі в обчисленнях уміст поля **a** необхідно перетворити в число.

$$\text{Val}(a.\text{Text}) = 5$$

**Перетворення числових даних у строкові** здійснюється за допомогою функції **Str**.

Синтаксис:

**Str (числове значення) – строкове вираження**

Наприклад:  $y.\text{Text} = \text{Str}(S) + \text{“грн”}$

Приклад 8.2 – Для форми, отриманої за умовою приклада 8.1, створити програмний код.

Виконання:

1). Відкрити проект **Лінійний алгоритм**:

Пункт меню **File** → команда **Open Project** → указати папку й ім'я файлу → кнопка **Відкрити**.

На екрані повинна з'явитися форма. У випадку відсутності форми після відкриття проекту варто виконати:

меню **View** → команда **Project Explorer** → у вікні, що відкрилося, розкрити **папку Project1 (Лінійний алгоритм)** ( **папку Forms** ( двічі клацнути **по Form1**).

2). Двічі клацнути по кнопці **Розрахунок** і з'явиться вікно програмного коду (**Code**), у якому:

- **у лівому списку, що** випадає, (**Object**) – міститься перелік усіх елементів проекту,
- **у правому списку, що** випадає, (**Procedure**) – відповідні їм процедури.

Вибрати в лівому списку елемент **Rachet**, а в правому – процедуру **Click**. Відкриється процедура, що містить початок і кінець, необхідно дописати її зміст – оператор обчислення **Y** (рисунок 8.6).

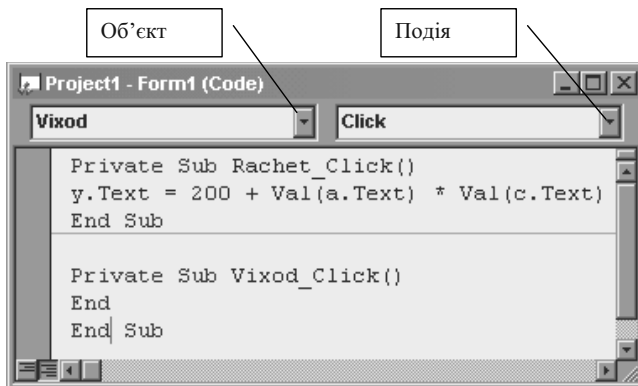


Рисунок 8.6 – Вікно програмного коду проекту – “Лінійний алгоритм”

Оскільки вхідні дані й результати вводяться через текстові вікна, а в програмі йде обчислення з числами, використовується функція перетворення текстових даних у числові – **Val**.

3). У лівому списку, що випадає, викликати елемент **Vixod**, а у правому – **Click**. Увести оператор закінчення роботи програми – **End**.

4). Виконати написану програму:

меню **Run** → команда **Start** → ввести у вікно **Перемінна**, а число **5**, а у вікно **Перемінна c** – число **6** → клацнути по кнопці **Розрахунок** → у вікні **Результат** у повинно вийти число **230**.

У випадку допущених помилок видається повідомлення й у тексті програми помилковий оператор виділяється кольором. Варто виправити помилку і знову спробувати виконати програму:

меню **Run** → команда **Continue**

5) Завершення роботи з програмою виконується клацанням по кнопці **Вихід**.

### 8.4.3 Надання привабливості формі й створення виконавчого файлу

При створенні проектів автор повинен подумати про зовнішню привабливість видаваних у процесі рішення задачі електронних форм. Навіть таку просту форму, як на рисунку 8.5, можна представити в

більш естетичному оформленні – можна зробити фон й елементи іншим(и) кольором, показати формулу, на підставі якої виконується розрахунок, уставити картинку (рисунок 8.7). Вирівнювати розміри або вміст полів, офарблювати об'єкти можна групою, використовуючи групове виділення графічних об'єктів за допомогою клавіші Ctrl і клацання миші.

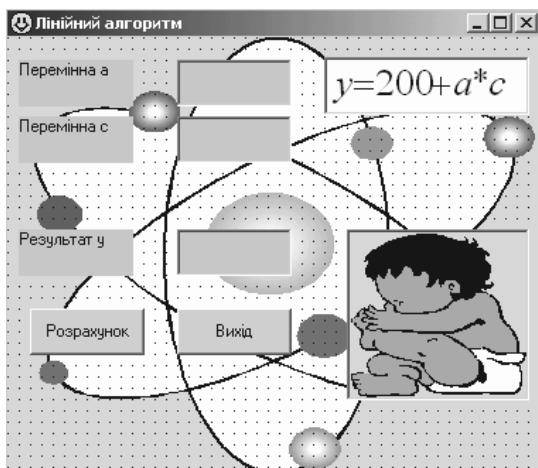


Рисунок 8.7 – Форма «Лінійний алгоритм» з об'єктами Picture, OLE і розфарбованими об'єктами.

Наприклад, для *зміни кольору фону форми або кожного з її елементів* необхідно:

- 1) виділити клацанням миші потрібний об'єкт;
- 2) у вікні **Properties** знайти властивість **BackColor** і за допомогою списку другого стовпчика, що випадає, (**вкладка Palette**) підібрати потрібний колір.

Усі написи і видавані результати можна *центрувати* у своїх вікнах. Для цього властивості **Alignment** кожного об'єкта варто привласнити зі списку другого стовпчика, що випадає: **2** – center.

Для того щоб *уставити* попередньо обрану чи намальовану *картинку у виді фону*, варто виділити форму, потім знайти й клацнути по властивості **Picture** і за допомогою кнопки наприкінці рядка викликати діалогове вікно **Load Picture**, у якому вказати папку (наприклад, **ClipArt**) і файл із картинкою. В екранну форму малюнок уставляється з верхнього лівого кута. Малюнок до форми не пристосовується і, якщо

він більше форми, то буде видно не весь. Збільшуючи форму, можна збільшити видиму частину малюнка.

**Забрати картинку** – очистити рядок властивостей **Picture**.

Щоб **установити картинку в окремо відведене місце** можна використовувати об'єкти **Picture** або **Image** (устанавлиються за допомогою відповідних кнопок панелі інструментів аналогічно раніше розглянутим об'єктам). В об'єкті **Picture** малюнок може редагуватися, а в **Image** – не може. Далі вибирається властивість **Picture** і виконується той же набір дій, що і для форми. Або можна скопіювати потрібний малюнок у буфер, а потім, клацнувши по об'єкту **Picture** або **Image**, виконати:

Пункт меню **Edit** → команда **Paste**.

Для вставок формул або об'єктів з інших додатків можна використовувати інструмент **OLE** (кнопка **OLE** на панелі інструментів). У діалоговому вікні **OLE** необхідно вибрати потрібну програму (об'єкт), наприклад, **Microsoft Equations**.

Після виконання всіх дій треба файл зберегти і можна **роздрукувати** всю зроблену роботу:

меню **File** → команда **Print** → вибрати прапорці **Form Image** (для друку виду електронної форми), **Code** (друк коду програми), **Form As Text** (друк властивостей усіх об'єктів проекту). Можна друкувати всі складові, а можна вибирати будь-яке потрібне сполучення.

Створений проект виконується тільки в середовищі Visual Basic. Щоб зробити його незалежним від середовища, варто створити виконавчий файл (із розширенням **exe**). **Exe-файл** може безпосередньо запускатися з Windows як додаток. Додатки Windows запускаються подвійним клацанням по відповідному значку. Якщо користувач хоче привласнити своїй програмі не стандартний значок файлу, то він повинен установити властивість **Icon** для усієї форми (кнопкою в другому стовпчику відкрити діалогове вікно й вказати папку й файл із потрібним значком). Наприклад, виберемо для проекту **Лінійний алгоритм** у папці **vb98\icons\misc** файл **face03**.

**Примітка** – У випадку відсутності папки **icons** можна скористатися меню **Найти** кнопки **Пуск** і пошукати файли з розширенням **.ico**.

Після цього: пункт меню **File** → команда **Make exe File**. У діалоговому вікні, що відкривається, користувач повинен вказати ім'я файлу, папку й **ОК**. **Exe-файл** буде створений.

#### 8.4.4 Використання лінійок прокручування

Для зручності користувача при виконанні операцій навігації за списком або визначення значення перемінної у програмах можна використовувати *лінійки прокручування*. Їх можна помістити у форму за допомогою елементів панелі інструментів:



- горизонтальна лінійка (**HScrollBar**);



- вертикальна лінійка (**VScrollBar**).

Переміщення бігунка на лінійці прокручування приводить до зміни властивості **Value**. Ця властивість визначає поточну позицію для елемента (конкретне його значення), що знаходиться в діапазоні, обумовленому властивостями **Max** і **Min** (цілі числа!).

З лінійками прокручування зв'язані *події*:

**Change** – настає в момент клацання по кнопках зі стрілками або між кнопкою зі стрілкою й бігунком, у момент відпускання бігунка після його переміщення;

**Scroll** – дозволяє одержати значення властивості **Value** при переміщенні бігунка до наступу події **Change**.

Крок переміщення при клацанні по кнопці зі стрілкою задається властивістю **SmallChange**, а при клацанні між кнопкою зі стрілкою й бігунком – властивістю **LargeChange**.

Приклад 8.3 – Спроекувати форму і створити програмний код для обчислення функції  $Y = 230 + a * c$ , якщо відомі діапазони зміни перемінних **a** і **c**:  $-1000 < a \leq 1000$  і  $-500 < c \leq 500$

Виконання:


1. В екранну форму, створену в прикладі 8.1, додати дві лінійки прокручування для вибору значень перемінних **a** і **c**. (рисунок – 8.8).

Послідовність дій по додаванню лінійок у форму й установці властивостей для них приведені в таблиці 8.3.



Рисунок 8.8 – Вид екранної форми для прикладу 8.3

Таблиця 8.3. – Послідовність дій по рішенню прикладу 8.3 – “ $Y = 230 + a * c$  із використанням лінійок прокручування”

№ п/п	Операція	Виконання
1	Відкрити форму проекту, створену в прикладі 8.1	Послідовність дій для відкриття форми див. у пункті 1 Прикладу 8.2.
2	Створення лінійки для вибору значень перемінної $a$ .	<p>1. На панелі інструментів вибрати елемент управління  <b>HScrollBar</b>, створити з його допомогою лінійку прокручування і розмістити її у формі вгорі як показано на рисунку 8.8.</p> <p>2. Активізувати вікно <b>Properties</b> і в ньому установити властивості для лінійки:</p> <p><b>Name</b> – <b>Linea</b>  <b>Max</b> – <b>1000</b>  <b>Min</b> – <b>-1000</b>  <b>Value</b> – <b>0</b>  <b>SmallChange</b> – <b>1</b>  <b>LargeChange</b> – <b>10</b></p>
3	Створення лінійки для вибору значень перемінної $c$ .	<p>1. На панелі інструментів вибрати елемент управління <b>HScrollBar</b>, створити з його допомогою лінійку прокручування і розмістити її у формі як показано на рисунку 8.8.</p> <p>2. Активізувати вікно <b>Properties</b> і в ньому установити властивості для лінійки:</p> <p><b>Name</b> – <b>Linec</b>  <b>Max</b> – <b>500</b>  <b>Min</b> – <b>-500</b>  <b>Value</b> – <b>0</b>  <b>SmallChange</b> – <b>1</b>  <b>LargeChange</b> – <b>5</b></p>

2. Внести доповнення в програмний код:

1). Подвійним клацанням лівої кнопки мишки по верхній (у формі проекту) лінійці прокручування відкрити вікно **Code**.

Для об'єкта **Linea** вибрати подію **Change** і дописати процедуру, виконувану, якщо уже відбудеться подія – переміщення бігунка на лінійці.

```
Private Sub LineA_Change()  
a.Text = LineA  
End Sub
```

2). Для об'єкта **LineA** вибрати подію **Scroll** і дописати зміст процедури, виконуваної під час переміщення бігунка по лінійці:

```
Private Sub LineA_Scroll()  
LineA_Change  
End Sub
```

3). Аналогічно пунктам 1 і 2 створити процедури для об'єкта **LineC**.

```
Private Sub LineC_Change()  
c.Text = LineC  
End Sub  
Private Sub Line_Scroll()  
Linec_Change  
End Sub
```

3. Виконати створену програму, вибираючи значення для перемінних **a** і **c** шляхом переміщення бігунків на лінійках прокручування.

4. Зберегти проект з ім'ям **Лінійний з прокручуванням**.

### **Питання й завдання для самостійного виконання**

1. Дайте характеристику об'єктно – орієнтованих мов на прикладі Visual Basic.
2. Що таке об'єкт у Visual Basic, привести приклади різних типів об'єктів.
3. Які характеристики властиві об'єктам і як вони задаються?
4. Чим відрізняються поняття «подія» і «метод» у Visual Basic, привести приклади.



5. Поняття «клас» і як воно співвідноситься з поняттям «об'єкт» ?
6. Які етапи проектування Ви знаєте?
7. Спроекувати додаток по обчисленню вартості товару (S), якщо відомі ціна (с) і кількість (k) купленого товару одного найменування.
8. Спроекувати додаток із розрахунку відсотка виконання плану (P), якщо відомі плановий (ПВ) і фактичний (ФВ) випуск продукції.
9. Визначити кількість порцій відбивних котлет вагою m, якщо відомо, що в їдальню надійшла туша яловичини першої категорії вагою P кг. Котлетне м'ясо складає 43% від усїєї туші. Норма закладки бруutto на готовий вирїб вагою 56 гр складає 97 гр.
10. Розрахувати дійсний фонд часу роботи устаткування по формулі:

$$D = \frac{S * D_1 * (100 - R) * P * T}{100 * N},$$

де:

N=12 – кількість місяців у розрахунковому періоді;

R – утрати часу на ремонт;

D1 – номінальний фонд часу при однозмінній роботі (321 година);

S – фактична змінність роботи устаткування;

T – час роботи устаткування;

P – продуктивність устаткування.

## 8.5 Реалізація розгалужених алгоритмів

У більшості програм виникає необхідність у виконанні операторів, керуючих ходом обчислювального процесу. До них відносяться: оператори *безумовного й умовного переходів, оператор вибору*.

- **Оператор безумовного переходу – GoTo**

Синтаксис оператора:

**GoTo <мітка>**

де **мітка** – це сполучення не більш 40 букв і цифр. Крім оператора, мітка ставиться на початку рядка, до якого потрібно перейти, і закінчується вона двокрапкою.

Наприклад: **Go To M1**

оператори програми

**M1:** оператор.....

- **Оператор умовного переходу:**

Синтаксис оператора:

1) **простий оператор:**



Приклад 7.2 (Розділ 7) можна записати за допомогою оператора:

**If**  $x \geq 0$  **Then**  $y = a * x + b$  **Else**  $y = a * x - b$ ,

де:  $y, a, x, b$  – імена перемінних.

Примітка – Якщо оператор не поміщається в рядку вікна, то його можна перенести на наступний рядок, використовуючи в місці розриву знак підкреслення \_.

2) **складний оператор:**

**If** умова1 **Then**  
                   [оператори]  
           **[Elseif** умова 2 **Then**  
                   [оператори]  
                   **[Else** [оператори]]  
**End if**

**Складний оператор** може включати вкладені умови. Він завжди записується за зазначеною структурою і закінчується зарезервованим словом **End if**.

Примітка – Якщо декілька операторів в одному рядку відносяться до частин **Then** або **Else**, то вони відокремлюються друг від друга дво-крапкою.

У якості оператора може використовуватися сам оператор **If** у простій або складній формі.

Наприклад (умова й блок-схема приклада 7.3):

$$y = \begin{cases} x / a - x / b & \text{при } a \leq 5 \text{ і } x > 6, \\ (a + b) / x & \text{при } a \leq 5 \text{ і } x = 6, \\ a^3 + b * e^{2,5} & \text{в інших випадках} \end{cases}$$

При обчисленні по першій формулі, крім зазначених умов, необхідно передбачити, щоб  $a \neq 0$ ,  $y \neq 0$ . Тоді запис складного оператора буде мати вигляд:

```
If Val(a.Text) <= 5 And Val(x.Text) = 6 Then
    y = (Val(a.Text) + Val(b.Text)) / Val(x.Text)
Elseif Val(a.Text) <= 5 And Val(x.Text) > 6 And Val(a.Text) <> 0 And
Val(b.Text) <> 0 Then
    y = Val(x.Text) / Val(a.Text) - Val(x.Text) / Val(b.Text)
Else
    y = Val(a.Text) ^ 3 + Val(b.Text) * exp(2.5)
End If
```

При записі зазначеного оператора передбачалося, що  $y$ ,  $a$ ,  $x$ ,  $b$  – імена текстових елементів (Textbox),  $e^{2.5}$  – експотенціальна функція.

- 2) Для виконання однієї з багатьох альтернативних дій зручно використовувати *оператор вибору*:

```
Select Case <перемінна>
    [Case порівняння 1
     [оператори]]
    [Case порівняння 2
     [оператори]]...
    [CaseElse
     [оператори]]
End select
```

Вибір варіанта визначається значенням керуючої перемінної, котре порівнюється з умовами, зазначеними після **Case**. Якщо всі порівняння невдалі, то виконуються оператори після **Case Else**. Умови в полє порівняння можуть мати досить різноманітний характер.

Наприклад:

```
Select Case alfa
    Case 1
        'оператори відсутні
    Case 2 To 4, 7 To 9,11,13, alfa > Max
        'значення alfa порівнюється з 2,3,4,7,8,9,11,13 і з Max.
        'Якщо значення перемінної співпаде з яким-небудь
```

числом або виконається порівняння, то відпрацюють два наступних оператора і потім **End Select**.

```

b.Text=alfa + MaxA
Printer.Print "b= ", b.Text
Case Else
alfa=0
End Select

```

### Приклади написання розгалужених програм

Приклад 8.4. – У створеному проєкті по обчисленню  $y = 200 + a * c$  (приклад 8.3) уставити командну кнопку по очищенню полів введення/виводу, передбачити можливість введення вхідних даних як за допомогою лінійок прокручування (якщо числа цілі і знаходяться в зазначених діапазонах), так і безпосередньо через текстові вікна; уставити оператори контролю введення даних користувачем (повинні вводитися тільки числа, в іншому випадку необхідно видати повідомлення про помилку).

Виконання: Контроль введення даних необхідно виконувати в текстових вікнах **a** і **c**. Користувач може набирати цифри, знаки + або – , використовувати клавіші **Delete** і **Backspace**, закінчувати введення клавішею **Enter**, всі інші символи повинні викликати повідомлення про помилку. При натисканні на зазначені клавіші на клавіатурі в комп'ютер надходить відповідний код, який можна перевіряти програмно. А тому що перевіряти потрібно досить велику кількість кодів, то краще використовувати оператор **Select case**. Перевірка введення повинна йти спочатку, а обчислення – при натисканні на клавішу **Enter**. Необхідні операції по прикладу приведені в таблиці 8.4


Таблиця 8.4. – Операції по доповненню проєкту – «Лінійний з прокручуванням»

№ п/п	Операція	Виконання
1	2	3
1	Відкриття проєкту	Пункт меню <b>Файл</b> → команда <b>Открыть</b> → указати папку й ім'я проєкту → кнопка <b>Открыть</b> .

Продовження таблиці 8.4.

1	2	3
2	Перейменування форми	Клацанням виділити форму й у властивості <b>Caption</b> написати: $y = 200 + a * c$ з контролем уведення. (рисунок 8.9)
3	Переміщення командних кнопок	Клацнути по кнопці <b>Розрахунок</b> (виділити об'єкт) і перетягнути її мишкою в правий нижній кут, аналогічно зробити з кнопкою <b>Вихід</b> .
4	Створення поля напису для повідомлення	Клацнути по кнопці <b>A</b> і на місці, що звільнилося, унизу форми намалювати вікно для повідомлення
5	Завдання властивостей поля для повідомлення	У вікні <b>Properties</b> у полі <b>Name</b> замість <b>Label4</b> написати <b>Mes</b> , поле <b>Caption</b> очистити
6	Додавання операторів контролю введення в код текстового вікна <b>a</b>	<p>Подвійним клацанням по текстовому вікну <b>a</b> викликати вікно коду й у ньому відкрити подію <b>KeyPress</b>. З'являється заготівка процедури:</p> <pre><b>Sub a_KeyPress (KeyAscii As Integer)</b> <b>End Sub</b></pre> <p>Оголошений автоматично параметр <b>KeyAscii</b> служить для прийняття коду натиснутої на клавіатурі клавіші.</p> <p>Додати в заготівку оператор по контролю введення таким чином:</p> <pre><b>Private Sub a_KeyPress(KeyAscii As Integer)</b> <b>Select Case KeyAscii</b> <b>Case 0, 8, 13, 45, 46, 48 To 57</b> '0, 8 – Delete, Backspace, 13 – Enter, '45 – «-», 46 – «.», 48 – 57 – цифри ' від 0 до 9 <b>Case Else</b> <b>KeyAscii = 0</b> <b>Mes.Caption = "Помилка введення!_</b> <b>Уведіть число"</b> <b>End Select</b> <b>End Sub</b></pre> <p><b>Примітка</b> – Код десяткової крапки на цифровій клавіатурі при роботі в російському алфавіті <b>44</b>, а в англійському – <b>46</b>.</p>

Продовження таблиці 8.4.

1	2	3
7	Додавання операторів контролю введення в код текстового вікна <b>c</b>	Для текстового вікна <b>c</b> усі дії виконуються аналогічно <b>АБО</b> у вікні коду <b>a</b> виділити рядки від <b>Select case</b> до <b>End select</b> і: Пункт меню <b>Edit</b> → команда <b>Copy</b> ; відкрити вікно коду текстового вікна <b>c</b> , установити подію <b>KeyPress</b> , а потім - пункт меню <b>Edit</b> → команда <b>Paste</b> .
8	Додавання повідомлення при введенні перемінної <b>a</b> за допомогою лінійки	<b>Private Sub Line_Change()</b> <b>a.Text = Line</b> <b>Mes.Caption = "Уведення перемінної a в діапазоні -1000 : 1000"</b> <b>End Sub</b>
9	Додавання повідомлення при введенні перемінної <b>c</b> за допомогою лінійки	<b>Private Sub LineC_Change()</b> <b>c.Text = Linec</b> <b>Mes.Caption = "Уведення перемінної c у діапазоні -500 : 500"</b> <b>End Sub</b>
10	Додавання командної кнопки <b>Очищення</b>	1) На панелі інструментів виконати подвійне клацання по кнопці  ( <b>CommandButton</b> ) і перетягнути отриману кнопку на відведене для неї місце у формі. 2) Активізувати клацанням миші вікно <b>Properties</b> ; у ньому у верхньому списку, що випадає, вибрати <b>CommandButton1</b> ; у списку властивостей виділити <b>Caption</b> , стерти <b>CommandButton1</b> і написати <b>Очищення</b> ; потім виділити <b>Name</b> і замість <b>CommandButton1</b> написати <b>Och</b> .
11	Занесення програмного коду для кнопки <b>Очищення</b>	Подвійним клацанням по кнопці <b>Очищення</b> викликати вікно <b>Code</b> , і записати наступні команди очищення всіх текстових вікон і вікна повідомлення для об'єкта <b>Och</b> події <b>Click</b> : <b>a.Text = " "</b> <b>c.Text = " "</b> <b>y.Text = " "</b> <b>mes.Caption = " "</b>
12	Перевірка працездатності проекту	Пункт меню <b>Run</b> → команда <b>Start</b> ; увести числа і прорахувати, а потім у текстових вікнах спробувати набрати букви.
13	Збереження зміненого проекту	Пункт меню <b>File</b> → команда <b>Save Project</b> → дати ім'я « <b>y = 200 + a*c</b> з контролем введення».

Примітка – Якщо дані форми будуть заповнюватися в російському алфавіті, то при контролі введення код десяткової крапки на цифровій клавіатурі – 44.

При необхідності, для зміненого проекту варто повторити процедуру створення .exe – файлу.

Для інших видів обчислювальних процесів основні методичні підходи залишаються такими ж.

Приклад 8.5 – Записати й налагодити програму простого розгалуженого процесу. Форма проекту представлена на рисунку 8.10.

$$y = \begin{cases} a * x & \text{при } x \geq 0 \\ a + x & \text{при } x < 0 \end{cases}$$

Операції по виконанню прикладу 8.5 «Просте розгалуження» показані в таблиці 8.5, алгоритм рішення приведений – у прикладі 7.2.

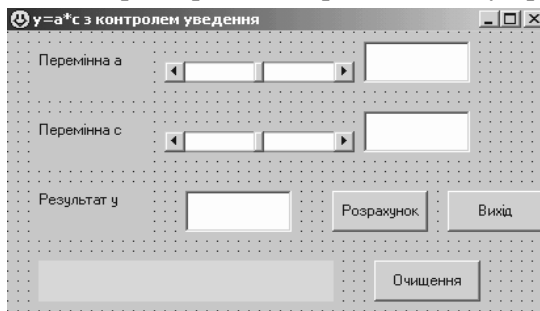


Рисунок 8.9 – Електронна форма проекту – « $y = 200 + a * c$  із контролем введення»

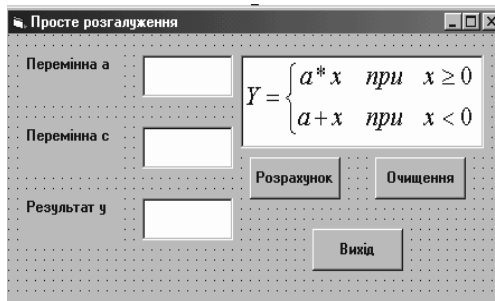


Рисунок 8.10 – Екранна форма прикладу 8.5 – «Просте розгалуження»

Таблиця 8.5. – Операції по виконанню приклада 8.5 – «Просте розгалуження»

№ п/п	Операція	Виконання
1	2	3
1	Створення електронної форми і завдання властивостей її елементів	Дії аналогічні попереднім прикладам. Властивість <b>Name</b> командної кнопки <b>Очищення</b> – <b>Och</b> . Імена текстових полів залишити стандартними – <b>Text1, Text2, Text3</b> .
2	Завдання видачі повідомлення при клацанні по полю введення <b>a</b>	Виконати подвійне клацання по текстовому полю <b>Text1</b> , установити подію <b>Click</b> , увести текст підпрограми: <b>Sub Text1_Click ()</b> ‘При клацанні по полю <b>Text1</b> буде видаватися повідомлення <b>Mes.Caption = «Уведення перемінної a»</b> <b>End Sub</b>
3	Контроль уведення даних у поле введення перемінної <b>a</b>	Установити для поля <b>Text1</b> подію <b>KeyPress</b> , увести текст підпрограми: <b>Sub Text1_KeyPress (keyAscii As Integer)</b> ‘Контроль уведення даних <b>Select Case keyAscii</b> <b>Case 0, 8, 13, 45, 46, 48 To 57</b> <b>Case Else</b> <b>keyAscii = 0</b> <b>Mes.Caption= «Уведіть число!»</b> <b>End Select</b> <b>End Sub</b>
4	Завдання видачі повідомлення при клацанні по полю введення <b>x</b>	У лівому верхньому списку, що випадає, вибрати об’єкт <b>Text2</b> , установити подію <b>Click</b> , увести текст підпрограми: <b>Sub Text2_Click ()</b> ‘При клацанні по полю <b>Text2</b> буде ‘видаватися повідомлення <b>Mes.Caption = «Уведення перемінної x»</b> <b>End Sub</b>
5	Контроль уведення даних у поле введення перемінної <b>x</b>	У вікні коду <b>Text1</b> події <b>KeyPress</b> виділити рядки від <b>Select case</b> до <b>End select</b> і виконати: пункт меню <b>Edit</b> → команда <b>Copy</b> ; відкрити вікно коду текстового вікна <b>Text2</b> , установити подію <b>KeyPress</b> , потім



Продовження таблиці 8.5.

1	2	3
		виконати: пункт меню <b>Edit</b> → команда <b>Paste</b> .
6	Завдання функції кнопки <b>Вихід</b>	Вибрати об'єкт <b>Vixod</b> і для події <b>Click</b> записати: <b>Private Sub Vixod_Click()</b> <b>End</b> <b>End Sub</b>
7	Завдання функцій кнопки <b>Очищення</b>	Вибрати об'єкт <b>Och</b> (властивість <b>Name</b> має ім'я <b>Och</b> ) і для події <b>Click</b> записати: <b>Private Sub Och_Click()</b> <b>Text1.Text = “ “</b> <b>Text2.Text = “ “</b> <b>Text3.Text = “ “</b> <b>Mes.Caption = “ “</b> <b>End Sub</b>
8	Завдання функцій кнопки <b>Розрахунок</b>	Вибрати об'єкт <b>Rachet</b> і для події <b>Click</b> записати: <b>Private Sub Rachet_Click()</b> <b>Dim a, x As Single</b> <b>a = Val(Text1.Text)</b> <b>x = Val(Text2.Text)</b> <b>If x &gt;= 0 Then Text3.Text = a * x Else</b> <b>Text3.Text = a + x</b> <i>' – оператор If необхідно записувати в один рядок</i> <b>End Sub</b>
9	Зберегти проект з ім'ям « <b>Просте розгалуження</b> »	Пункт меню <b>File</b> → команда <b>Save Project</b> → задати імена файлам і вибрати папку для збереження.
10	Налагодити програму	Пункт меню <b>Run</b> → команда <b>Start</b> , увести вхідні дані контрольного прикладу: <ul style="list-style-type: none"> <li>• <i>для першої гілки</i> – <b>a = 2, x = 8</b>, повинна вийти відповідь <b>y = 16</b>. Потім клацнути по кнопці <b>Очищення</b> і ввести – <b>a = 10, x = 0</b>, повинна вийти відповідь <b>y = 0</b></li> <li>• Повторити очищення і ввести дані для <i>другої гілки</i> – <b>a = 7, x = -2</b>, повинна получитися відповідь <b>y = 5</b>. При виявленні помилок їх варто виправити.</li> </ul>
11	Налагоджену програму зберегти повторно.	Пункт меню <b>File</b> → команда <b>Save Project</b>

**Приклад 8.6** – Записати й налагодити програму складного розгалуженого процесу:

$$y = \begin{cases} x / a - x / b & \text{при } a \leq 5 \text{ і } x > 6, \\ (a + b) / x & \text{при } a \leq 5 \text{ і } x = 6, \\ a^3 + b * e^{2.5} & \text{в інших випадках.} \end{cases}$$

**Виконання.** Спочатку необхідно спроектувати електронну форму (див. рисунок 8.11).

Занесення формули рекомендується виконати в текстовому редакторі *Word*, а потім скопіювати в керуючий елемент **Image**.

Алгоритм рішення приклада розглянутий у розділі 7 – приклад 7.3.

Послідовність операцій по виконанню приклада 8.6 «Складне розгалуження» описана в таблиці 8.6.

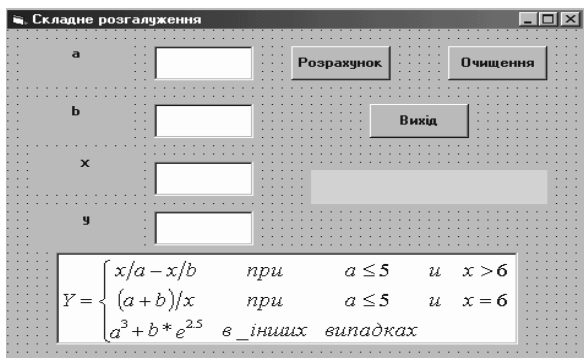


Рисунок 8.11 – Екранна форма для приклада 8.6 – “Складне розгалуження”

Таблиця 8.6. – Операції по виконанню приклада 8.6 – «Складне розгалуження».

№ п/п	Операція	Виконання
1	2	3
1	Створення електронної форми і завдання властивостей її елементів	Дії аналогічні попереднім прикладам (рисунок 8.11). Імена текстових полів залишити стандартними.
2	Контроль уведення даних у поле <b>Text1</b>	Установити для текстового поля <b>Text1</b> подію <b>KeyPress</b> , увести текст підпрограми: <b>Sub Text1_KeyPress (keyAscii As Integer)</b>

Продовження таблиці 8.6.

1	2	3
		<pre> ‘Контроль уведення даних Select Case keyAscii Case 0, 8, 13, 45, 46, 48 To 57 Case Else keyAscii = 0 Mes.Caption= «Уведіть число!» End Select End Sub </pre>
3	Контроль уведення даних у поле <b>Text2</b>	<p>У підпрограмі <b>Sub Text1_KeyPress (keyAscii As Integer)</b> виділити текст підпрограми → меню <b>Edit</b> → команда <b>Copy</b> → установити для текстового поля <b>Text2</b> подію <b>KeyPress</b> → меню <b>Edit</b> → команда <b>Paste</b>.</p>
4	Контроль уведення даних у поле <b>Text3</b>	<p>Установити для текстового поля <b>Text3</b> подію <b>KeyPress</b> → меню <b>Edit</b> → команда <b>Paste</b>.</p>
5	Завдання функції кнопки <b>Вихід</b>	<p>Вибрати об'єкт <b>Vixod</b> і для події <b>Click</b> записати:</p> <pre> Private Sub Vixod_Click() End End Sub </pre>
6	Завдання функцій кнопок <b>Очищення</b>	<p>Вибрати об'єкт <b>Och</b> (властивість <b>Name</b> має ім'я <b>Och</b>) і для події <b>Click</b> записати:</p> <pre> Private Sub Och_Click() Text1.Text = “ “ Text2.Text = “ “ Text3.Text = ” “ Text4.Text = “ “ Mes.Caption = “ “ End Sub </pre>
7	Завдання функцій кнопок <b>Розрахунок</b>	<p>Вибрати об'єкт <b>Rachet</b> і для події <b>Click</b> записати:</p> <pre> Private Sub Rachet_Click() Dim a,b,x As Single a = Val(Text1.Text) b = Val(Text2.Text) x = Val(Text3.Text) If a &lt;= 5 And x = 6 Then ' натиснути ↵ Text4.Text = (a + b) / x ' натиснути ↵ </pre>

Продовження таблиці 8.6.

1	2	3
		<pre> ElseIf a &lt;= 5 And x &gt; 6 And a &lt;&gt; 0 And_ b &lt;&gt; 0 Then     ' натиснути ↵     Text4.Text = x / a - x / b ' натиснути ↵ Else     ' натиснути ↵     Text4.Text = a ^ 3 + b * exp(2.5) ' ↵ End If End Sub                     </pre>
8	Зберегти проект з ім'ям «Складне розгалуження»	Пункт меню <b>File</b> → команда <b>Save Project</b> → задати імена файлам і вибрати папку для збереження.
9	Налагодити програму	<p>Меню <b>Run</b> → команда <b>Start</b>, увести вхідні дані контрольного приклада:</p> <ul style="list-style-type: none"> <li>• для <i>першої гілки</i> <b>a = 2, b = 5, x = 10</b>, повинна вийти відповідь <b>y = 3</b>.</li> <li>• Потім клацнути по кнопці <b>Очищення</b> і ввести дані <i>для другої гілки</i> – <b>a = 2, b = 10, x = 6;</b> <b>відповідь y = 2.</b></li> <li>• Повторити очищення і ввести дані для <i>3-й гілки</i> – <b>a = 10, b = 4, x = 2</b> повинна вийти відповідь <b>y ≈ 1048,73</b>.</li> </ul> <p>При виявленні помилок їх варто виправити.</p>
10	Налагоджену програму зберегти повторно.	Пункт меню <b>File</b> → команда <b>Save Project</b> .

## 8.6 Селекторні кнопки (перемикачі), прапорці, рамки

В усіх попередніх прикладах користувач вирішував задачу «наосліп» – задавав дані й одержував відповідь. Усе, про що його інформувала програма – це умова задачі. Можна зробити користувача більш активним учасником рішення – дати йому можливість вибрати потрібну формулу для розрахунку або кілька необхідних параметрів задачі. Для цього на першому етапі проектування варто передбачити використання таких керуючих елементів, як **селекторні кнопки (перемикачі)**,





- **виключена**. Для установки такого стану необхідно вибрати значення властивості **Value** рівним **False** (**Неправда**).

Аналізуючи значення властивості **Value**, програма вибирає формулу (задачу) для рішення.

Щоб об'єднати й ідентифікувати кілька селекторних кнопок у групу, варто спочатку створити в межах форми їхню групову рамку за допомогою елемента панелі інструментів **Frame**. Допускається розміщення перемикачів і прямо у формі, без їхнього об'єднання в групу.

Приклад 8.7 – Обчислити значення функції по одній із трьох формул:

$$y = x / 6, \quad y = x * 5 \quad \text{або} \quad y = x / 4 \quad \text{для} \quad 0 < x \leq 100$$

Для вибору формул при розрахунку використовувати перемикачі. Значення перемінної  $x$  задавати за допомогою лінійки прокручування.

#### Виконання:

1. Створити екранну форму (малюнок 8.13)

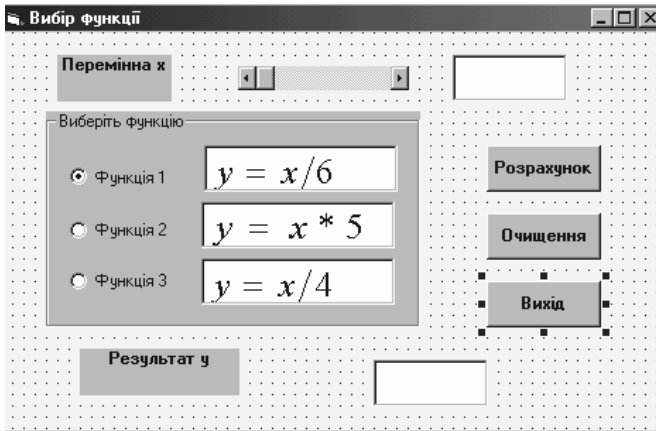


Рисунок 8.13 – Вид екранної форми для приклада 8.7 – “Вибір функції”

Послідовність дій по створенню форми приведена в таблиці 8.7

Таблиця 8.7. – Послідовність дій по створенню форми – «Вибір функції»

№ п/п	Операція	Властивість об'єкта	Значення властивості
1	2	3	4
1	Дати ім'я електронній формі	<b>Caption</b>	<b>Вибір функції</b>
2	Створити лінійку прокручування для вибору значень перемінної x (інструмент <b>HScrollBar</b> )	<b>Name</b>	<b>LineX</b>
		<b>Max</b>	<b>100</b>
		<b>Min</b>	<b>0</b>
		<b>Value</b>	<b>0</b>
		<b>SmallChange</b>	<b>1</b>
	<b>LargeChange</b>	<b>10</b>	
3	Створити рамку для групи перемикачів (інструмент <b>Frame</b> )	<b>Caption</b>	<b>Виберіть функцію</b>
4	Створити перемикач для вибору функції 1 (інструмент <b>OptionButton</b> )	<b>Name</b>	<b>F1</b>
		<b>Caption</b>	<b>Функція 1</b>
		<b>Value</b>	<b>True</b>
5	Створити перемикач для вибору функції 2 (інструмент <b>OptionButton</b> )	<b>Name</b>	<b>F2</b>
		<b>Caption</b>	<b>Функція 2</b>
		<b>Value</b>	<b>False</b>
6	Створити перемикач для вибору функції 3 (інструмент <b>OptionButton</b> )	<b>Name</b>	<b>F3</b>
		<b>Caption</b>	<b>Функція 3</b>
		<b>Value</b>	<b>False</b>
7	Створити у формі для ілюстрації розрахункову формулу $y = x / 6$ (інструмент <b>OLE</b> )	<b>Name</b>	<b>OLE1</b>
8	Аналогічно п.7 дві інші формули	<b>Name</b>	<b>OLE2</b>
		<b>Name</b>	<b>OLE3</b>
9	Створити етикетку для ідентифікації поля зі значенням перемінної x (інструмент <b>Label</b> )	<b>Name</b>	<b>Label1</b>
		<b>Caption</b>	<b>Значення x</b>
10	Створити поле для перегляду значення перемінної x (інструмент <b>Text</b> )	<b>Name</b>	x
		<b>Text</b>	
11	Створити етикетку для ідентифікації поля результату розрахунків y (інструмент <b>Label</b> )	<b>Name</b>	<b>Label2</b>
		<b>Caption</b>	<b>Результат y</b>

Продовження таблиці 8.7.

1	2	3	4
12	Створити поле для перегляду значення результату <b>в</b> (інструмент <b>Text</b> )	<b>Name</b>	<b>у</b>
		<b>Text</b>	
13	Створити командну кнопку для виконання розрахунку по обр-ній формулі (інструмент <b>CommandButton</b> )	<b>Name</b>	<b>Розрахунок</b>
		<b>Caption</b>	<b>Розрахунок</b>
14	Створити командну кнопку для очищення полів <b>x</b> і <b>y</b> (інструмент <b>CommandButton</b> )	<b>Name</b>	<b>Очищення</b>
		<b>Caption</b>	<b>Очищення</b>
15	Створити командну кнопку для виходу з програми (інструмент <b>CommandButton</b> )	<b>Name</b>	<b>Вихід</b>
		<b>Caption</b>	<b>Вихід</b>

## 2. Створити програмний код.

1). Подвійним клацанням лівої кнопки мишки по лінійці прокручування в формі проекту відкрити вікно **Code**.

Для об'єкта **Line** вибрати подію **Change** і дописати процедуру, виконувану, якщо уже відбудеться подія – переміщення бігунка на лінійці:

```
Private Sub LineX_Change()  
    x.Text = LineX.Value  
End Sub
```

2). Для об'єкта **Line** вибрати подію **Scroll** і дописати зміст процедури, виконуваної під час переміщення бігунка по лінійці:

```
Private Sub LineX_Scroll()  
    LineX_Change  
End Sub
```

3). Вибрати в лівому списку об'єкт **Розрахунок**, а в правому – подію **Click**. Описати послідовність дій, виконуваних у програмі при клацанні мишкою по кнопці «**Розрахунок**»:



```

Private Sub Розрахунок_Click()
  If F1.Value = True Then
    y.Text = Val(x.Text) / 6
  ElseIf F2.Value = True Then
    y.Text = Val(x.Text) * 5
  Else
    y.Text = Val(x.Text) / 4
  End If
End Sub

```

4). Вибрати в лівому списку об'єкт **Очищення**, а в правому – подію **Click**. Описати послідовність дій, виконуваних у програмі при клацанні мишкою по кнопці «**Очищення**»:

```

Private Sub Очищення_Click()
  x.Text = " "
  y.Text = " "
End Sub

```

5). Вибрати в лівому списку об'єкт **Вихід**, а в правому – подію **Click**. Описати послідовність дій, виконуваних у програмі при клацанні мишкою по кнопці «**Вихід**»

```

Private Sub Вихід_Click()
  End
End Sub

```

У прикладі 8.7 необхідно було для виконання розрахунків вибрати одну з трьох формул, якщо ж потрібно робити вибір типу «**Так/Ні**», то варто використовувати *прапорець*. Наявність «галочки» у квадратику відповідає «**Так**» (властивість **Value – 1 Checked**), відсутність – «**Ні**» (властивість **Value – 0 Unchecked**).

Приклад 8.8 – Спроекувати додаток із розрахунку ціни підключення до системи кабельного телебачення: варто вибрати пакет – *мінімальний* (10 програм), *повний* (50 програм), *вибірковий*; указати наявність спеціального кабелю; забезпечити введення стандартної ціни підключення, необхідного метражу кабелю і його ціни. При виборі типу пакета варто інформувати користувача про розмір щомісячної оплати в

майбутньому (мінімальний – 10 грн., повний – 20 грн., вибірковий – 15 грн).

**Виконання.** При проектуванні форми вибір пакета здійснювати за допомогою селекторних кнопок, при цьому клацання по кнопці повинне викликати інформаційне вікно з даними по майбутній щомісячній оплаті; наявність кабелю вказувати прапорцем, введення й вивід здійснювати через текстові вікна. Усі повідомлення видаються у виді інформаційних вікон. Плату за установку і щомісячну оплату занести у виді констант, оголосивши їх у головному модулі. Плату за установку занести в текстове вікно при завантаженні форми з константи. Екранна форма представлена на рисунку 8.14, послідовність дій по виконанню – у таблиці 8.8. Як імена перемінних і констант використовувати значенні назви.

Рисунок 8.14 – Екранна форма для прикладу 8.8 – «Кабельне телебачення»

Таблиця 8.8. – Послідовність дій для рішення прикладу 8.8 – «Кабельне телебачення»

№ п/п	Операція	Виконання
1	2	3
1	Створення електронної форми і завантаження	Дії аналогічні попереднім прикладам (рисунком 8.14). Імена текстових полів: <b>Ціна_підкл</b> , <b>Ціна_кабелю</b> , <b>Кількість</b> ,

## Продовження таблиці 8.8.

1	2	3
	властивостей її елементів	<p><b>Усього.</b>            Заборонити користувачеві коректувати значення текстового поля <b>Ціна_підкл</b> – властивість <b>Enabled – False</b>.            Імена селекторних кнопок – <b>Min, Full, M_F</b>, прапорець має стандартне ім'я <b>Check1</b> (Властивість <b>Value – 1 Checked</b>).</p>
2	Завдання констант у головному модулі програми	<p>У списку об'єктів вікна <b>Code</b> викликати <b>General</b>, а в списку подій – <b>Declaration</b>, увести код:  <b>Const Плата_хв = 10,00</b> (вийде 10#)  <b>Const Плата_повн = 20,00</b> (аналогічно)  <b>Const Плата_виб = 15,00</b> ----- “ -----  <b>Const Установка = 45,00</b> ----- “ -----</p>
3	Присвоєння значення полю <b>Ціна_підкл</b> при завантаженні форми	<p>У списку об'єктів вікна <b>Code</b> викликати <b>Form</b>, а в списку подій – <b>Load</b> (завантажити), увести код:  <b>Private Sub Form_Load()</b>  <b>Ціна_підкл.Text = Установка</b>  <b>End Sub</b></p>
4	Контроль уведення даних у поле <b>Ціна_кабелю</b>	<p>Установити для текстового поля <b>Ціна_кабелю</b> подію <b>KeyPress</b>, увести текст підпрограми:  <b>Sub Ціна_кабелю_KeyPress (keyAscii As Integer)</b>        ‘Контроль уведення даних  <b>Select Case keyAscii</b>        <b>Case 0, 8, 13, 45, 46, 48 To 57</b>        <b>Case Else</b>            <b>KeyAscii = 0</b>            <b>MsgBox “Помилка! Уведіть ціну кабелю”</b>        <b>End Select</b>  <b>End Sub</b></p>
5	Контроль уведення даних у поле <b>Кількість</b>	<p>Установити для текстового поля <b>Кількість</b> подію <b>KeyPress</b>, увести текст підпрограми:  <b>Sub Кількість_KeyPress (keyAscii As Integer)</b>  <b>Select Case keyAscii</b></p>

Продовження таблиці 8.8.

1	2	3
		<b>Case 0, 8, 13, 45, 46, 48 To 57</b> <b>Case Else</b> <b>KeyAscii = 0</b> <b>MsgBox "Помилка! Уведіть_ необхідний метраж кабелю"</b> <b>End Select</b> <b>End Sub</b>
6	Завдання функції кнопки <b>Вихід</b>	Установити об'єкт Вихід і для події Click записати: <b>Private Sub Вихід_Click()</b> <b>End</b> <b>End Sub</b>
7	Завдання функції кнопки <b>Очищення</b>	Установити об'єкт <b>Очищення</b> (властивість <b>Name</b> має ім'я <b>Очищення</b> ) і для події <b>Click</b> записати: <b>Private Sub Очищення_Click()</b> <b>Ціна_кабелю = " "</b> <b>Кількість = " "</b> <b>Усього = " "</b> <b>End Sub</b>
8	Проектування видачі повідомлення при виборі селекторної кнопки « <b>мінімальний</b> »	Установити об'єкт <b>Min</b> , подію <b>Click</b> . <b>Private Sub Min_Click()</b> <b>MsgBox "щомісячна плата" + _</b> <b>Str(Плата_хв)</b> <b>End Sub</b>
9	Проектування видачі повідомлення при виборі селекторної кнопки « <b>повний</b> »	Установити об'єкт <b>Full</b> , подія <b>Click</b> . <b>Private Sub Full_Click()</b> <b>MsgBox "щомісячна плата" + _</b> <b>Str(Плата_повн)</b> <b>End Sub</b>
10	Проектування видачі повідомлення при виборі селекторної кнопки « <b>вибірковий</b> »	Установити об'єкт <b>M_F</b> , подія <b>Click</b> . <b>Private Sub M_F_Click()</b> <b>MsgBox "щомісячна плата" + _</b> <b>Str(Плата_виб)</b> <b>End Sub</b>
11	Завдання функції кнопки <b>Розрахунок</b>	Установити об'єкт <b>Розрахунок</b> і для події <b>Click</b> записати: <b>Private Sub Розрахунок_Click()</b> <b>If Check1.Value = 1 And Ціна_кабелю.Text</b>

Продовження таблиці 8.8.

1	2	3
		<pre> = "" Then MsgBox "Уведіть ціну кабелю":_ GoTo M1 ( If Check1.Value = 1 And Кількість.Text= "" Then MsgBox "Уведіть необхідний_ метраж кабелю": GoTo M1 ↵ If Check1.Value = 1 Then Усього.Text =_ Val(Ціна_підкл.Text) +_ Val(Ціна_кабелю.Text) *_ Val(Кількість.Text) Else Усього.Text =_ Val(Ціна_підкл.Text) ↵ M1: End Sub </pre>
12	Зберегти проект з ім'ям «Кабельне телебачення»	Меню <b>File</b> → команда <b>Save Project</b> → задати ім'я папки й дати імена файлам.
13	Налагодити програму	Меню <b>Run</b> → команда <b>Start</b> .
14	Налагоджену програму зберегти повторно.	Меню <b>File</b> → команда <b>Save Project</b> → дати ім'я <b>Кабельне телебачення</b> → <b>OK</b>

**Примітка** – Константи використовуються в програмі для присвоєння умовно-постійних даних. При зміні цін варто поміняти тільки значення констант, а інші підпрограми залишаються тими ж самими.

### Питання і завдання для самостійного виконання

1. Що таке розгалужений процес?
2. Яка кількість розгалужень допускається в алгоритмах, що розгалужуються?
3. Якими блоками позначається перевірка умов у розгалужених алгоритмах, їхні особливості?
4. Які оператори мови характерні саме для розгалужених алгоритмів, як вони працюють?
5. Чим відрізняється складний оператор **If ... Then ... Else** від простого?
6. Що означає неповний оператор **If ...**?
7. Для чого використовуються селекторні кнопки в програмах, як задати їм початкове значення?
8. Призначення прапорців, привести приклади.
9. Як змусити ПК видавати повідомлення у виді інформаційних вікон?
10. Для чого використовуються константи в програмах?
11. Спроекувати програму на Visual-Basic:

а) розрахунку податку на зарплату: якщо нараховано не більше 150 гривень, то податок не платиться. Від суми більш 150 гривень, але менш 350 гривень нараховується 10% податку, більш 350 гривень – 15% податку.

$$H = \begin{cases} 0 & \text{при } Z \leq 150, \\ 0,1 * Z & \text{при } 150 < Z \leq 350, \\ 0,15 * Z & \text{при } >350. \end{cases}$$

$$б) \quad y = \begin{cases} a * c + 8,5 * x & \text{при } x > 7, \\ a * c & \text{при } x = 7, \\ (a * c - b) / 5 * x & \text{при } x < 7. \end{cases}$$

$$в) \quad y = \begin{cases} 5 * z + 5 - 4 * x & \text{при } a = 1, \\ 0,5 * e^x - (x - 1)^2 & \text{при } a = 2, \\ 625 & \text{в інших випадках.} \end{cases}$$

г) Створити проект із розрахунку вартості замовлення на одну людину в підприємстві суспільного харчування, якщо в меню міститься наступна інформація:

Перші блюда	Ціна	Другі блюда	Ціна
1. Борщ український з пампушками	3,20	1. Котлети по – донбаські	3,12
2. Щи	2,85	2. Кури гриль	5,07
3. Суп курячий з вермішельлю	2,50	3. Риба смажена	2,08
4. Розсольник	2,70	<b>Гарніри</b>	
5. Солянка	2,45	1. Картопля відварна	0,50
		2. Капуста тушкована	0,46
		3. Каша гречана	0,39
<b>Напої</b>		<b>Салати</b>	
1. Чай	0,50	1. Салат зі св. капусти	0,25
2. Кава	0,60	2. Салат з огірків	0,30
3. Компот із св. фруктів	0,70	3. Салат із помідорів	0,56

Замовник може вказати, чи потрібен йому хліб, а при виборі чаю або кави – чи потрібен цукор.

## 8.7 Проектування циклічних процесів

Як було розглянуто в 7.3.3, циклічні програми можуть бути різних видів – арифметичні, ітераційні, складні, з індексованими перемінними. Для їхнього написання використовуються різні оператори.

### 8.7.1 Арифметичні цикли, використання декількох форм у проекті

**Арифметичні цикли** – цикли, у яких оператори повторюються заздалегідь відоме число раз, для їхнього запису використовується наступний оператор:

**For** *перемінна* = початкове значення **To** кінцеве значення [**Step...**]  
оператори  
**[Exit For]**  
оператори  
**Next** [*перемінна* [,*перемінна*]...]

Відповідно до цього оператора перемінній буде привласнене початкове значення, виконана перевірка на кінець циклу, якщо не кінець, то будуть виконані оператори циклу до **Next**, значення перемінної буде збільшено на значення кроку (**Step**) або, якщо останній не зазначений, – на **1**, потім усе знову повторюється з процесу перевірки на кінець циклу. Якщо цикл закінчено, то починає виконуватися оператор, що стоїть після **Next**. **Exit For** – використовується при достроковому виході з циклу.

Приклад 8.9 – Написати програму обчислення  $y = n!$  (блок-схема на ри-сунку 7.4 приклада 7.4, операції по виконанню – таблиця 8.9 , вид екранної форми на рисунку 8.15)

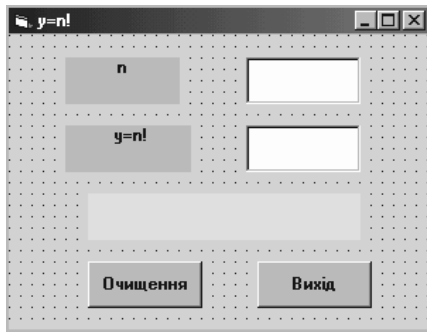


Рисунок 8.15 – Вид екранної форми для приклада 8.9 – “Факторіал”

Таблиця 8.9. – Операції по виконанню приклада 8.9 – «Факторіал»

№ п/п	Операція	Виконання
1	2	3
1	Створення електронної форми і завдання властивостей її елементів	Дії аналогічні попереднім прикладам (рисунок 8.15). Імена текстових полів <b>n</b> і <b>y</b> , поля напису для повідомлення – <b>Mes</b> , імена кнопок – <b>Очищення</b> і <b>Вихід</b>
2	Завдання функції кнопки	Установити об'єкт <b>Вихід</b> і для події
	<b>Вихід</b>	<b>Click</b> записати: <b>Private Sub Вихід_Click()</b> <b>End</b> <b>End Sub</b>
3	Завдання функцій кнопки <b>Очищення</b>	Установити об'єкт <b>Очищення</b> і для події <b>Click</b> записати: <b>Private Sub Очищення_Click()</b> <b>n.Text = “ ”</b> <b>y.Text = “ ”</b> <b>Mes.caption = “ ”</b> <b>End Sub</b>
4	Контроль уведення даних у поле <b>n</b> і розрахунок факторіала	Установити для текстового поля <b>n</b> подію <b>KeyPress</b> , увести текст підпрограми: <b>Sub n_KeyPress (KeyAscii As Integer)</b> <b>Dim i As Integer</b> ‘оголошення параметра циклу як цілого <b>y = 1</b> ‘початкове значення <b>y</b> ‘Контроль уведення даних <b>Select Case KeyAscii</b> <b>Case 0, 8, 46, 48 To 57</b> <b>Case 13</b> <b>For i = 1 To Val(n.Text)</b> <b>y.Text = Val(y.Text) * i</b> <b>Next i</b> <b>Case Else</b> <b>Mes.Caption = “Уведіть позитивне _</b> <b>число “</b> <b>End Select</b> <b>End Sub</b>
5	Зберегти проект з ім'ям « <b>Факторіал</b> »	Меню <b>File</b> → команда <b>Save Project</b> → задати ім'я файлу й папку.



Продовження таблиці 8.9.

1	2	3
6	Налагодити програму	<p>Меню <b>Run</b> → команда <b>Start</b>, увести вхідні дані контрольного приклада:</p> <ul style="list-style-type: none"> <li>• <b>n = 5</b> і натиснути клавішу <b>Enter</b>, повинна вийти відповідь <b>y = 120</b>.</li> <li>• Потім клацнути по кнопці <b>Очищення</b> і ввести <b>n = -7</b>, повинне з'явитися повідомлення.</li> </ul> <p>При виявленні помилок їх варто виправити.</p>
7	Налагоджену програму зберегти	Меню <b>File</b> → команда <b>Save Project</b> .

У попередньому прикладі використовувалися прості перемінні, котрі вводилися в текстові вікна. У наступному прикладі розглянемо *арифметичний цикл з індексованими перемінними, тобто роботу з масивом*.

Приклад 8.10 – Написати програму обчислення суми товарообігу підприємства за місяць, якщо відомо щоденний товарообіг по формулі:

$$y = \sum_{i=1}^n x_i$$

де: **n** – кількість днів у місяці,  
**x<sub>i</sub>** – щоденний товарообіг,  
**i** – поточний день місяця.

Форма проекту представлена на малюнку 8.16

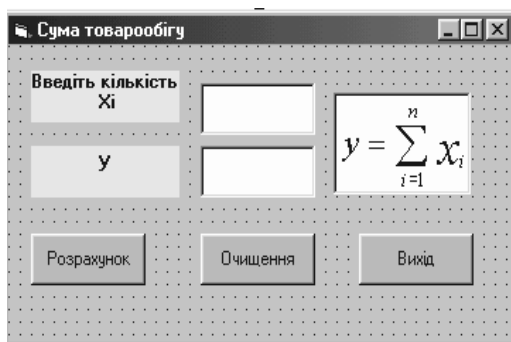


Рисунок 8.16 – Вид екранної форми для приклада 8.10 – “Цикл із запитом”

## Виконання:

У даному прикладі з'явився новий запис –  $x_i$ , що називається індексованою перемінною і використовується при роботі з масивами (приклад 7.6, блок-схема на рисунку – 7.6). Для занесення таких даних можна використовувати засіб убудованого діалогу – *панель введення*. Панель введення викликається функцією **InputBox**.

Спрощений формат функції:

< перемінна > = **InputBox** (“текст запрошення, видаваного користувачеві“)

Операції по виконанню приклада показані в таблиці 8.10.

Таблиця 8.10. – Операції по виконанню приклада 8.10 – «Цикл із запитом»

№ п/п	Операція	Виконання
1	2	3
1	Створення електронної форми «Цикл з запитом» і завдання властивостей її елементів	Дії аналогічні попереднім прикладам (рисунк 8.16). Ім'я текстового поля для введення кількості $x_i$ – <b>n</b> .
2	Завдання функції кнопки <b>Вихід</b>	Установити об'єкт <b>Вихід</b> і для події <b>Click</b> записати: <b>Private Sub Вихід_Click()</b> <b>End</b> <b>End Sub</b>
3	Завдання функцій кнопки <b>Очищення</b>	Установити об'єкт <b>Очищення</b> і для події <b>Click</b> записати: <b>Private Sub Очищення_Click()</b> <b>n.Text = “ “</b> <b>y.Text = “ “</b> <b>End Sub</b>
4	Контроль введення даних у поле <b>n</b>	Установити для текстового поля <b>n</b> подію <b>KeyPress</b> , увести текст підпрограми: <b>Private Sub n_KeyPress(KeyAscii As Integer)</b> <b>Select Case KeyAscii</b> <b>Case 0, 8, 13, 48 To 57</b> <b>Case Else</b> <b>KeyAscii = 0</b> <b>MsgBox “Помилка! Уведіть кількість_робочих днів у місяці”</b> <b>End Select</b> <b>End Sub</b>

Продовження таблиці 8.10.

1	2	3
5	Уведення підпрограми для кнопки <b>Розрахунок</b> .	Установити об'єкт <b>Розрахунок</b> і для події <b>Click</b> записати: <b>Private Sub Розрахунок_Click()</b> <b>Dim i As Integer, p As Single</b> <b>For i = 1 To Val(n.Text)</b> <b>p = InputBox("Уведіть товарообіг_за"</b> <b>&amp; Str(i) &amp; " день")</b> <b>y.Text = Val(y.Text) + p</b> <b>Next i</b> <b>End Sub</b>
6	Зберегти проект із ім'ям «Цикл із запитом»	Меню <b>File</b> → команда <b>Save Project</b> → задати ім'я файлу й папку.
7	Налагодити програму	Меню <b>Run</b> → команда <b>Start</b> , ввести кількість днів – <b>2</b> , клацнути по кнопці <b>Розрахунок</b> , ввести у вікно запиту дані контрольного приклада <b>X1 = 5, X2 = 7</b> , повинна вийти відповідь <b>y = 12</b> . Потім клацнути по кнопці <b>Очищення</b> і ввести <b>n = -7,2</b> , ПК не повинен реагувати на введення. При виявленні помилок їх варто виправити.
8	Налагоджену програму зберегти	Меню <b>File</b> → команда <b>Save Project</b> .

У прикладі 8.10 у результаті рішення виводиться підсумкова сума, а на підставі яких даних вона отримана не видно. Можна одержати рішення в більш наочному вигляді – додати в проект нову форму і вивести на цю форму суми щоденної продажі і загальну суму товарообігу за місяць.

Для додавання нової форми в проект використовується:

п.м. **Project** → к. **Add Form**

Також необхідно використовувати такі методи як: **Cls, Show, Hide, Print** (див. 7.2.2).

У нову форму доцільно додати дві кнопки – **Вихід** (властивість **Name** – **Вихід2**) і **Повернутися до розрахунку** (властивість **Name** – **Повернення**). Екранна форма з результатними даними при введенні продажу за 5 днів показана на рисунку 8.17.



Рисунок 8.17 – Екранна форма з результатами для приклада 8.10

Додаткові операції по модифікації приклада 8.10, а також змінена програма кнопки **Розрахунок** показані в таблиці 8.11.

Таблиця 8.11. – Додаткові операції по модифікації приклада 8.10

№ п/п	Операція	Виконання
1	Створення другої електронної форми « <b>Результати</b> » і завдання властивостей її елементів	м. <b>Project</b> → к. <b>Add Form</b> , залишити стандартне ім'я <b>Form2</b> , властивість <b>Caption</b> – <b>Результати</b> . На формі створити дві командні кнопки і дати їм відповідно: <ul style="list-style-type: none"> <li>імена <b>Повернення</b> й <b>Вихід2</b>,</li> <li>властивості <b>Caption</b> – <b>Повернутися до розрахунку</b> й <b>Вихід</b>, як на рисунку 8.17</li> </ul>
2	Завдання функції кнопки <b>Вихід2</b>	Установити об'єкт <b>Вихід2</b> і для події <b>Click</b> записати: <b>Private Sub Вихід2_Click()</b> <b>End</b> <b>End Sub</b>
3	Завдання функції кнопки <b>Повернутися до розрахунку</b>	<b>Private Sub Повернення_Click()</b> <b>Form2.Cls</b> ' <b>Очищення форми</b> <b>Form2.Hide</b> ' <b>Сховати форму</b> <b>Form1.Show</b> ' <b>Показати форму</b> <b>End Sub</b>

Продовження таблиці 8.11

1	2	3
4	Змінена програма кнопки <b>Розрахунок</b>	<pre> Private Sub Розрахунок_Click()   Dim i As Integer, p As Single   Form1.Hide   Form2.Show   For i = 1 To Val(n.Text)     p = InputBox("Уведіть товарообіг_ за" &amp; Str(i) &amp; "день")     y.Text = Val(y.Text) + p     Form2.Print " I="; i, " товарообіг_ "; p   Next i   Form2.Print " ----- -----"   Form2.Print "      Разом      ", y End Sub </pre>

## 8.7.2 Ітераційні цикли

**Ітераційні цикли** – це цикли, у яких заздалегідь не відоме число повторень, але є визначені умови, при яких цикл повинен бути довершений. Для їхнього запису використовуються наступні оператори:

- |  |  |
|--|--|
| 1) <b>Do While</b> умова<br>оператори<br><b>Loop</b> | 2) <b>Do Until</b> умова<br>оператори<br><b>Loop</b> |
|--|--|

Оператори циклу 1) будуть виконуватися доти, поки буде виконуватися умова, тобто умова **правдива**, в іншому випадку управління буде передано оператору, що стоїть за **Loop**.

Оператори циклу 2) будуть виконуватися доти, поки умова **помилкова**. Умова перевіряється перед кожним проходом циклу.

Є модифікації зазначених операторів, коли умова переноситься в кінець, після **Loop**, тоді цикл виконується перший раз обов'язково.

Приклад 8.11. – Визначити витрати повітря, необхідного для видалення надлишку тепла з харчоблока при варіації значень надлишку тепла по формулі:

$$V = \frac{Q}{(T_1 - T_2) * L * C} ,$$

де: V – витрата повітря;

- Q** – надлишок тепла в приміщенні;
- T1** – температура повітря, що йде з приміщення;
- T2** – температура приточного повітря;
- L** – щільність приточного повітря;
- C** – питома теплоємність повітря.

Виконання:

У даному прикладі розрахункова формула повинна повторюватися кілька разів, закінчення розрахунку повинен визначати користувач по запити програми. Електронна форма для приклада представлена на рисунку 8.18.

Дії по формуванню коду різних елементів форми аналогічні попереднім прикладам, відрізняється тільки код кнопки «**Розрахунок**», що приводиться нижче:

- програма кнопки «**Розрахунок**» при використанні операторів циклу 1) :

**Private Sub Розрахунок\_Click()**

**Dim q As Single, a As Integer**

**a = 1**

**Do While a = 1**

**Form1.Hide**

**q = InputBox("Уведіть величину надлишку тепла з харчоблока")**

**v.Text = q / ((Val(t1.Text) – Val(t2.Text)) \* Val(L.Text) \* Val(c.Text))**

**MsgBox ("Витрата повітря – " & V.Text)**

**a = Val(InputBox("Якщо потрібен рахунок -1, ні-0"))**

**Loop**

**Form1.Show**

**End Sub**

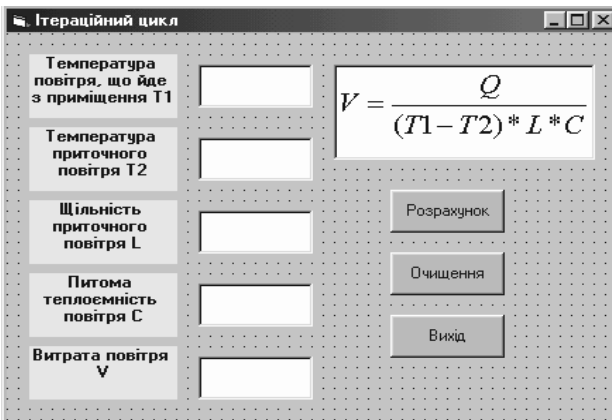


Рисунок 8.18 – Електронна форма для приклада 8.11 – “Визначення витрати повітря”

- програма кнопки «Розрахунок» при використанні операторів циклу 2) :

```

Dim q As Single, a As Integer
a = 1
Do Until a <> 1
Form1.Hide
q = InputBox("Уведіть величину надлишку тепла з харчоблока")
V.Text = q / ((Val(T1.Text) – Val(T2.Text)) * Val(L.Text) *
Val(C.Text))
MsgBox ("Витрата повітря – " & V.Text)
a = Val(InputBox("Якщо потрібен рахунок -1, ні-0"))
Loop
Form1.Show
End Sub

```

### 8.7.3 Складні цикли, використання меню

**Складні** – це цикли з розгалуженнями або вкладеними циклами. Для їхнього написання використовуються ті ж оператори, що і для простих циклів і розгалужених процесів.

Приклад 8.12. – Написати програму підрахунку товарообігу підприємства за місяць (**S**), якщо є дані продажі по окремих найменуваннях товарів щодня ( $x_{ij}$ ),

$j = 1 \div n$ , **n** – кількість найменувань товарів,

$i = 1 \div m$ , **m** – кількість робочих днів у місяці.

$$S = \sum_{i=1}^m \sum_{j=1}^n x_{ij} \quad \text{– вхідна формула.}$$

Виконання. Блок-схему алгоритму див. у прикладі 7.8, рисунок 7.8  
Екранна форма проекту представлена на рисунку 8.19.

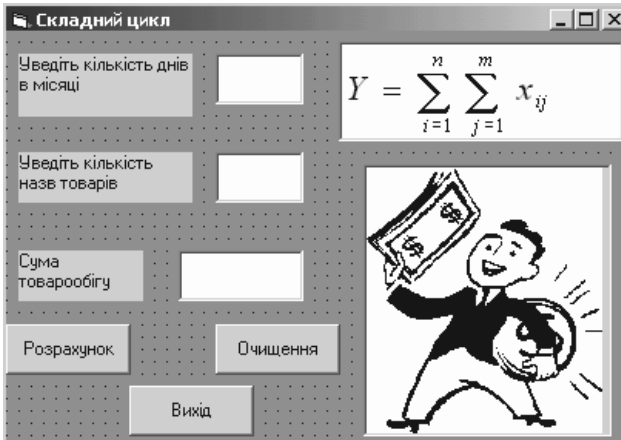


Рисунок 8.19 – Екранна форма для прикладу 8.12

Підпрограми по створенню проекту приведені нижче.

```

Dim i, j As Integer           'Об'єкт Global, подія Declaration
Dim p As Single

Private Sub Очищення_Click()
    n.Text = " "
    m.Text = " "
    S.Text = " "
End Sub

Private Sub Розрахунок_Click()
    Form1.Hide
    For i = 1 To n
        For j = 1 To m
            p = InputBox("Уведіть суму продажу за " & Str(i) & " день по " & Str(j) & " найменуванню")
            S.Text = Val(S.Text) + p
        Next j
    Next i
    MsgBox ("Підсумкова сума товарообігу – " & S.Text)
    Form1.Show
End Sub

```

У результаті роботи спроектованої програми буде отримана відповідь у виді одного числа. У той же час по вхідним даним можна одержувати ін-формацію в різних розрізах:



- продаж за день по всіх найменуваннях,
- продаж по одному найменуванню за місяць.

Але для цього дані повинні зберігатися в машині на час роботи програми у виді масиву.

Масиви можна оголошувати *фіксованого розміру і динамічними*. Якщо *довжина масиву постійна*, то його можна оголосити з указівкою розмірності, наприклад, **Dim p(1, 20) as Single**. Однак оголошення масиву постійної розмірності не завжди зручно.

Для приклада 8.12 кількість робочих днів у місяці коливається від 21 до 31 у підприємстві, а для контрольного приклада досить перевірити 2 точки. Можна змусити програму порахувати суму товарообігу за будь-яку кількість днів. Також кількість найменувань є змінною величиною.

У Visual Basic є можливість визначити необхідний розмір пам'яті для збереження індексованих перемінних безпосередньо при виконанні програми по даним, що фактично вводяться. Для цього оголошують *динамічний масив*. Оголошення його йде у два прийоми:

- 1) звичайне оголошення масиву, але з порожньою розмірністю – **Dim x() As Single**;
- 2) за допомогою оператора **ReDim** указати розмірність масиву у виді числа або вираження – **ReDim x(1 To n, 1 To m)**.

Крім того, необхідно надати користувачеві можливість вибирати потрібні варіанти розрахунку. Для цього зручно використовувати меню, що не важко створити в Visual Basic.

Для проектування меню усіх видів використовується редактор меню **Menu Editor**, що викликається:

п. м. **Tools** → к. **Menu Editor**

Вікно редактора показане на рисунку 8.20.

Елементи управління властивостями меню наступні:

- поле **Caption** – найменування пункту меню, що буде бачити користувач.
- поле **Name** – ім'я меню для звертання до нього в коді програми.
- поле **Index** – ідентифікатор пункту меню в масиві елементів керування.
- список **Shortcut**, що випадає, – призначає комбінацію клавіш для швидкого виклику команди меню.
- поле **HelpContextID** – посилання на тему в довідковій системі.

- прапорець **Checked** – має значення, якщо при включенні команди ліворуч від нього з'являється галочка.
- прапорець **Enabled** – має значення **True**, якщо пункт меню доступний.
- прапорець **Visible** – має значення **True**, якщо пункт меню видний на екрані.
- прапорець **WindowsList** – визначає наявність списку відкритих вікон.

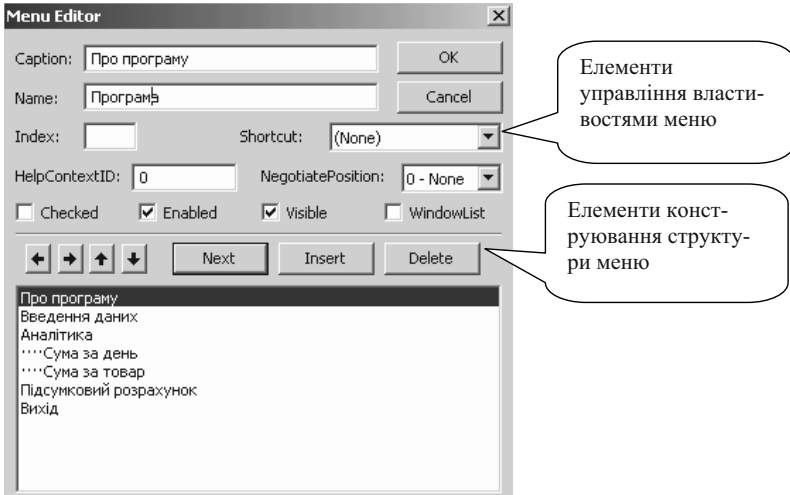


Рисунок 8.20 – Вікно редактора меню Menu Editor

Елементи конструювання структури меню дозволяють додавати (**Insert**), видаляти (**Delete**) пункти меню, переміщати пункти по вертикалі (стрілки нагору, униз), змінюючи порядок їхнього проходження, і по горизонталі (стрілки вліво, вправо), змінюючи розташування пунктів в ієрархії системи меню.

Для кожного пункту меню можна викликати вікно коду подвійним клацанням по пункті.

Модифікуємо умову приклада 8.12

**Приклад 8.13.** – Написати програму підрахунку товарообігу підприємства за місяць, по днях місяця і по окремих найменуваннях за місяць, якщо є дані продажі по окремих найменуваннях товарів щодня ( $x_{ij}$ ), де  $n$  – кількість найменувань товарів,  $m$  – кількість робочих днів у місяці. Надати користувачеві можливість вибирати необхідні види розрахунків через меню.

**Виконання.** У програмі варто створити форму з меню (рисунок 8.21) і допоміжні форми для виведення результатів роботи окремих пунктів меню (крім **Підсумковий розрахунок** і **Вихід**). Допоміжні форми будуть містити тільки одну командну кнопку – **ОК**. Розрахункові пункти меню повинні бути доступні користувачу тільки при завданні значень **m** і **n**.

Оскільки звертання до масиву можливо з декількох пунктів меню, то оголосити його необхідно як глобальний – у розділі **General Declaration** головного модуля додатка.

Для написання циклічних програм необхідно скласти робочі формули. Для підрахунку підсумкової суми товарообігу використовується формула попереднього приклада.

Визначення суми продажу по днях припускає підсумовування всіх сум по найменуваннях товарів, проданим за один день:

$$z = z + x(i,j) \quad \text{при} \quad j = 1 + m,$$

де: **z** – сума продажу за **i** – **ий** день,

**x(i,j)** – сума продажу за **i** – **ий** день по **m** найменуваннях товарів.



Рисунок 8.21 – Екранна форма для прикладу 8.13 – «Складний цикл із меню»

Визначення суми продажу по окремих найменуваннях за місяць припускає підсумовування всіх сум по окремому найменуванню за всі дні місяця:

$$z = z + x(i, j) \text{ при } i = 1 + n,$$

де:  $z$  – сума продажу по  $j$  – му найменуванню товару за  $n$  днів місяця.

Оскільки зазначені суми обчислюються в різних незалежних друг від друга пунктах меню, то у формулах можна використовувати одну й ту перемінну  $z$  із початковим значенням = 0 в обох випадках. Оголосити її необхідно як глобальну.

Операції по створенню проекту представлені в таблиці 8.12

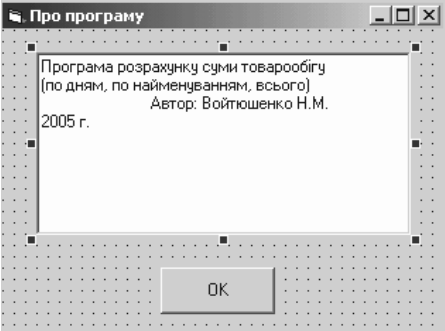
Таблиця 8.12. – Операції по виконанню приклада 8.13 «Складний цикл із меню»

№ п/п	Операція	Виконання
1	2	3
1	Створення електронної форми і завдання властивостей її елементів	Дії аналогічні попередньому прикладу в частині повторюваних елементів, ім'я форми – <b>Складний цикл</b> Імена текстових полів: <ul style="list-style-type: none"> <li>• для введення кількості днів – <math>n</math>,</li> <li>• найменувань товару – <math>m</math>,</li> </ul> для загальної суми – $S$ .
2	Створення меню й завдання властивостей його складовим	П.м. <b>Tools</b> → к. <b>Menu Editor</b> → у поле <b>Caption</b> набрати назва пункту меню <b>Про програму</b> , а в поле <b>Name</b> – ім'я пункту <b>Програма</b> (рисунок 8.20) → кнопка <b>Next</b> → у поле <b>Caption</b> набрати назву – <b>Уведення даних</b> , у поле <b>Name</b> – ім'я <b>Уведення</b> → зняти прапорець <b>Enabled</b> (пункт буде недоступна) → кнопка <b>Next</b> → у поле <b>Caption</b> набрати назву – <b>Аналітика</b> , у поле <b>Name</b> – <b>Аналітика</b> → зняти прапорець <b>Enabled</b> → кнопка <b>Next</b> → клацнути по кнопці зі стрілкою вправо, щоб зробити підменю → у поле <b>Caption</b> набрати назва – <b>Сума за день</b> , у поле <b>Name</b> – <b>За_день</b> → кнопка <b>Next</b> → клацнути по кнопці зі стрілкою вправо → у поле <b>Caption</b> набрати назву – <b>Сума за товар</b> , у поле <b>Name</b> – <b>За_товар</b> → кнопка <b>Next</b> → клацнути по кнопці зі стрілкою вліво → у поле <b>Caption</b> набрати назву – <b>Підсумковий розрахунок</b> , у поле <b>Name</b> – <b>Підсумок</b> → зняти прапорець <b>Enabled</b> → кнопка <b>Next</b> → у поле <b>Caption</b> набрати назву – <b>Вихід</b> , у поле <b>Name</b> – <b>Вихід</b> .

Продовження таблиці 8.12.

1	2	3
3	Оголошення глобальних перемінних	Відкрити вікно коду програми, викликати об'єкт <b>General</b> подія <b>Declaration</b> , увести наступні оператори: <b>Dim i, j As Integer</b> <b>Dim x() As Single</b> <b>Dim z As Single</b>
4	Перевірка доступності розрахункових	<b>Private Sub n_Change()</b> <b>If n.Text &lt;&gt; "" And m.Text &lt;&gt; "" Then</b> <b>Аналітика.Enabled = True</b>
	пунктів меню при заповненні текстового вікна <b>n</b>	<b>Уведення.Enabled = True</b> <b>Підсумок.Enabled = True</b> <b>End If</b> <b>End Sub</b>
5	Перевірка доступності розрахункових пунктів меню при заповненні текстового вікна <b>m</b>	<b>Private Sub m_Change()</b> <b>If n.Text &lt;&gt; "" And m.Text &lt;&gt; "" Then</b> <b>Аналітика.Enabled = True</b> <b>Уведення.Enabled = True</b> <b>Підсумок.Enabled = True</b> <b>End If</b> <b>End Sub</b>
6	Завдання функцій пункту меню <b>Вихід</b>	Подвійним клацанням по пункту <b>Вихід</b> викликати вікно коду і записати підпрограму: <b>Private Sub Вихід_Click()</b> <b>If MsgBox("Ви дійсно бажаєте вийти з програми?", vbYesNo) = vbYes_ Then End</b> <b>'значення vbYesNo див. у додатку Б; vbYes –</b> <b>'константа «Так»</b> <b>End Sub</b>
7	Завдання функцій пункту меню <b>Підсумковий розрахунок</b>	Подвійним клацанням по пункту <b>Підсумковий розрахунок</b> викликати вікно коду і записати підпрограму: <b>Private Sub Підсумок_Click()</b> <b>Form1.Hide</b> <b>S.Text = 0</b> <b>For i = 1 To n</b> <b>For j = 1 To m</b> <b>S.Text = Val(S.Text) + x(i, j)</b> <b>Next j</b> <b>Next i</b> <b>MsgBox ("Підсумкова сума товарообігу - "</b> <b>&amp; S.Text)</b> <b>Form1.Show</b> <b>End Sub</b>

Продовження таблиці 8.12.

1	2	3
8	Завдання функцій кнопки <b>Очищення</b>	<b>Private Sub Очищення_Click()</b> <b>n.Text = ""</b> <b>m.Text = ""</b> <b>S.Text = ""</b> <b>Аналітика.Enabled = False</b> <b>Уведення.Enabled = False</b> <b>Підсумок.Enabled = False</b> <b>End Sub</b>
9	Створення допоміжної форми для пункту меню <b>Про програму</b>	<p>П.м. <b>Project</b> → к. <b>Add Form</b> → дати ім'я формі <b>FПрограма</b> → у властивості <b>Caption</b> написати <b>Про програму</b> → у центрі форми</p>  <p>розмістити текстове вікно → задати для нього властивості: <b>Multiline – True</b> (для занесення багаторядкового тексту), <b>Enabled – False</b> (для неприступності тексту), <b>Text</b> – внести потрібний текст, перехід на новий рядок по клавіші <b>Enter</b> → унизу форми розмістити командну кнопку <b>OK</b></p>
10	Створення програмного коду для кнопки <b>OK</b> форми <b>Про програму</b>	<b>Private Sub Command1_Click()</b> <b>FПрограма.Hide</b> <b>Form1.Show</b> <b>End Sub</b>
11	Створення допоміжної форми для пункту меню <b>Уведення даних</b>	<p>П.м. <b>Project</b> → к. <b>Add Form</b> → дати ім'я формі <b>FУведення</b> → у властивості <b>Caption</b> написати <b>Уведення даних</b> → унизу форми розмістити командну кнопку <b>OK</b> → записати для кнопки <b>OK</b> код програми:</p> <b>Private Sub Command1_Click()</b> <b>FУведення.Hide</b> <b>Form1.Show</b> <b>End Sub</b>

Продовження таблиці 8.12.

1	2	3
12	Створення допоміжних форм для пунктів меню <b>Сума за день</b> і <b>Сума за товар</b>	П.м. <b>Project</b> → к. <b>Add Form</b> → дати ім'я формі <b>FЗа_день (Fза_товар)</b> → у властивості <b>Caption</b> написати <b>Продаж за день (За товар)</b> → унизу форми розмістити командну кнопку <b>OK</b> → записати для кнопки <b>OK</b> код програми: <b>Private Sub Command1_Click()</b> <b>Fза_день.Hide</b> <b>Form1.Show</b> <b>End Sub</b>  <b>(Private Sub Command1_Click)</b> <b>FЗа_товар.Hide</b> <b>Form1.Show</b> <b>End Sub)</b>
13	Завдання функцій пункту меню <b>Про програму</b>	Подвійним клацанням по пункту <b>Про програму</b> викликати вікно коду і записати підпрограму: <b>Private Sub Програма_Click()</b> <b>FПрограма.Show</b> <b>Form1.Hide</b> <b>End Sub</b>
14	Завдання функцій пункту меню <b>Уведення даних</b>	Подвійним клацанням по пункту <b>Уведення даних</b> викликати вікно коду і записати підпрограму: <b>Private Sub Уведення_Click()</b> <b>ReDim x(n, m)</b> <b>Form1.Hide</b> <b>FУведення.Show</b> <b>FУведення.Cls</b> <b>FУведення.Print "День ", "_ Найменування", "Сума"</b> <b>For i = 1 To n</b> <b>For j = 1 To m</b> <b>x(i, j) = InputBox("Уведіть суму _ продажу за " &amp; Str(i) &amp; " день по " &amp; Str(j) &amp; " найменування")</b> <b>FУведення.Print i; Tab(24); j; Tab(50); x(i, j)</b> <b>Next j</b> <b>Next i</b> <b>End Sub</b>
15	Завдання функцій пункту меню <b>Сума за день</b>	Розкрити п. Меню <b>Аналітика</b> і подвійним клацанням по пункту <b>Сума за день</b> викликати вікно коду і записати підпрограму:

Продовження таблиці 8.12.

1	2	3
		<pre> Private Sub За_день_Click()     Form1.Hide     Фза_день.Show     Фза_день.Cls     S.Text = 0     For i = 1 To Val(n.Text)         z = 0         For j = 1 To Val(m.Text)             z = z + x(i, j)         Next j         Фза_день.Print "Сума товарообігу за "; i; " _ день", z         S.Text = Val(S.Text) + z     Next i     Фза_день.Print „----- -----”     Фза_день.Print Tab(24); "Разом: "; Tab(43); S End Sub </pre>
16	<p>Завдання функцій пункту меню Сума за товар</p>	<p>Подвійним клацанням по пункту <b>Сума за товар</b> викликати вікно коду і записати підпрограму:</p> <pre> Private Sub За_товар_Click()     Form1.Hide     Фза_товар.Show     Фза_товар.Cls     S.Text = 0     For j = 1 To Val(m.Text)         z = 0         For i = 1 To Val(n.Text)             z = z + x(i, j)         Next i         Фза_товар.Print "Сума товарообігу_ по "; j; " найменуванню", z         S.Text = Val(S.Text) + z     Next j     Фза_товар.Print "----- -----"     Фза_товар.Print Tab(24); "Разом: ";_ Tab(57); S End Sub </pre>



Продовження таблиці 8.12.

1	2	3
17	Зберегти проект з ім'ям «Складний цикл з меню»	Створити папку з ім'ям «Складний цикл із меню» і зберегти в ній усі компоненти проекту.

У разі потреби висновку результатних даних на друк і на екран одночасно варто додати оператор:

**Printer.Print** "Сума товарообігу за "; i; " день", z  
після оператора виводу на допоміжну форму і відповідні команди друку після операторів виводу лінії і підсумкової суми.

Для додання проекту більш привабливого виду в найпростішому варіанті можна використовувати різні картинки, що з'являються при вказівці на визначені об'єкти форми. Для цього варто завантажувати у властивість **Picture** об'єкта потрібні картинки за допомогою функції:

**LoadPicture** (« повний шлях до графічного файлу»)

Але якщо вказати у виді константи повний шлях до файлу, то при переміщенні файлу з картинкою чи проекту буде порушена працездатність програми, тому раціональніше файли з картинками розмістити в папці проекту, увести перемінну, наприклад **Path**, для позначення шляху до графічного файлу і скористатися властивістю **App.Path** (об'єкт **App** – створюваний проект): **Path = App.Path**.

Приклад 8.14. – Змусьте завантажуватися відповідні картинки в створений об'єкт **Picture** при завантаженні форми, коли покажчик миші попадає на кнопку **Очищення**, на текстові вікна **n** і **m** (подія **MouseMove**). Приклади використовуваних вікон показані на рисунках 8.22 – 8.24.

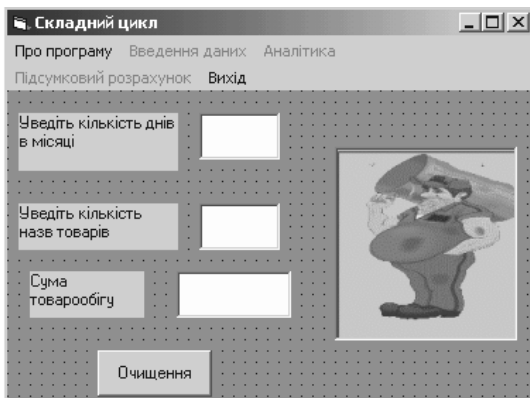


Рисунок 8.22 – Вид екранної форми при проходженні покажчика миші над кнопкою **Очищення**



Рисунок 8.23 – Вид екранної форми при проходженні покажчика миші над текстовим полем **Уведіть кількість найменувань товару**



Рисунок 8.24 – Вид екранної форми при проходженні покажчика миші над текстовим полем **Уведіть кількість днів у місяці**

### Виконання.

Таблиця 8.13 – Додаткові операції по завантаженню різних картинок до приклада 8.13 – «Складний цикл із меню»

№ п/п	Операція	Виконання
1	2	3
1	Оголошення глобальних перемінних	<b>Dim i, j As Integer</b> <b>Dim p() As Single</b> <b>Dim z As Single</b> <b>Dim Path, PName As String</b> 'нові
2	Занесення в перемінну шляху до папки з даним додатком	<b>Private Sub Form_Load()</b> <b>Path = App.Path</b> <b>End Sub</b>
3	Завантаження основної картинки при прямуванні покажчика миші за екранною формою	<b>Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)</b> <b>If PName &lt;&gt; "bd06135_.wmf" Then</b> <b>Picture1.Picture = LoadPicture(Path +</b> <b>"\bd06135_.wmf")</b> <b>PName = "bd06135_.wmf"</b> <b>End If</b> <b>End Sub</b>
4	Завантаження картинки при прямуванні покажчика миші над кнопкою <b>Очищення.</b>	<b>Private Sub Очищення_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)</b> <b>If PName &lt;&gt; "itrash.gif" Then</b> <b>Picture1.Picture = LoadPicture(Path +</b> <b>"\itrash.gif")</b> <b>PName = "itrash.gif"</b> <b>End If</b> <b>End Sub</b>
5	Завантаження картинки при прямуванні покажчика миші над текстовим вікном <b>n</b>	<b>Private Sub n_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)</b> <b>If PName &lt;&gt; "bs01072_.wmf" Then</b> <b>Picture1.Picture = LoadPicture(Path +</b> <b>"\bs01072_.wmf")</b> <b>PName = "bs01072_.wmf"</b> <b>End If</b> <b>End Sub</b>

Продовження таблиці 8.13.

1	2	3
6	Завантаження картинки при прямуванні покажчика миші над текстовим вікном <b>m</b>	<pre> Private Sub m_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)     If PName &lt;&gt; "so02253_.wmf" Then         Picture1.Picture = LoadPicture(Path + "so02253_.wmf")         PName = "so02253_.wmf"     End If End Sub </pre>

Примітка – Завантаження картинки при прямуванні покажчика миші над розглянутими в таблиці об'єктами буде йти безупинно, що виразиться в мельканні картинки. Щоб уникнути цього, картинка вантажиться один раз, а потім просто перевіряється її наявність оператором **IF**.

### Питання й завдання для самостійного виконання

1. У чому розходження в написанні програм для арифметичних і ітераційних циклів?
2. Як працює оператор для запису арифметичних циклів?
3. Які оператори для запису ітераційних циклів Ви знаєте, у чому їхнє розходження?
4. Що нового з'являється в програмах із складними циклами в порівнянні з програмами простих циклічних процесів?
5. Як додати нові форми в проект і для чого їх можна використовувати?
6. Що дає використання меню в програмі?
7. Що потрібно, щоб використовувати ті самі перемінні в різних підпрограмах?
8. Написати проект для обчислення  $y = n!$  Для вибору значення **n** використовувати лінійку прокручування.
9. Скласти програму обчислення по формулі  $F = \sum i^2$ , де  $i = 1+n$ , вивести значення **i**,  $i^2$ , **F**.
10. Написати проект завантаження масиву чисел і вибору з нього максимального або мінімального значення (селекторні кнопки) і вибору можливості розрахунку середнього введених чисел (прапорець).
11. Розробити проект обчислення Суми продажу товарів по кожному найменуванню товару, загальну суму продажу, мінімальну, максимальну і середню ціну по проданих товарах, якщо відомі кількість (**Ki**) і ціна (**Ci**) по кожному найменуванню товару.

12. Змінити проект створений по п.11 в) параграфа 3.4, додавши в його умову необхідність нагромадження даних по всіх покупцях а) у загальній сумі, б) по окремих видах блюд (створення меню).
13. Як видати результати розрахунків на екран, на печатку, одночасно на екран і печатку?
14. Які функції використовуються при зміні картинок в об'єкті Picture?
15. Додайте у свої проекти змінювані картинки придатні за змістом.

## 8.8 Робота з файлами

У реальних задачах більшість даних зберігається у файлах і зчитується з файлів. Традиційний підхід при роботі з файлами полягає у використанні функцій і операторів, що забезпечують прямий доступ до інформації у файлах.

У Visual Basic виділяються наступні типи файлів – файли послідовного доступу (текстові чи файли аналогічні їм; структурна одиниця – рядок), файли довільного доступу (файли баз даних; структурна одиниця – запис) і двійкові файли (файли з побайтним доступом). Подібний розподіл файлів на типи визначається особливостями організації файлів і доступу до даних у них. Робота з кожним типом файлів має свої особливості. Для приклада буде розглянута робота з файлами довільного доступу.

Файл довільного доступу має заздалегідь задану структуру і складається із записів – порції даних строго визначеного розміру, що мають конкретний номер у файлі. Доступ до даних у файлі здійснюється саме по номеру запису. Користувач може сам задавати структуру запису файлу. Для цього варто додати програмний модуль до проекту й у ньому за допомогою оператора **Type** оголосити тип перемінної необхідної структури.

Додавання програмного модуля виконується за наступною схемою:

меню **Project** → команда **Add module** → вкладка **New** → значок

**Module** → кнопка **Відкрити**.

Формат оператора **Type**:

**Type** <ім'я запису>

<ім'я поля> **As** <тип перемінної >

.....

<ім'я поля> **As** <тип перемінної >

**End Type**

Для символічних перемінних необхідно задавати довжину в символах (байтах), наприклад: **FIO as String\*50**, де 50 – загальна довжина П.І.Б.

**Приклад 8.15** – Оголосити перемінні про адресу клієнта: поштовий індекс (5 зн.), країна (15 зн.), місто (район) (25 зн.), населений пункт (20 зн.), вулиця (20 зн.), будинок (5зн.), квартира (3зн.).

**Виконання.**

**M. Project** → к. **Add module** → вкл. **New** → значок **Module** → кнопка **Відкрити**.

Набрати наступний оператор:

```

Type Adres
  Indeks As Integer
  Country As String*15
  Region As String*25
  Punkt As String*20
  Street As String*20
  House As String*5
  Flat As Integer

```

**End Type**

При роботі з файлами довільного доступу використовуються наступні основні оператори й функції:

1. Оператор **відкриття** файлу.

**Open Fname [ For Random] as Fnum Len = Dlina**

Де: **Fname** – повне ім'я файлу в лапах;

**[ For Random]** – режим довільного доступу – використовується за замовчуванням;

**Fnum** – номер файлу, наприклад, 1 або #1;

**Dlina** – довжина запису в байтах. Якщо довжина запису не відома, її можна обчислити за допомогою функції **Len** (перемінна для збереження запису).

2. Оператор **запису** у файл.

**Put Fnum, [Zapnum], Pzap**

Де: **Fnum** – номер файлу, зазначений в операторі **Open**;

**[Zapnum]** – перемінна чи вираження, що задають номер запису у файлі;

**Pzap** – буферна перемінна, що вказує джерело записуваних даних. За структурою вона повинна відповідати структурі запису файлу і може з'являтися за допомогою перемінною користувальницького типу. У неї заноситися запис перед виконанням оператора **Put**.

### 3. Оператор читання з файлу.

**Get Fnum, [Zapnum], Pzap ,**

де параметри аналогічні оператору **Put**. У цьому операторі в буферну перемінну читається запис перш, ніж із нею почнуть працювати. Схема роботи буферної перемінної представлена на рисунку 8.25.

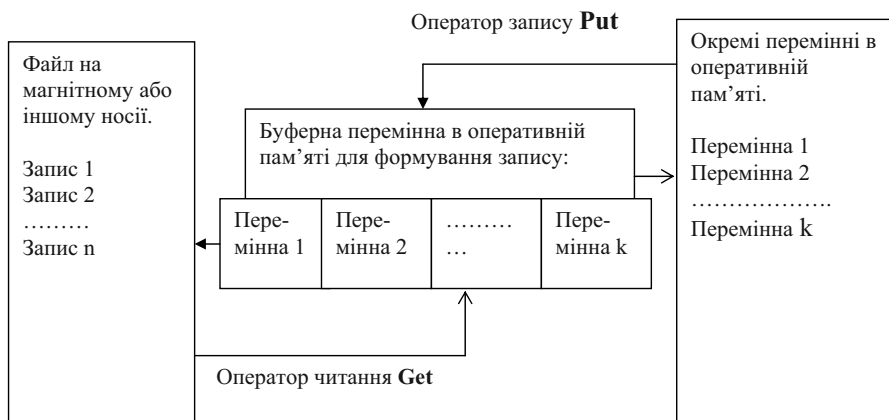


Рисунок 8.25 – Схема роботи буферної перемінної запису файлу

### 4. Оператор закриття файлу.

**Close [ список Fnum]**

Де: [ **список Fnum** ]- список номерів файлів, що закриваються. Якщо номер не зазначений, то закриваються усі відкриті файли.

### 5. Функція, що повертає розмір відкритого файлу

**Lof(<номер файлу>)**

## 6. Функція видалення файлу

### Kill(<номер файлу>)

Приклад 8.16. – Створити проект запису й читання даних про товари у файл довільного доступу з ім'ям **Tovar**. Структура запису повинна бути наступної: код товару (ім'я **Kod**) – 5 цифр, найменування товару (**Nam**) – 30 символів, одиниці виміру (**Ed**) – 20 символів, кількість (**K**) і ціна (**C**) – по 10 цифр із крапкою.

При введенні ціни і кількості товару після натискання клавіші **Enter** повинна підраховуватися сума (**Sum**) по введеному найменуванню, а по завершенню запису всіх даних необхідно порахувати загальну суму виторгу (**Всего**).

Виконання. При занесенні ціни в текстове вікно відразу буде підраховуватися сума товару. Запис і читання даних із файлу буде здійснюватися при клацанні миші по відповідній кнопці на формі (рисунок 8.26). Підрахунок загальної суми буде виконуватися при клацанні по кнопці **Розрахунок**. Схема рішення приклада показана на рисунку 8.27, операції по виконанню – у таблиці 8.14.

Код товару	<input type="text"/>	Сума виторгу за день	<input type="text"/>	Запис
Найменування товару	<input type="text"/>			Читання
Одиниця виміру	<input type="text"/>			Розрахунок
Кількість	<input type="text"/>	Сума	<input type="text"/>	Очищення
Ціна	<input type="text"/>			Вихід

Рисунок 8.26 – Електронна форма до приклада 8.16 «Робота з файлом»



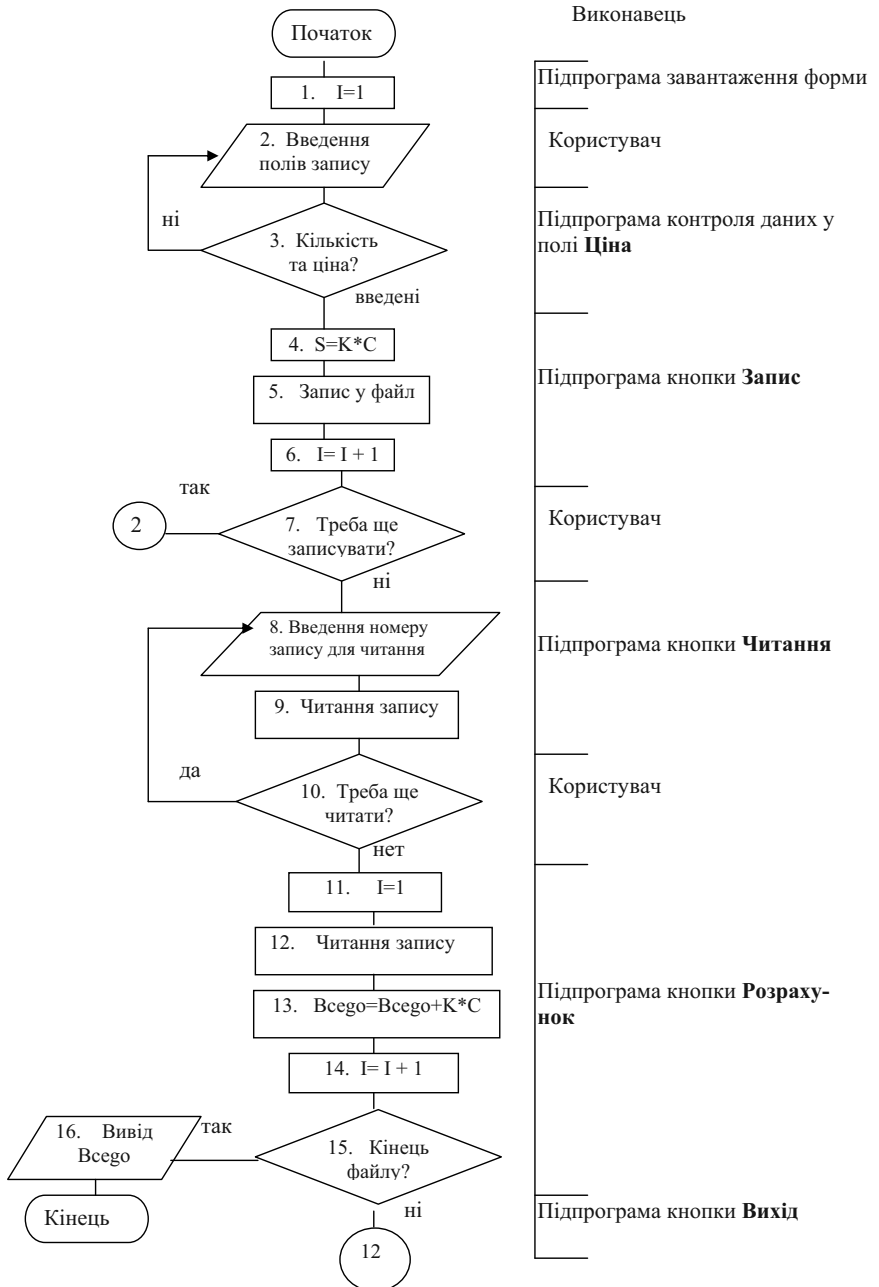


Рисунок 8.27 – Схема рішення приклада 8.15

Таблиця 8.14. – Операції по виконанню приклада 8.16 «Робота з фай-  
ЛОМ»

№ п/п	Операція	Виконання
1	2	3
1	Створення електронної форми і завдання властивостей її елементів	Дії аналогічні попереднім прикладам (рисунок 8.26).
2	Завдання функції кнопки <b>Вихід</b>	Установити об'єкт <b>Вихід</b> і для події <b>Click</b> записати: <b>Private Sub Vixod_Click()</b> <b>End</b> <b>End Sub</b>
3	Завдання функцій кнопки <b>Очищення</b>	Установити об'єкт <b>Очищення</b> (властивість <b>Name</b> має ім'я <b>Очищення</b> ) і для події <b>Click</b> записати: <b>Private Sub Очищення_Click()</b> <b>Kod.Text = " "</b> <b>Nam.Text = " "</b> <b>Ed.Text = " "</b> <b>K.Text = " "</b> <b>C.Text = " "</b> <b>Mes.Caption = " "</b> <b>Sum.Text = " "</b> <b>Vcego.Text = " "</b> <b>End Sub</b>
4	Контроль уведення даних у поле <b>Kod</b>	Установити для текстового поля <b>Kod</b> подію <b>KeyPress</b> , увести текст підпрограми: <b>Private Sub Kod_KeyPress(KeyAscii As Integer)</b> <b>Select Case KeyAscii</b> <b>Case 0, 8, 13, 48 To 57</b> <b>Case Else</b> <b>KeyAscii = 0</b> <b>Mes.Caption = "Помилка! Уведіть_ціле 5 – значне число"</b> <b>End Select</b> <b>End Sub</b>
5	Контроль уведення даних у поле <b>C</b>	<b>Private Sub C_KeyPress(KeyAscii As Integer)</b> <b>Select Case KeyAscii</b> <b>Case 0, 8, 46, 48 To 57</b>

Продовження таблиці 8.14.

1	2	3
		<pre> <b>Case 13</b> <b>If K.Text&lt;&gt;"" Then</b>   <b>If C.Text&lt;&gt;"" Then</b>     <b>Sum.Text = Val(K.Text) * Val(C.Text)</b>   <b>Else</b>     <b>Mes.Caption = "Помилка! Уведіть ці-</b> <b>ну"</b>   <b>End If</b> <b>Else</b>   <b>Mes.Caption="Помилка! Уведіть _</b> <b>кількість"</b> <b>End If</b> <b>Case Else</b>   <b>KeyAscii = 0</b>   <b>Mes.Caption = "Помилка! Уведіть _</b> <b>число"</b> <b>End Select</b> </pre>
6	Контроль уведення даних у поле <b>К</b>	<pre> <b>Private Sub K_KeyPress(KeyAscii As</b> <b>Integer)</b> <b>Select Case KeyAscii</b>   <b>Case 0, 8, 13, 46, 48 To 57</b>   <b>Case Else</b>     <b>KeyAscii = 0</b>     <b>Mes.Caption = "Помилка! Уведіть _</b> <b>число"</b> <b>End Select</b> <b>End Sub</b> </pre>
7	Оголошення користувальницького типу запису	<p>Меню <b>Project</b> → команда <b>Add module</b> → вкладка <b>New</b> → значок <b>Module</b> → кнопка <b>Відкрити</b></p> <p>Записати</p> <pre> <b>Type zapic</b> <b>Kod As Long</b> <b>Nam As String * 30</b> <b>Ed As String * 20</b> <b>K As Single</b> <b>C As Single</b> <b>End Type</b> </pre> <p>Зберегти модуль з ім'ям «Робота з файлом»</p>

Продовження таблиці 8.14.

1	2	3
8	Оголошення буферної перемінної для запису файлу й лічильника записів	У зв'язку з тим, що перемінні <b>ZapicBox</b> і <b>I</b> використовуються в декількох модулях, то раціональніше їх оголосити однократно в глобальному модулі: у вікні коду проекту установити об'єкт <b>General</b> і подія <b>Declaration</b> , потім записати <b>Dim ZapicBox As zapic</b> <b>Dim i As Integer</b>
9	Установка початкового значення лічильника записів файлу (при завантаженні форми)	<b>Private Sub Form_Load()</b> <b>i = 1</b> <b>End Sub</b>
10	Програмний код кнопки <b>Записати</b>	<b>Private Sub Zap_Click()</b> <b>Open "D:\Tovar" For Random As #1</b> <b>Len_ = Len (ZapicBox)</b> <b>ZapicBox.Kod = Kod.Text</b> <b>ZapicBox.Nam = Nam.Text</b> <b>ZapicBox.Ed = Ed.Text</b> <b>ZapicBox.K = Val(K.Text)</b> <b>ZapicBox.C = Val(C.Text)</b> <b>Put #1, i, ZapicBox</b> <b>i = i + 1</b> <b>Close #1</b> <b>End Sub</b>
11	Програмний код кнопки <b>Читати</b>	<b>Private Sub Read_Click()</b> <b>Open "D:\Tovar" For Random As #1_</b> <b>Len = Len(ZapicBox)</b> <b>i = InputBox("Уведіть номер_ потрібної_ запису")</b> <b>Get #1, i, ZapicBox</b> <b>Kod.Text = ZapicBox.Kod</b> <b>Nam.Text = ZapicBox.Nam</b> <b>Ed.Text = ZapicBox.Ed</b> <b>K.Text = ZapicBox.K</b> <b>C.Text = ZapicBox.C</b> <b>Close #1</b> <b>End Sub</b>
12	Програмний код кнопки <b>Розрахунок</b>	<b>Private Sub Розрахунок_Click()</b> <b>Open "D:\Tovar" For Random As #1_</b>

Продовження таблиці 8.14.

1	2	3
		<b>Len = Len(ZapicBox)</b> <b>Всего = 0</b> <b>For i = 1 To Lof(1)</b> <b>Get #1, i, ZapicBox</b> <b>Всего.Text = Val(Всего.Text) +</b> <b>ZapicBox.К * ZapicBox.С</b> <b>Next i</b> <b>Close #1</b> <b>End Sub</b>
13	Зберегти проект з ім'ям «Робота з файлом»	Меню <b>File</b> → команда <b>Save Project</b> → задати ім'я файлу й папку.
14	Налагодити програму	Меню <b>Run</b> → команда <b>Start</b> , увести кілька записів, уведення кожної завершувати клацанням по кнопці <b>Запис</b> . Потім прочитати записані дані в будь-якому порядку, попередньо клацаючи по кнопці <b>Читання</b> . При виявленні помилок їх варто виправити. Виконати обчислення показника <b>Сума виторгу за день</b> .
15	Налагоджену програму зберегти	Меню <b>File</b> → команда <b>Save Project</b> .

### Питання й завдання для самостійного виконання

1. Які типи файлів створюються й обробляються в Visual Basic ? Для чого вони ?
2. Що таке запис і як можна задати його структуру ?
3. Назвіть основні оператори для роботи з файлами.
4. Які функції для роботи з файлами Ви знаєте ?
5. Де з'являються перемінні, котрі використовуються в декількох модулях програми ?
6. Для чого в програмі використовується буферна перемінна запису файлу?
7. Написати програму створення файлу **Підсумки сесії**. Структура запису файлу: факультет, група, П.І.Б студента, назва предмета, вибір для предмета форми контролю – залік або іспит, оцінка про здачу заліку або оцінка іспиту. Програма повинна дозволяти читати дані по будь-якому студенту, пі-

- драховувати середній бал групи по предмету і по всіх іспитах. Довжину текстових перемінних визначити самостійно.
8. Додати в програму п.7 можливість відбирати студентів-боржників.

## **Література**

1. Ананьев А., Федоров А. Самоучитель Visual Basic 6.0 – СПб: БХВ – Санкт-Петербург, 2000 – 624с.
2. Войтюшенко Н.Н., Остапец А.И. Компьютерная техника и программирование. Часть 2. Основы алгоритмизации и программирования на Visual Basic: Учебное пособие по базовой подготовке для студентов дневной и заочной форм обучения. – Донецк: ДонГУЭТ, 2002. – 150 с.
3. Маккелви Майкл VISIAL BASIC без проблем. – М.: БИНОМ, 1996
4. Справочная система пакета Visual Basic.

## Розділ 9. Створення баз даних засобами MS Access

### 9.1 Призначення й загальні принципи побудови СУБД

Традиційно, поняття **база даних** означає набір взаємозалежних даних, що зберігаються в організованій формі. Створення й роботу з базами даних здійснюють за допомогою спеціальних програм – систем управління базами даних (СУБД). На даний період на програмному ринку налічується близько 50 різних СУБД, які можна класифікувати по наступних ознаках:

**по призначенню й областям застосування** – спеціалізовані (наприклад, для збереження графічної геофізичної інформації) і загального застосування, які призначені для рішення широкого кола завдань (наприклад, ведення складського обліку, торговельних операцій, обліку кадрів і ін.).

**по складності й функціональних можливостях** – СУБД для персонального використання або настільні (наприклад, *Paradox*, *Access*, *FoxPro* і ін.), мережні СУБД, які функціонують у режимі клієнт-сервер (наприклад, *Oracle*, *Informix* і *Ingres*).

СУБД для персонального використання працюють на одному ПК або в локальній мережі. При цьому використання загальної для багатьох користувачів бази даних практично неможливе. СУБД, які функціонують у режимі клієнт-сервер, дозволяють працювати з єдиною, спільною для всіх користувачів БД, що перебуває на файлі-сервері. Сервер також здійснює обробку запитів до БД, які надходять від клієнтів і передає їм готові результати. Така технологія дозволяє значно зменшити потік даних, які проходять по мережі.

В основі СУБД лежить концепція моделі даних, тобто деякої абстракції подання даних. У більшості випадків передбачається, що дані представлені у вигляді файлів, що складаються із **записів**. Структура всіх записів у файлі однакова, а кількість записів у файлі є змінним.

Елементи даних, із яких складається кожний запис, називаються **полями**. Оскільки у всіх записах є ті самі поля (із різними значеннями), полям зручно дати унікальні імена. Багато практично важливих випадків добре укладаються в таке представлення даних.

Наприклад, у відділі кадрів інформація про співробітників має таку природу: співробітників приймають на роботу й звільняють, але форма особового листка по обліку кадрів залишається незмінною для кожного співробітника. Товарно-матеріальні цінності приходять і йдуть, але форма інвентарної картки також залишається незмінною. Число прикладів легко можна помножити, але вже ясно, що СУБД є підходящим засобом у всіх випадках, коли вхідну інформацію можна представити у вигляді таблиці постійної структури, але невизначеної довжини або у вигляді картотеки, що містить невизначену кількість карток постійної структури. Усі СУБД підтримують у тій або іншій формі чотири основних операції:

- додати у базу даних один або кілька записів;
- видалити з бази даних один або кілька записів;
- знайти в базі даних один або кілька записів, що задовольняють заданій умові;
- оновити в базі даних значення деяких полів в одному або декількох записах.

Більшість СУБД підтримує, крім того, який-небудь механізм зв'язків між різними файлами, що входять у базу. Залежно від способу встановлення зв'язків у базі даних вони розділяються на: **ієрархічні, мережні й реляційні**.

**Ієрархічна структура (модель)** будується у вигляді ієрархічної деревоподібної структури, у якій для кожного головного об'єкта існує кілька підлеглих, а для кожного підлеглого об'єкта може бути тільки один головний. На найвищому рівні ієрархії перебуває кореневий об'єкт. Прикладом ієрархічної структури даних може бути організація каталогів на диску, різного роду класифікації, структура державної влади й т.п..

У **мережній моделі** один і той самий об'єкт може одночасно виступати як у ролі головного, так і підлеглого елемента. Це означає, що кожний об'єкт може брати участь у довільній кількості зв'язків. Зв'язок у цьому випадку може встановлюватися явно, коли значенням деяких полів є посилання на дані, що утримуються в іншому файлі. Прикладом мережної структури БД може бути структура автобусних маршрутів (із будь-якого населеного пункту існують маршрути в інші).

Ієрархічна й мережна моделі даних стали застосовуватися в системах управління базами даних на початку 60-х років.



У **реляційній моделі** дані й взаємозв'язки між ними представляються за допомогою прямокутних таблиць. Рядки в **реляційній** базі даних називають *записами*, а стовпці – *полями*. Модель реляційної бази даних була вперше розроблена доктором Э. Ф. Коддом на початку 70-их ХХ ст. як більш зручний засіб збереження, вибірки й маніпулювання даними, чим ієрархічні й мережні бази даних. Модель двомірної таблиці дозволяє звертатися до даних як по рядках, так і по стовпцях, що є значною перевагою.

Ті програми управління базами даних, у яких зв'язування таблиць не передбачено, називаються процесорами плоских файлів. Вони використовуються в основному для ведення найпростіших баз, таких як список прізвищ, адрес і телефонів, розкладів поїздів і т.п.

У більш складних і об'ємних базах даних зв'язок між таблицями встановлюється неявним образом – по збігу значень полів у різних таблицях.

Наприклад, якщо на підприємстві створюється база даних для обліку замовлень, то в неї необхідно включити дані про клієнтів, про товари, про замовлення конкретним клієнтом необхідних товарів. Користувач не повинен вводити реквізити клієнтів більше одного разу. Кожному клієнтові потрібно привласнити унікальний код, всю інформацію про нього разом із кодом помістити в окрему таблицю й, щоб указати, яким клієнтом зроблено замовлення, користуватися кодом клієнта. Подібним же чином інформація про товари повинна бути винесена в окрему таблицю, де кожний товар описаний тільки один раз. Тоді запис у таблиці замовлень може складатися з номера замовлення, дати замовлення, коду клієнта, коду товару, його кількості й ціни. При такій схемі зберігання інформації уведення даних про замовлення значно спрощується. А при виводі інформації про замовлення на екран до записів таблиці замовлень приєднується інформація з таблиць клієнтів і товарів, здійснюється так назване об'єднання таблиць. Таблиці замовлень, товарів і клієнтів зв'язані між собою за допомогою кодів. Коди ці унікальні, завдяки чому по коду клієнта можна відразу знайти запис про нього в таблиці клієнтів, а по коду товару – запис у таблиці всіх товарів. Отримана в результаті об'єднана віртуальна таблиця містить повну інформацію про замовлення, зібрану з декількох вхідних таблиць.

Перевагою **реляційної** моделі є відносна простота інструментальних засобів її підтримки. Завдяки відсутності дублювання даних, для

реляційних СУБД значно знижуються вимоги до пам'яті й дискового простору.

Усі настільні реляційні СУБД дозволяють користувачеві вводити, редагувати, переглядати й роздруковувати інформацію, що втримується в одній або декількох таблицях. У цьому змісті вони мало чим відрізняються від звичайних електронних таблиць. Щоб усе-таки визначити різницю між електронними таблицями й реляційними СУБД, варто звернути увагу на три основних моменти.

- Усі реляційні СУБД призначені для обробки дуже великих обсягів даних – набагато більших чим ті, якими без відчутних незручностей для користувача можуть оперувати електронні таблиці.

- У реляційних СУБД таблиці можна без зусиль зв'язувати, представляючи їхні дані у вигляді єдиної таблиці. Користуючись електронними таблицями, зробити це найчастіше нелегко, якщо взагалі можливо.

- У реляційних СУБД дублювання інформації зведене до мінімуму. Практично повторюються тільки коди, що зв'язують дані декількох таблиць. У таблицях СУБД не утримуються дані, що обчислюються, тоді як табличні процесори працюють із таблицями, які обчислюються.

У якому ж випадку використовувати СУБД? Якщо інформація перебуває в одній таблиці, що складається з невеликої кількості полів (10-12), а кількість записів не перевищує 25-30, то можна скористатися програмами Word або Excel. Якщо даних багато, використовуються вони багатьма користувачами й (або) для різних цілей, потрібен строгий контроль за схоронністю даних – у кожному з цих випадків необхідно використовувати базу даних і відповідну СУБД.

*Microsoft Access* – система управління реляційною базою даних – складова частина пакета Microsoft Office. *Microsoft Access* має розвинуту систему розробки додатків для Windows, надає широкі можливості для роботи з даними з інших джерел, включаючи найбільш популярні СУБД для ПК (наприклад, *dBase*, *Paradox*, *FoxPro*, інші бази *Access*) і бази даних *Oracle*, *SQL*, що перебувають на серверах, мініЕОМ і великих центральних ЕОМ. *Access* може обмінюватися даними з іншими додатками Microsoft Office – *Word*, *Excel*, *Power Point*, *Outlook*. *Access 2000* містить новий засіб – сторінки доступу до даних, що дозволяє легко створювати додатки для роботи в інтрамережі.

## 9.2 Створення бази даних

Сучасні технології дозволяють кваліфікованому користувачеві швидко створювати досить складні додатки. Однак, якщо користувач перед цим не витратить певних зусиль на визначення завдань і принципів роботи додатка, то згодом йому доведеться втратити значно більше часу на всілякі переробки. Тому перед безпосередньою роботою з *Microsoft Access* варто виконати ряд етапів.

Створення бази даних починається з визначення цілей і завдань, які дана база буде забезпечувати на сьогоднішній день і в майбутньому.

Варто відповісти на питання:

- Для чого Вам потрібна база даних?
- Які звіти, результатну інформацію Ви хочете з неї одержувати?

Якщо завдань багато, необхідно впорядкувати їх за часом виконання. Таке впорядкування допоможе при зв'язуванні таблиць, побудові запитів і звітів.

Для кожної з виділених і впорядкованих задач необхідно скласти докладний перелік даних і визначити їхній вид: вхідні або ті, що обчислюються; постійні, що модифікуються у процесі рішення задачі, що видаються, що додаються.

Проектування таблиць бази даних здійснюється так, щоб останні не містили дублюючих даних за винятком сполучних кодів. Для кожної таблиці варто визначити структуру запису (перелік полів, тип записуваної у них інформації), ключові поля і поля для зв'язку з іншими таблицями. Процес організації даних у таблиці називається нормалізацією. Чотири основні правила нормалізації [1]:

1. Кожне поле таблиці повинне представляти унікальний (не повторюваний) тип інформації.
2. Кожна таблиця повинна мати унікальний *первинний ключ*, що складається з одного або декількох полів. *Ключем* називається поле, значення якого використовується для пошуку певного запису. Значення ключа для різних записів різне й строго унікальне.
3. Для кожного значення первинного ключа значення в стовпцях даних повинні відноситися до об'єкта таблиці й повністю його описувати.
4. Необхідно мати можливість змінювати значення кожного поля, яке входить у первинний ключ, без впливу на дані інших полів.

Для одержання необхідних результатів варто продумати зв'язки між таблицями, тому що зміна вже створених зв'язків у працюючій базі даних може привести до перепроєктування останньої.

Відповідно до цілей рішення задач створити необхідні форми, запити, звіти.

**Приклад 9.1.** – Розробити для підприємства базу даних **Замовлення** з метою поліпшення роботи з клієнтами й підвищення прибутку від діяльності. База даних повинна забезпечити фахівців підприємства інформацією про поточний стан виконання замовлень клієнтів, даними для аналізу замовлень в асортиментному й ціновому розрізах по групах клієнтів і тимчасовим періодам.

**Виконання.** Для створення початкового фрагмента бази як мінімум необхідні відомості:

- про клієнтів,
- про товари й
- про замовлення.

Усі ці відомості можна представити у вигляді відповідних таблиць.

Таблиця даних про клієнтів – **Довідник клієнтів** – повинен містити код клієнта, його найменування, адресу, МФО банку, найменування банку клієнта, розрахунковий рахунок.

Таблиця даних про товари – **Довідник товарів** – повинен містити код товару, його найменування, одиниці виміру, ціну.

Таблиця замовлень – **Замовлення** – повинна містити номер і дату замовлення, код клієнта, код товару, кількість, дату виконання замовлення.

Структура запису й типи даних у них показані на рисунку 9.1.

Таблиці не повинні містити даних, що розраховані! Кожен запис таблиці (рядок) повністю описує об'єкт таблиці – клієнта або товар, або замовлення. Зміна кожного поля таблиці, наприклад, адреси клієнта, не зачіпає інші поля даної таблиці й, тим більше, інших таблиць. Кожен запис має унікальний код (ключове поле) – код клієнта або товару, або замовлення.

Усі таблиці пов'язані за допомогою кодів полів. Таблиця **Замовлення** зв'язана з **Довідником клієнтів** по коду клієнта, а з **Довідником товарів** по коду товару.

Таким чином, не можна сказати, що в базі даних узагалі немає дублювання інформації. Воно є, але мінімально необхідне – повторюються тільки коди полів, по яких зв'язуються різні таблиці.

Для створення своєї бази даних, роботи з нею або з готовими прикладами, необхідно запустити *Access*.

Кнопка **Пуск** → п. **Програми** → п/п **Microsoft Access**

При запуску *Access* відкривається діалогове вікно (рисунок 9.2). Для визначення режиму роботи (необхідно встановити потрібний індикатор вибору: відкрити або створити базу даних через майстра або безпосередньо вручну).

Якщо буде обраний майстер для створення бази даних, то він запропонує цілий ряд заготовок різних баз даних (Студенти й заняття, Склад, Замовлення на роботи, Особисте майно й т.д.). У заготівках пропонується відразу ряд таблиць із пойменованими полями, які можна змінювати й додавати. Однак для внесення серйозних змін у базу даних потрібні певні знання й навички, які можна одержати при створенні бази даних із нуля.

### Довідник клієнтів

Код клієнта	Найменування клієнта	Адреса клієнта	МФО банку клієнта	Найменування банку	Розрахунковий рахунок
5 символ, числовий, ключове поле	30 символів, текстове	60 символів, текстове	6 символів, числове	30 символів, текстове	12 символів, числове

### Таблиця Замовлення

Номер замовлення	Дата замовлення	Код клієнта	Код товару	Кількість	Дата виконання
5 символів, числовий, ключове поле	Тип даних – Дата	5 символів, числовий, ключове поле	7 символів, числовий, ключове поле	10 символів, числове	Тип даних – Дата

### Довідник товарів

Код товару	Найменування товару	Одиниці виміру	Ціна
7 символів, числовий, ключове поле	30 символів, текстове	10 символів, текстове	10 символів, числове

Рисунок 9.1 – Форма й зв'язки таблиць фрагмента бази дані **Замовлення**

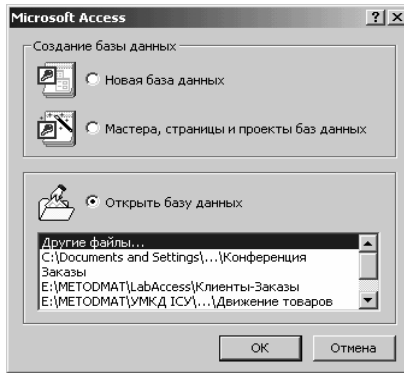


Рисунок 9.2 – Діалогове вікно визначення режимів роботи

Якщо буде обраний параметр **Новая база данных**, то необхідно в наступному діалоговому вікні задати ім'я бази даних, наприклад, **Замовлення** (автоматично буде додане розширення **.mdb**) і клацнути по кнопці **Создать**. Завантажується первинне вікно програми *Access* з усіма властивими йому атрибутами. Як уже говорилося раніше, усі однакові операції для додатків пакета Microsoft Office згруповані в однойменних пунктах меню: **Файл, Правка, Вид, Вставка, Сервис, Окно, Справка**.

У первинному вікні програми відкриється вікно бази даних, у якому будуть створюватися всі інші об'єкти (рисунок 9.3).

У файлі бази даних *Access* можуть зберігатися об'єкти семи основних типів. Разом ці об'єкти виконують усі основні функції СУБД і становлять закінчений додаток.

- **Таблиці** містять елементи даних у табличному форматі (рядок-стовпець), подібному тому, що використовується в електронних таблицях. База даних *Access* може включати до 32768 об'єктів (загальна кількість таблиць, форм, звітів, запитів і т.д.), при цьому одночасно можна відкрити до 254 таблиць, при наявності достатніх ресурсів.

Таблиці можна імпортувати з баз даних інших додатків (таких як *xBase, Paradox*), баз даних архітектури клієнт-сервер (таких як *Microsoft SQL Server*) або з електронних таблиць (наприклад, *Microsoft Excel* і *Lotus 1-2-3*). Бази даних *Access* можна зв'язувати з таблицями баз даних інших типів (*DBase, FoxPro, Paradox*) форматованими файлами (такими, як текстові файли у форматі ASCII і робочі листи *Excel*) і з іншими базами даних *Access*.

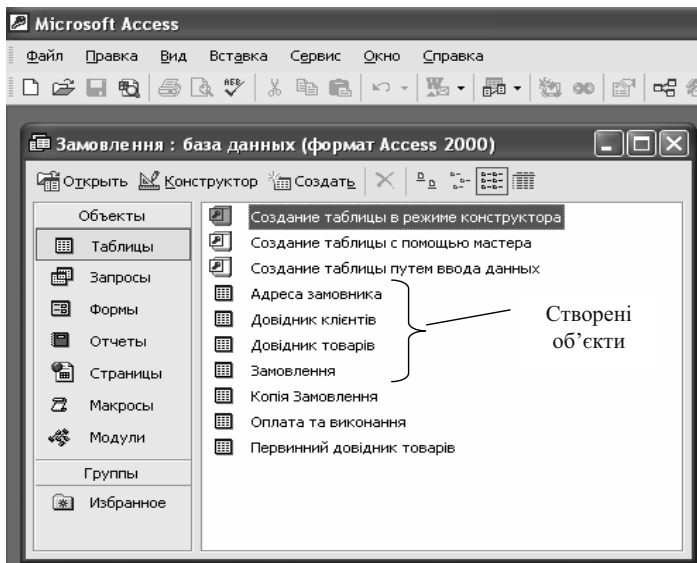


Рисунок 9.3 – Вікно *Access* і бази даних **Замовлення**

- **Запити** зв'язують між собою дані декількох таблиць (до 16 таблиць) і відображують їх у вигляді деякої віртуальної таблиці. Крім того, за допомогою запитів можна вказати, які саме поля обраних таблиць потрібно відібрати (усього до 255 полів) і як на їхній підставі сформулювати поля віртуальної таблиці. Критерії відбору визначають не тільки поля, але й рядки даних, що відбираються.

- **Форми** відображають дані, що утримуються в таблицях і запитах, і дозволяють додавати, видаляти або редагувати інформацію. У форми можна вставляти малюнки й діаграми, а при наявності звукової карти ще й коментарі й музику. У форми можна також уставляти командні кнопки, прапорці й інші керуючі елементи й зв'язувати з ними макроси або процедури Visual Basic.

- **Звіти** дають можливість форматувати, обчислювати підсумки, друкувати дані таблиць і запитів практично в будь-якому форматі. *Access* дозволяє включати у звіт графічні об'єкти, завдяки чому можна, наприклад, роздрукувати повний ілюстрований каталог продукції фірми.

• **Сторінки доступу до даних** – об'єкти, що містять файл HTML і допоміжні файли, які забезпечують доступ до даних із Internet Explorer. Опублікування сторінок доступу до даних, наприклад, у корпоративній комп'ютерній мережі, дозволить іншим користувачам знаходити, переглядати й змінювати дані (для Office 2000 і вище).

• **Макроси** – дії користувача автоматично записані комп'ютером і привласнені кнопці або сполученню клавіш. Макроси можна використовувати для відкриття таблиць, виконання запитів, перегляду й друку звітів, перевірки значення поля при зміні його вмісту й ін. З макросу можна запустити інший макрос або процедуру Visual Basic.

• **Модулі** являють собою процедури, написані мовою програмування VBA, які служать для різних цілей. Наприклад, процедури зв'язуються з певними подіями й виконуються при їх настанні. Прикладом подібної події може бути клацання миші на певній кнопці форми. Крім того, можна створювати власні функції для використання їх у формах, звітах, запитах і т.д. Використовуючи такі процедури й функції у базах даних, можна створювати закінчені додатки з власними меню й панелями інструментів.

У кожен даний момент часу може бути відкрита тільки одна база даних. Це обмеження не є принциповим, тому що відкрита база даних може використовувати таблиці з інших баз даних.

### 9.2.1 Таблиці

Першим і головним питанням, що підлягає рішенню при створенні будь-якого додатка, заснованого на деякій СУБД, є проектування бази даних. Це припускає визначення того, які таблиці будуть входити в базу, які в них будуть поля, як будуть зв'язані таблиці, яка інформація буде зберігатися в таблицях бази в готовому виді, а яка інформація буде обчислюватися за допомогою запитів і т.д. Це питання є головним. Всі інші питання, наприклад, розробка конкретних запитів, форм або звітів, вирішуються майже автоматично при наявності ремісничої навички, що легко здобувається.

*Access* надає користувачеві 5 способів створення таблиць:

- у режимі таблиці,
- у режимі конструктора,
- за допомогою майстра таблиць,
- шляхом імпорту готової таблиці,
- за допомогою встановлення зв'язку з іншою базою даних.



Режим таблиці викликається за допомогою значка **Создание таблицы путем ввода данных**. Користувачеві надається порожня таблиця схожа на звичайну електронну таблицю, куди можна безпосередньо вводити дані. Стандартні назви стовпчиків **Поле1**, **Поле2** і т.п. замінюють на змістовні шляхом подвійного клацання по імені поля або через контекстне меню: м. **Формат** → к. **Переименовать**. Ім'я поля може бути мовою користувача, не довше 64 символів, без службових знаків. Пробіл в іменах використовувати можна. Довжина самого поля не повинна перевищувати 64 символів і може складатися з букв, цифр, пробілів і деяких спеціальних символів. При заповненні таблиці, СУБД аналізує дані й визначає їхній тип. Уведення даних у рядок, форматування таблиці аналогічно відповідним операціям у табличному процесорі *Excel*. *Access* автоматично зберігає запис (рядок таблиці), як тільки користувач покине його межі. При закритті таблиці в користувача запитується її ім'я. Назва таблиці може містити не більше 64 символів, у т.ч. пробіли. Заборонено використати в іменах крапки, !, [,].

Кожна таблиця повинна мати ключове поле, яке СУБД пропонує створити автоматично з ім'ям **Код** і типом даних **Счетчик**. У цьому випадку записи будуть нумеруватися послідовно. Якщо даний варіант не підходить, наприклад, у наших таблицях ключові поля мають іншу значність, то варто відповісти **Нет**, а після створення таблиці визначити ключове поле самостійно в режимі конструктора.

**Режим конструктора** дозволяє керувати процесом створення структури таблиці, задаючи кожному полю тип і відповідні властивості.

Режим конструктора викликається за допомогою значка **Создание таблицы в режиме конструктора**.

*Access* пропонує заповнити спеціальний бланк (рисунок 9.4), у якому визначаються поля, які входять у таблицю.

При заповненні бланка можна вводити додаткову інформацію в таблиці **Свойства поля**, що знаходиться в нижній частині бланка (розмір поля, формат поля, число десяткових знаків, підпис, значення за замовчуванням, умова на значення, повідомлення про помилку, обов'язкове поле, індексоване поле).

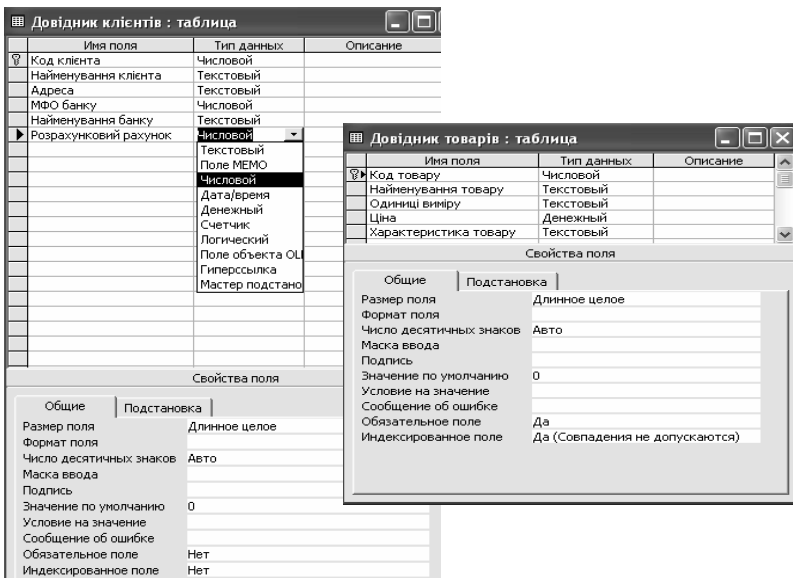


Рисунок 9.4 – Бланки створення таблиць **Довідник клієнтів** і **Довідник товарів** у режимі конструктора

Кожна таблиця повинна мати ключ. Щоб визначити ключове поле, необхідно виділити клацанням потрібне поле й клацнути по інструментальній кнопці з намальованим ключем або: меню **Правка** → п. **Ключевое поле**. *Access*, у випадку відсутності ключового поля, автоматично додає першим ключове поле у вигляді лічильника. Якщо користувача це не влаштовує, то можна перевизначити ключове поле таким само способом, як і його визначення.

**Приклад 9.2.** – Створити таблиці довідників і таблицю замовлень відповідно до результатів розробки структури бази **Замовлення** в прикладі 9.1.

**Виконання.** Кнопка **Пуск** → к. **Программы** → п. **Microsoft Access** → селекторна кнопка **Новая база данных** → указати папку й ім'я бази – **Замовлення** → указати об'єкт **Таблицы** → значок **Создание таблицы в режиме конструктора** → заповнити бланк для **Довідника клієнтів** (рисунок 9.4) → клацнути по полю **Код клієнта** → клацнути по кнопці → закрити вікно бланка у стандартний спосіб → на питання про збереження таблиці клацнути по кнопці **Да** → дати ім'я таблиці

**Довідник клієнтів** → **ОК** → на вкладці таблиці повинен з'явитися значок **Довідник товарів**.

Аналогічним образом створити й зберегти таблиці **Довідник товарів** (бланк конструктора показаний на рисунку 9.4), **Замовлення** (бланк конструктора показаний на рисунку 9.5).

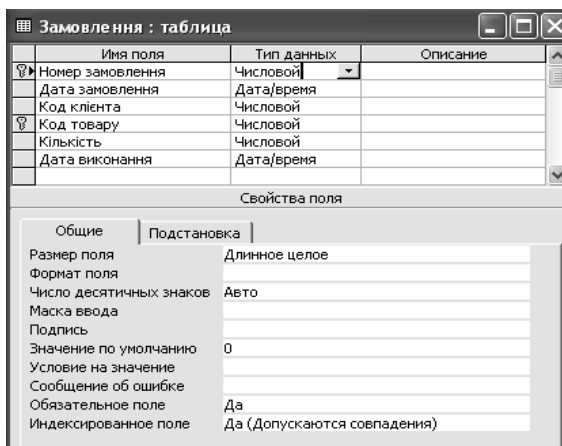


Рисунок 9.5 – Бланк створення таблиці **Замовлення** в режимі конструктора

Первинний ключ може складатися й з декількох полів, тоді він називається **складовим ключем**. Наприклад, для таблиці замовлень номер замовлення, його дата й код клієнта будуть однозначно ідентифікувати будь-яке замовлення. Щоб створити складовий первинний ключ варто виконати наступні дії: відкрити таблицю замовлень у режимі конструктора → клацнути по кнопці вибору поля **Номер замовлення** → утримуючи клавішу **[Ctrl]**, клацнути по кнопці вибору поля **Дата замовлення** → утримуючи клавішу **[Ctrl]**, клацнути по кнопці вибору поля **Код клієнта** → клацнути по інструментальній кнопці з ключем. Ліворуч від назви кожного з виділених полів з'явиться значок із зображенням ключа, що говорить про те, що поле є частиною первинного ключа. Щоб визначити в якому порядку поля повинні входити в первинний ключ, варто клацнути на кнопці панелі інструментів **Індекси**. У діалоговому вікні, що з'явилося, буде представлена інформація про всі індекси таблиці. Для таблиці замовлень він буде один. Кожний складовий індекс може включати до десяти полів.

Створену форму таблиці потрібно зберегти:

меню **Файл** → п. **Сохранить** → дати ім'я таблиці (наприклад, **Замовлення**) → **ОК**

Інші таблиці створюються аналогічно. Варто створити **Довідник клієнтів** і **Довідник товарів**.

Створені таблиці поки порожні. Таблиці простої структури типу довідників, інформація яких носить відносно статичний характер, можуть заповнюватися стандартними можливостями *Access*:

вкладка **Таблицы** → вибрати зі списку назву довідника (наприклад, **Довідник товарів**) → кнопка **Открыть**.

Далі йде просте заповнення рядків таблиці з клавіатури (таблиця 9.1, таблиця 9.2).

Таблиця 9.1. – Довідник клієнтів

Код клієнта	Найменування клієнта	Адреса	МФО банку	Найменування банку	Розрахунковий рахунок
11	ТОВ Мрія	г. Харцизьк	125632	АК ПІБ	123654789
12	МПП Веселка	г. Горлівка	456321	Аваль	147852369
13	НП Корнієнко	г. Донецьк	235421	ДОД УСБ	741258963
14	ДМЗ	г. Донецьк	256642	Ворош.від. ПІБ	142536987
15	Стірол	г. Горлівка	582415	ПІБ	633214548

Таблиця 9.2. – Довідник товарів

Код товару	Найменування товару	Одиниці виміру	Ціна	Характеристика товару
1	маргарин вершковий	пачка, 250	1,50р.	
2	маргарин сонячний	пачка, 250	1,10р.	
3	маргарин молочний	кг	3,40р.	
4	шоколадний десерт	пачка, 250	2,80р.	
5	масло вершкове	кг	12,50р.	

Рядки таблиць мають маркування: з ключем – ключове поле; із трикутником – поточний запис; з олівцем – змінюваний запис.

Редагування й видалення запису йде звичайним для додатків Microsoft Office способом. Причому *Access* стежить за дотриманням цілісності бази даних, тобто, щоб не було порушень у зв'язаних таблицях.

Майстер таблиць завантажує спеціальну програму, що допомагає користувачеві, послідовно виконуючи вказівки, створювати таблицю на основі існуючих шаблонів.

Режим імпорту дозволяє вибрати існуючий файл із даними й імпортувати його в таблицю *Access*. Зокрема, цей режим дозволяє імпортувати таблиці *Excel*.

За допомогою встановлення зв'язку з таблицями іншої БД, можна зв'язати робочу базу даних з однією або декількома таблицями іншої бази даних. При цьому зв'язані таблиці будуть використовуватися так, начебто вони належать до однієї бази.

Як правило, СУБД обробляє кілька таблиць. Як установити зв'язок між таблицями? Зв'язок між таблицями встановлюється по значенням деяких полів. У теорії СУБД відомі чотири варіанти зв'язків між двома таблицями:

1. **Відношення один – до – одного.** При таких відношенні кожному запису першої таблиці відповідає не більше одного запису другої таблиці, і навпаки. Можна сказати, що дві таблиці, пов'язані відношенням один до одного, являють собою одну таблицю, розділену надвоє однією рисою, так, що частина полів опинилася в одній таблиці, а частина – в іншій.

2. **Відношення один – до – багатьох.** При цьому кожному запису первинної (базової) таблиці може відповідати кілька записів іншої таблиці. Наприклад, одному коду товару **Довідника товарів** може відповідати безліч записів у таблиці замовлень.

3. **Відношення багато – до – одного.** При таких відношенні будь-якому запису тієї таблиці бази даних, зв'язок з якою ми розглядаємо, може відповідати будь-яка кількість записів створеної нами нової таблиці, але не навпаки. Цей тип відносин найпоширеніший. Саме можливість створювати міжтабличні зв'язки типу **багато-до-одного** й відрізняє реляційні СУБД від звичайних процесорів плоских файлів. Можна сказати, що це відношення один-до-багатьох навпаки. Наприклад, коду товару, який повторюється в таблиці замовлень, відповідає тільки один код товару в **Довіднику товарів**.

4. **Багато – до – багатьох.** У цьому випадку кожному запису однієї з таблиць може відповідати будь-яка кількість записів іншої таблиці, і навпаки. Наприклад, зв'язок таблиць замовлень і товарів.

*Access* вимагає, щоб у полів, по яких зв'язуються дві таблиці між собою, були однакові типи даних і розмір.

Для таблиць, що зв'язують, одна таблиця виступає в ролі головної, інші – підлеглі. У підлеглій таблиці треба визначити поле **вторинного**

(зовнішнього) ключа, тип даних і розмір якого збігаються з полем первинного ключа головної таблиці. Звичайно обом полям привласнюють однакові імена (але це не обов'язково). Якщо, наприклад, взяти таблицю **Замовлення**, то це буде **Код клієнта**. Вторинні ключі відрізняються від первинних тим, що для них допускається наявність однакових значень полів. Для завдання вторинного ключа в підлеглій таблиці в режимі конструктора варто вибрати поле вторинного ключа, наприклад, **Код клієнта** (рисунок 9.5). В області властивостей поля перейти в “індексоване поле” (останнє за списком) і вибрати елемент **ДА** (збіги допускаються). Аналогічно повинно виглядати індексоване поле для **Коду товару** таблиці **Замовлення**. Змінену структуру таблиці необхідно зберегти.

Далі варто закрити головну й підлеглу таблиці. Потім виконати наступні дії: меню **Сервіс** → команда **Схема даних**. Подвійне клацання мишею по імені таблиці в діалоговому вікні **Добавление таблицы** (рисунок 9.6), поміщає таблицю у вікно **Схема даних**.

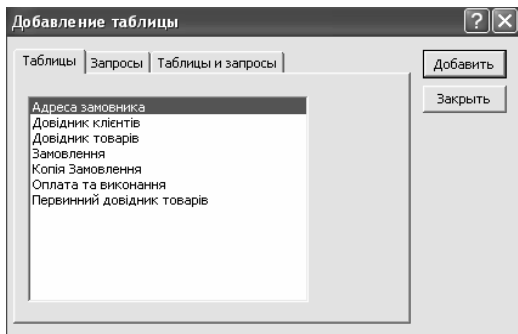


Рисунок 9.6 – Вікно **Добавление таблицы** для перенесення таблиць, що зв'язуються, у вікно **Схема Данных**

**Приклад 9.3.** – Створити схему зв'язку для таблиць бази даних **Замовлення** відповідно до рисунка 9.1.

**Виконання.** 1. Відкрити таблицю **Замовлення** в конструкторі й перевірити, щоб для полів **Код клієнта** й **Код товару** в таблиці **Свойства поля** вкладка **Общие** по рядку **Индексированное поле** мали параметр **Да**.

2. Меню **Сервіс** → команда **Схема даних** → Якщо вікно **Добавление таблицы** не з'явилося поверх вікна **Схема даних**, то клацну-

ти по кнопці **Отобразить таблицу** → виділити ім'я таблиці **Замовлення** → кнопка **Добавить** → аналогічно додати у вікно схеми даних довідники клієнтів і товарів → кнопка **Закрить** вікна **Добавление таблицы** → перетягнути назву поля **Код клієнта** з таблиці **Довідник клієнтів** на поле **Код клієнта** таблиці **Замовлення** → уключити прапорць **Обеспечение целостности данных** у діалоговому вікні **Связи** → перевірити індикатор вибору відносин (повинно бути «**один-до-багатьох**») → кнопка **Создать**. Відношення буде представлено на екрані у вигляді лінії, проведеної між сполучними полями таблиць (рисунок 9.7). Закрийте вікно зі збереженням виконаних установок. Аналогічно варто встановити зв'язок по полю **Код товару** між **Довідником товарів** і таблицею **Замовлення**.

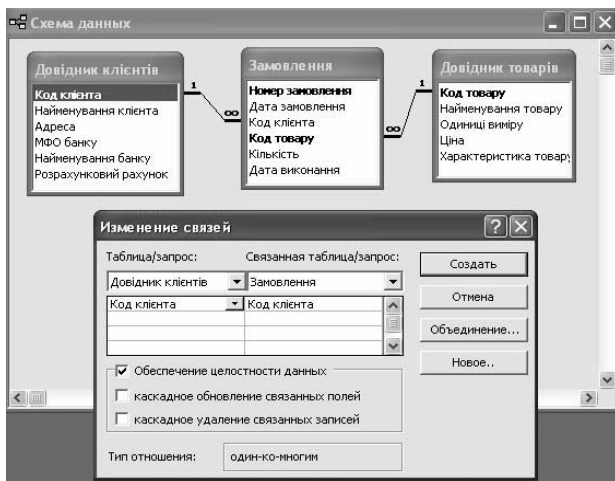


Рисунок 9.7 – Вікно **Схема даних** з установленими зв'язками між таблицями

Примітка 1. У випадку, якщо *Access* показує не той, котрий потрібен, тип відносин, треба в першу чергу перевірити правильність вказівки первинних і вторинних ключів у таблицях, потім можна уточнити тип створюваних зв'язків, клацнувши по кнопці **Объединение.....**

Примітка 2. Рекомендується встановлювати зв'язки до заповнення таблиць. Якщо зв'язки встановлюються для заповнених таблиць, то можливе повідомлення про неможливість створити зв'язок через порушення цілісності даних. Це означає, що в заповнених таблицях є повто-

рювані або неприпустимі коди ключових полів. Варто видалити ці дані й повторити процедуру зв'язування таблиць.

Редагування, видалення й додавання записів у зв'язані таблиці строго контролюється *Access*. Наприклад, якщо запис редагується у підлеглій таблиці, то він буде збережений лише тоді, коли значення сполучного поля присутнє в головній таблиці. При редагуванні головної таблиці можна видалити запис, якщо він не пов'язаний із записами підлеглої таблиці. Уведення в підлеглі таблиці буде збережено, якщо в нових записах використовуються вторинні ключі зі значеннями, які вже є в головній таблиці. Наприклад, із кодами товарів, які вже уведено в довідник.

Дуже часто в процесі розробки бази даних виявляється, що потрібно поміняти місцями поля таблиці, змінити їхні типи або міжтабличні зв'язки. Змінювати можна практично все, однак це може викликати втрату даних, особливо коли вже створені форми або звіти.

Зміну порядку полів таблиці можна виконувати як у режимі таблиці так і в режимі конструктора. Необхідно виділити переміщуваний стовпець або стовпці клацанням по кнопці з ім'ям поля, покажчик миші поставити на заголовок поля, затиснути ліву кнопку й перетягнути на нове місце. На нову позицію лівого краю поля вказує при перетягуванні товста вертикальна риса.

Зміна міжтабличних зв'язків виконується тільки шляхом видалення цих зв'язків і створення нових:

закрити таблиці → відкрити вікно **Схема даних** за допомогою відповідної кнопки панелі інструментів → клацнути по лінії потрібного зв'язку → натиснути клавішу **[Delete]** і підтвердити видалення → внести необхідні зміни в таблицю → створити зв'язки заново.

При роботі з великими таблицями пошук потрібних записів утруднюється. *Access* дозволяє спростити цей процес. Необхідно клацнути по полю, у якому варто виконати пошук, потім меню **Правка** → команда **Найти** або відповідна кнопка на панелі інструментів. Необхідно визначити поля **Образец поиска** й **Совпадение** (поля цілком, з початку поля, із будь-якою частиною поля), а також поле **Области поиска** (у поточному полі, у всіх полях таблиці). Процес пошуку запускається кнопкою **Первое вхождение**. Наступний збіг шукається через кнопку **Поиск далее**.



У таблицях дані завжди відсортовані по первинному ключу, але їх можна пересортувати й по іншому полю. Треба виділити відповідний стовпець і клацнути по одній із кнопок сортування.

### 9.2.2 Екранні форми



Для заповнення більш складних динамічних таблиць у СУБД, а також для захисту інформації від несанкціонованих дій створюються екранні форми. *Access* дозволяє довільним образом форматувати поля форми й розташовувати їх на екрані, додавати, якщо потрібно, такі елементи керування, як прапорці, кнопки й т.п., задавати додаткові правила перевірки цілісності даних, прив'язувати до форми і її елементів програми, керовані подіями.

*Access 2000* дозволяє створювати прості й складні форми, форми з діаграмами й складними таблицями, кнопочві форми й форми – заставки. Форми створюються на підставі таблиць або запитів. Якщо форма будується на підставі декількох таблиць, то спочатку варто створити відповідний запит, а потім за його допомогою форму.

Форму можна створювати в режимі конструктора й за допомогою майстра. При цьому в режимі конструктора елементи на форму можна заносити за допомогою **Панелі елементів**, що нагадує панель інструментів Visual Basic:



Для починаючих користувачів при побудові електронних форм передбачений **Мастер форм**, що викликається в такий спосіб:

вкладка **Форми** → кнопка **Создать** → вибрати зі списку **Мастер форм** → **ОК**. Розкрити список **Таблицы/запросы** (рисунок 9.8) й вибрати з нього потрібну таблицю, виділити доступне поле й нажити кнопку , вибрати наступне поле,  ... нажити кнопку **Далее**, указати як розташовувати текстові поля у формі (наприклад, в один стовпець), кнопка **Далее**, вибрати шрифт (**Обычный**), кнопка **Далее**, задати ім'я форми й указати індикатор вибору **открытие формы для просмотра** або **ввода данных**, кнопка **Готово**.

Якщо користувач хоче змінити властивості електронної форми, наприклад, заборонити зміну раніше уведених записів, потрібно перейти в режим конструювання форми:

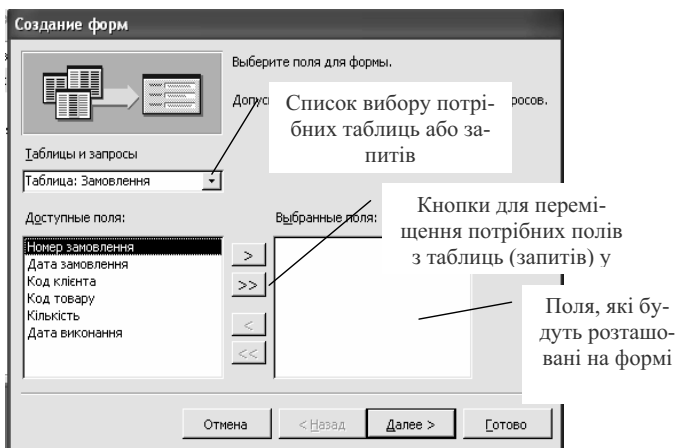



Рисунок 9.8 – Вікно майстра для створення форми

Меню **Вид** → команда **Конструктор** або кнопка . Приклад форми для таблиці **Замовлення** показаний на рисунку 9.9.

Властивості має як форма в цілому, так і кожний елемент окремо. Для того щоб працювати із властивостями саме форми, досить клацнути правою кнопкою миші на кнопці виділення форми й викликати команду **Свойства** або м. **Вид** → к. **Свойства**.

Наприклад, на вкладці **Данные** для керування доступом можна використати наступні властивості:

Разрешить изменение	–	ні
Разрешить удаление	–	ні
Разрешить добавление	–	так
Ввод данных	–	так.

Якщо необхідно змінити властивості якого-небудь поля, то варто викликати через контекстне меню властивості цього поля. Крім того, можна змінити форму представлення поля (наприклад, використати список або поле зі списком) через команду **Преобразовать элемент В.....** контекстного меню.

**Приклад 9.4.** – Створити форму для уведення й додавання даних у таблицю **Замовлення**, при цьому дані в поля **Код клієнта** й **Код товару** повинні вибиратися з наявних даних відповідних довідників. При

заповненні форми користувачеві дозволено вводити й коректувати записи.

**Виконання.** 1. Вкладка вікна бази дані **Формы** → подвійне клацання по кнопці **Создание формы с помощью мастера** → Розкрити список **Таблицы/запросы** й вибрати таблицю **Замовлення** (рисунок 9.8) → клацнути по кнопці **>>** для перенесення всіх полів таблиці у форму → кнопка **Далее** → вибрати селекторну кнопку **в один столбец** → кнопка **Далее** → вибрати стиль оформлення (наприклад, **диффузный**) → кнопка **Далее** → задати ім'я форми (**Замовлення**) і вказати індикатор вибору **Изменить макет формы** → кнопка **Готово**.

У результаті форма **Замовлення** буде відкрита в режимі конструктора для уточнення властивостей самої форми, зміни форми представлення й властивостей її окремих полів (рисунок 9.9 а).

2. Викликати контекстне меню поля **Код клієнта** → к. **Преобразовать элемент в** → вибрати **Поле со списком** → контекстне меню поля **Код клієнта** → к. **Свойства** → вкладка **Данные** (рисунок 9.10) → уточнити параметри відповідно до рисунка → вкладка **Макет** → у рядку **Число столбцов** вказати **2** для того, щоб користувач бачив не тільки код клієнта, але і його назву.

Збільшити форму й поля зі списками як графічні об'єкти відповідно до рисунка 9.9 б, щоб у списках, що випадають, було видно дві графи.

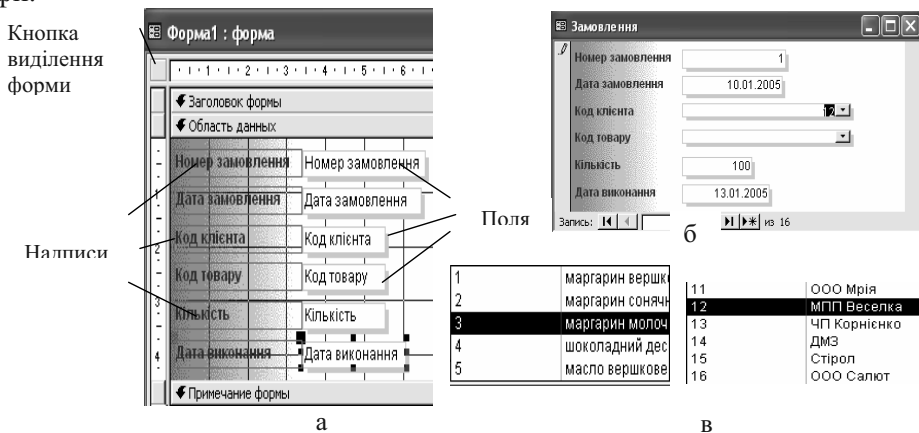


Рисунок 9.9 – Форма **Замовлення** в режимі конструктора: а – вхідна, б – з уточненими елементами й властивостями, в – фрагменти списків, що випадають, уточненої форми.

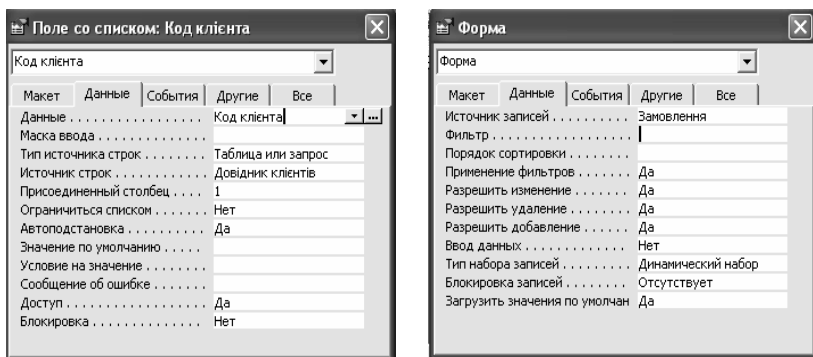


Рисунок 9.10 – Вікна **Властивостей** поля зі списком **Код клієнта** й **Форми**

3. Аналогічним чином перетворити елемент **Код товару** й уточнити його властивості ( Джерело рядків – **Довідник товарів**).

4. Для уточнення властивостей форми в цілому – контекстне меню по кнопці виділення форми → к. **Властивості** → уточнити рядки вкладки **Дані** відповідно до рисунка 9.10.

Якщо форма готова, залишилося її зберегти:

меню **Файл** → команда **Сохранить** → дати ім'я **Замовлення** → **ОК**.

Використовуючи створену форму необхідно заповнити таблицю 9.3.

Таблиця 9.3. – **Замовлення**

Номер замовлення	Дата замовлення	Код клієнта	Код товару	Кількість	Дата виконання
1	10.01.2005	12	1	100	13.01.2005
7	11.01.2005	13	2	75	12.01.2005
3	11.01.2005	11	2	25	14.01.2005
2	11.01.2005	15	2	50	13.01.2005
5	12.01.2005	11	3	75	15.01.2005
4	12.01.2005	12	4	110	14.01.2005
8	15.01.2005	15	5	200	18.01.2005
6	20.01.2005	14	3	50	23.01.2005


Як видно з таблиці 9.3 у прикладі використовуються однорядкові замовлення. Якщо клієнт в одному замовленні випишує кілька товарів, то спливає кілька питань:


- недостатньо у якості ключа використовувати поле **Номер замовлення**, тому що він повинен повторюватися на декількох рядках;
- при уведенні даних необхідно вводити дані по клієнту з мінімальним дублюванням.

Варіант рішення даного завдання розглянемо на прикладі 9.5.

Приклад 9.5. – На підставі бази даних **Замовлення** створити базу даних **Замовлення зі складною формою**, у якій зберегти структуру таблиць і створити форму, що дозволяє створювати багаторядкові замовлення для одного клієнта.

Виконання. 1. В Windows скопіювати базу даних **Замовлення** й перейменувати її у **Замовлення зі складною формою**.

2. Відкрити базу даних **Замовлення зі складною формою** → відкрити об'єкти **Таблиці** → відкрити таблицю **Замовлення** в режимі конструктора ( ви-ділити таблицю і кнопка  ).

3. Перевизначити ключове поле таблиці, створивши складовий ключ: при натиснутій клавіші **Ctrl** виділити рядок **Код товару** → клацнути по кнопці **Ключевое поле**  . Значок ключа повинен стояти у двох рядків –**Номер замовлення** й **Код товару**. Для нашого приклада такий ключ буде однозначно визначати кожний рядок таблиці.

4. При перегляді або заповненні замовлення форма повинна відображати дані про клієнта з **Довідника клієнтів**, дані про замовлення, дані про замовлений товар із **Довідника товарів**. У цьому випадку зручніше використати складну форму.

Перейти до об'єктів **Форми** → кнопка **Создать** → вибрати **Мастер форм** → **ОК** → Розкрити список **Таблицы/запросы** й вибрати таблицю **Довідник клієнтів** → клацнути по кнопці **>>** для перенесення всіх полів таблиці у форму → у списку **Таблицы/запросы** вибрати таблицю **Замовлення** → перенести у форму поля **Номер замовлення, Дата замовлення, Код товару, Кількість і Дата виконання** → у списку **Таблицы/запросы** вибрати таблицю **Довідник товарів** → перенести у форму поля **Найменування товару, Одиниці виміру, Ціна** → кнопка **Далее** → у вікні **Создание форм** (рисунок 9.11) указати селектору

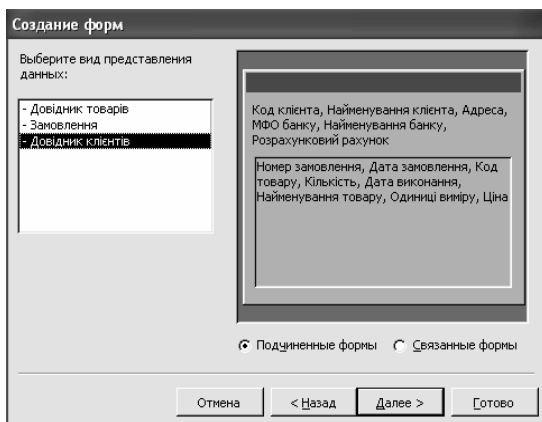




Рисунок 9.11 – Вікно по створенню підлеглих і зв'язаних форм

кнопку **Подчиненные формы** → кнопка **Далее** → вибрати зовнішній вигляд підлеглої форми табличний → кнопка **Далее** → вибрати стиль оформлення → кнопка **Далее** → дати ім'я основній формі **Клієнти й замовлення**, підлеглий **Замовлення підлегла форма** → кнопка **Готово**.

Зовнішній вигляд вікна показаний на рисунку 9.12. Як бачимо таблиця видна не повністю, уведення утруднене через відсутність довідкових даних – форма вимагає форматування.

5. На вкладці **Формы** в результаті роботи майстра з'явилося дві форми, які можна формувати окремо.

визвати форму **Клієнти й замовлення** в режимі конструктора → збільшити загальне поле для створення форми → перемістити й збільшити поля по клієнту відповідно до рисунка 9.12 б) → перетворити поле **Код клієнта** в поле зі списком аналогічно прикладу 9.4 → за допомогою кнопки  визвати панель елементів для створення ілюстрації → клацнути по кнопці  й на формі окреслити місце для малюнка → автоматично викликається вікно для вибору файлу з малюнком → виділити потрібний файл і **ОК** → визвати властивості верхньої таблиці (через контекстне меню кнопки виділення форми) → вкладка **Макет** → у рядку **Полосы прокрутки** вказати **Отсутствуют** → зберегти зміни й закрити форму → відкрити форму й перевірити бажаний результат.

6. У режимі відкритої форми відредагувати ширину стовпців таблиці підлеглої форми.

7. Відкрити форму **Замовлення підлегла форма** в режимі конструктора → перетворити поле **Номер замовлення** в поле зі списком → як джерело рядків використати таблицю **Замовлення** (щоб не тримати в голові номер останнього замовлення) → перетворити поле **Код товари** в поле зі списком аналогічно прикладу 9.4) → викликати властивості всієї таблиці → вкладка **Макет** → у рядку **Полосы прокрутки** поставити **Тільки по вертикали** → у рядку **Кнопки перехода** – **Нет** → зберегти зміни й перевірити роботу форми в цілому.

Клієнти й замовлення

Код клієнта: 11  
 Найменування клієнта: 000 Мрія  
 Адреса: м.Харцизьк  
 МФО банку: 653214  
 Найменування банку: АК ПІБ  
 Розрахунковий рахунок: 123654789

Замовлення підлегла форма

Номер за	Дата замов	Код това	Кількість	Дата викон	Найменування товару	
1	12.01.2005	1	100	15.01.2005	маргарин вершковий	пачк
3	11.01.2005	2	25	14.01.2005	маргарин сонячний	пачк
5	12.01.2005	3	75	15.01.2005	маргарин молочний	кг

Запис: 1 из 3  
 Запис: 1 из 7

а)

Клієнти й замовлення

Код клієнта: 11  
 Найменування клієнта: 000 Мрія  
 Адреса: м.Харцизьк  
 МФО банку: 653214  
 Найменування банку: АК ПІБ  
 Розрахунковий рахунок: 123654789

Замовлення підлегла форма

Номер за	Дата замов	Код това	Кількість	Дата викон	Найменування товару	Одиниці виміру
5	12.01.2005	1	100	15.01.2005	маргарин вершковий	пачка, 250
3	11.01.2005	2	25	14.01.2005	маргарин сонячний	пачка, 250
5	12.01.2005	3	75	15.01.2005	маргарин молочний	кг

Запис: 1 из 3  
 Запис: 1 из 7

б)

Рисунок 9.12 – Первісний а) і остаточний б) вид складної форми до прикладу 9.5

### 9.2.3 Запити

Запити забезпечують швидкий і ефективний доступ до даних, що зберігаються в таблицях. Завдяки запитам можна виконати сортування, обчислити вираження (обчислюють поля, що у таблицях *Access* не дозволяється використовувати), звести разом дані зі зв'язаних таблиць.

При виконанні запиту СУБД відображає результат у режимі таблиці, але цей результат не зберігається. У запиті *Access* зберігає тільки інструкції (програму) про те, як повинні бути організовані дані в результаті виконання запиту. Саме це й дозволяє щораз при формуванні запиту одержувати актуалізовані дані.

В *Access* можливе створення запитів наступних чотирьох основних типів, що дозволяють вирішувати різні завдання.

- **Запит на вибірку.** Ці запити витягають дані з однієї або декількох таблиць і відображують їх у таблиці.

- **Перехресний запит.** Ці запити підсумовують дані з однієї або декількох таблиць у формі електронної таблиці. Такі запити корисні для аналізу даних і створення графіків або діаграм, заснованих на сумі значень числових полів багатьох записів.

- **Запити на зміну.** Ці запити створюють нові таблиці бази даних із таблиць запитів або вносять у таблиці значні зміни. Такі запити дозволяють додавати в таблицю нові записи або видаляти їх, або вносити зміни в записи за допомогою виражень, убудованих у конструкцію запиту.

- **Запити з параметрами** (також називані параметризованими запитами). Ці запити повторно застосовують наявні запити, із умовами відбору, які задає користувач при зверненні. Після виконання запиту з параметрами, *Access* відображає діалогове вікно, що пропонує користувачеві ввести нову умову відбору. Запити з параметрами фактично не є окремими запитами, оскільки функції запитів із параметрами можна додати до запитів на вибірку, запитів на зміну й перехресних запитів.

#### Запити на вибірку

Для створення запиту на вибірку варто вибрати об'єкт **Запросы** → кнопка **Создать** → виділити зі списку **Конструктор** → **ОК**



Потім варто визначити вхідні дані (рисунок 9.13), із якими буде працювати запит, а потім його зберегти з певним ім'ям.

**Приклад 9.6.** – Створити запит, що дозволяє визначити, який клієнт замовив які товари (запит “Замовлено”).

Виконання. Об'єкт **Запросы** → кнопка **Создать** → виділити зі списку **Конструктор** → **ОК** → вкладка **Таблицы** → вибрати зі списку таблицю **Замовлення** → кнопка **Добавить** → вибрати **Довідник клієнтів** → кнопка **Добавить** → вибрати **Довідник товарів** → кнопка **Добавить** → кнопка **Закричь**.

Тепер із таблиць необхідно мишкою перетягнути потрібні поля, які повинні ввійти в результат запиту (**Номер замовлення, Дата замовлення, Найменування клієнта, Найменування товару, Кількість, Ціна**), у вікно запиту (у рядок **Поле**). Структуру запиту необхідно зберегти з ім'ям **Замовлено**. Побачити результат можна, якщо виділити позначку **Замовлено** на вкладці **Запросы** й клацнути по кнопці **Открыть** (таблиця 9.4).

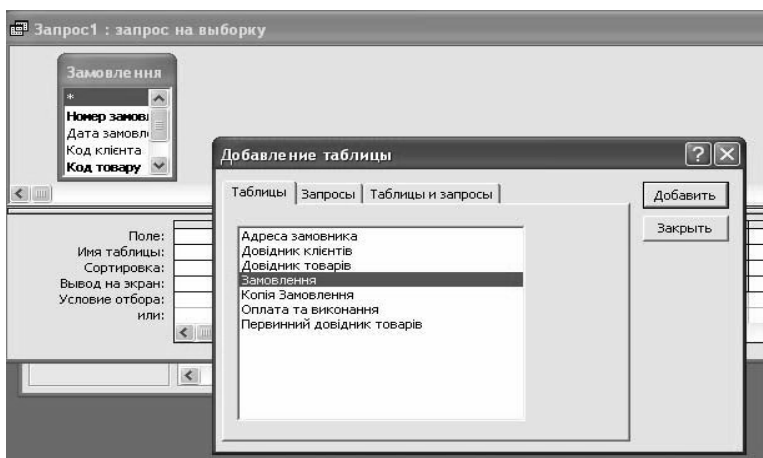


Рисунок 9.13 – Вікно для вибору даних запиту на вибірку

Таблиця 9.4. – Результат роботи запиту **Замовлено**

Номер замовлення	Дата замовлення	Найменування клієнта	Найменування товару	Кількість	Одиниці виміру	Ціна
1	10.01.2005	МПП Веселка	маргарин вершковий	100	пачка, 250	1,50р.
2	11.01.2005	Стірол	маргарин сонячний	50	пачка, 250	1,10р.
3	11.01.2005	ТОВ Мрія	маргарин сонячний	25	пачка, 250	1,10р.
4	12.01.2005	МПП Веселка	шоколадний десерт	110	пачка, 250	2,80р.
5	12.01.2005	ТОВ Мрія	маргарин молочний	75	кг	3,40р.
6	20.01.2005	ДМЗ	маргарин молочний	50	кг	3,40р.
7	11.01.2005	НП Корнієнко	маргарин сонячний	75	пачка, 250	1,10р.
8	15.01.2005	Стірол	масло вершкове	200	кг	12,50р.

У запитах на вибірку можна задавати критерії відбору записів, для чого в рядку **Условия отбора** бланка запиту в потрібнім полі варто вказати критерій за допомогою оператора **Between...And**, операторів порівняння (<, >, <>, <= і >=), підстановочних символів або виражень. Наприклад, у поле **Кількість** >=100 або в поле **Ціна** <50. Зразки критеріїв відбору показані в таблиці 9.5.

Таблиця 9.5. – Зразки критеріїв відбору в запитах *Access*

Вираження	Опис
>234	Числа, що перевищують 234
<b>Between #02.02.93# And #01.12.93#</b>	Дати в діапазоні від 2-фев-93 до 1-дек-93
<1200.45	Числа, менші чим 1200,45
>="Іванов"	Всі прізвища, починаючи з «Іванов» і до кінця алфавіту

Бланк створення запиту **Замовлено** за певні дати показаний на рисунку 9.14.

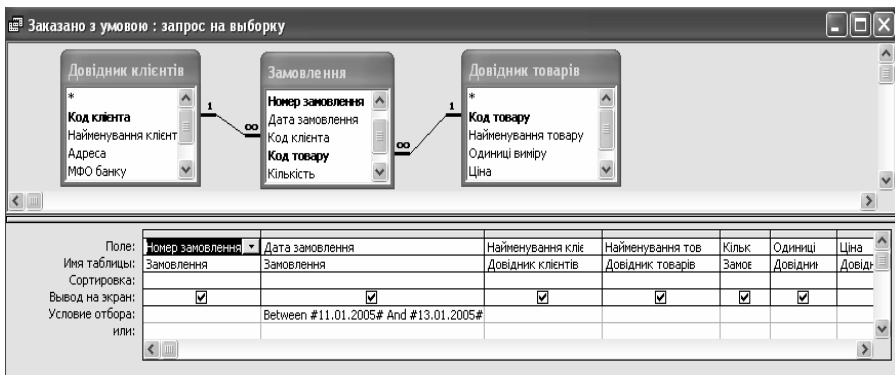


Рисунок 9.14 – Бланк запити на вибірку з умовою

У базах даних *Microsoft Access* для пошуку значень полів, записів або імен файлів у діалогових вікнах **Пошук** і **Заміна**, у запитах, командах і вираженнях допускається використання підстановочних символів (таблиця 9.6).

Таблиця 9.6 – Підстановочні символи для пошуку в *Access*

Символ	Використання	Приклад
1	2	3
*	Відповідає будь-якій цифрі або будь-якому символу. Може використатися в якості першого або останнього символу текстового рядка.	Wh* – пошук слів what, white і why.
?	Відповідає будь-якому текстовому символу.	B?ll – пошук слів ball, bell і bill.
[ ]	Відповідає будь-якому одному символу з узятих у дужки.	B[ae]ll – пошук слів ball і bell, але не bill.
!	Відповідає будь-якому одному символу, крім узятих у дужки.	b[!ae]ll – пошук слів bill і bull, але не bell або ball.
-	Відповідає будь-якому символу з діапазону. Необхідно вказувати цей діапазон по зростанню (від A до Z, але не від Z до A).	b[a-c]d – пошук слів bad, bbd і bcd.
#	Відповідає будь-якій цифрі.	l#3 – пошук значень 103, 113, 123.

Примітки – При використанні підстановочних знаків для пошуку символу зірочки (\*), знаку питання (?), знака номера (#), відкриваючої квадратної дужки ([]) або дефіса (-) необхідно містити ці символи у квадратні дужки. Наприклад, для пошуку знаку питання в діалоговому вікні **Поиск** варто задати зразок пошуку [?]. При пошуку дефіса разом з іншими символами варто розмістити дефіс до або після всіх інших символів у дужках. (Однак, якщо після відкриваючої дужки розташовується знак оклику, дефіс варто помістити після знаку оклику). Якщо виконується пошук знаку оклику (!) або закриваючої дужки (]), то містити їх у квадратні дужки не потрібно, пари квадратних дужок необхідно укласти в круглі дужки ([ ]).

Для вибору за допомогою запиту записів, що містять значення, які не збігаються з певним значенням, необхідно ввести в чарунку рядка **Условие отбора** для відповідного поля вираження, що містить оператор **Not**.

Наприклад, вираження **Not Ф\*** в чарунці **Условие отбора** для поля **Найменування клієнта** визначає відбір всіх назв, що не починаються з букви «Ф», а вираження **Not 2** для поля **Код клієнта** визначає відбір усіх клієнтів, код яких не дорівнює 2.

Вибір за допомогою запиту записів із непустими значеннями – увести в чарунку рядка **Условие отбора** для відповідного поля вираження **Not Null** або **Is Not Null**.

Вибір за допомогою запиту записів із порожніми значеннями – увести в чарунку рядка **Условие отбора** для відповідного поля вираження **Is Null**.

Вибір за допомогою запиту записів, що містять значення поточної дати – увести в чарунку рядка **Условие отбора** для відповідного поля вираження **Date()** (без пробілу між дужками).

Наприклад, щоб знайти всі замовлення, дата яких збігається з поточною датою, уведіть вираження **Date()** у поле **Дата замовлення**.


Вибір за допомогою запиту записів, що містять значення зі списку значень – увести в чарунку рядка **Условие отбора** для відповідного поля вираження, у якому використовується оператор **In**. Наприклад, для пошуку замовників, які обслуговуються в банках ДОД УСБ, АК ПІБ, Аваль варто ввести в чарунку **Условие отбора** для поля **Найменування банку** таблиці **Замовлення** вираження:

In(ДОД УСБ, Аваль, АК ПІБ)

Ця ж умова може бути задана в наступному виді:


ДОД УСБ Or Аваль Or АК ПІБ

### Поля, які розраховуються у запитах

Як уже говорилося, у полях таблиць бази даних зберігаються первинні дані, те, що можна обчислити в базі даних не зберігають, а одержують за допомогою запитів. Поле, яке розраховується, створюється шляхом простого уведення вираження для обчислення в порожньому стовпці бланка запиту. Так, наприклад, вартість замовлення обчислюється за допомогою наступного вираження: [Ціна] \* [Кількість]. *Access* автоматично підставляє **Выражение1: [Ціна] \* [Кількість]**. **Выражение1**: можна замінити на більш змістовну назву Сума. Назви полів у квадратних дужках повинні строго відповідати назвам полів вхідних таблиць або запитів. З цього погляду надійніше користуватися спеціальним засобом для побудови формул – **Построителем**. Цей засіб стає доступним у режимі **Конструктор** через контекстне меню поля з формулою → к. **Построить** або через кнопку  панелі інструментів **Конструктор запитів** (рисунок 9.15).

Приклад 9.7. Створити запит **Сума замовлення**, у якому по кожному рядку замовлення буде розраховуватися сума.

Виконання. Необхідний запит зручно створити на підставі вже існуючого запиту на вибірку **Замовлено**.

Вибрати об'єкт **Запросы** → кнопка **Создать** → виділити **конструктор** → **ОК** → вкладка **Запросы** → виділити **Замовлено** → кнопка **Добавить** → кнопка **Закреть** → перетягнути поля запиту **Замовлено** у відповідні поля нового запиту → клацнути по полю для заголовка **Сума** → кнопка  панелі інструментів **Конструктор запитів** → у вікні **Построитель выражений** (рисунок 9.15) у дереві папок відкрити папку вхідного об'єкта (у нашому прикладі це запит **Замовлено**) → подвійним клацанням по назві потрібного поля (**Кількість**) відправити його у верхню частину вікна → клацнути по кнопці з операцією (\*) → подвійним клацанням по назві потрібного поля (**Ціна**) відправити його

у верхню частину → кнопка **ОК**. У бланку запиту у формулі, що з'явилася, замінити слово **Виразення1** на більш підходяще за змістом (**Сума**) → зберегти запит з ім'ям **Сума замовлення**.

У результаті запиту буде отримана таблиця з новою графою **Сума**, правити яку користувач не може (таблиця 9.7).



Рисунок 9.15 – Бланк конструктора запитів і вікно **Построителя выражений** при формуванні запиту з розрахунками

Таблиця 9.7. – Результат роботи запиту **Сума замовлення**

Номер замовлення	Дата замовлення	Найменування клієнта	Найменування товару	Кількість	Одиниці виміру	Цена	Сума
1	10.01.2005	МПП Веселка	маргарин вершковий	100	пачка, 250	1,50р.	150,00р.
2	11.01.2005	Стірол	маргарин сонячний	50	пачка, 250	1,10р.	55,00р.
3	11.01.2005	ТОВ Мрія	маргарин сонячний	25	пачка, 250	1,10р.	27,50р.
4	12.01.2005	МПП Веселка	шоколадний десерт	110	пачка, 250	2,80р.	308,00р.
5	12.01.2005	ТОВ Мрія	маргарин молочний	75	кг	3,40р.	255,00р.
6	20.01.2005	ДМЗ	маргарин молочний	50	кг	3,40р.	170,00р.

Номер замовлення	Дата замовлення	Найменування клієнта	Найменування товару	Кількість	Одиниці виміру	Ціна	Сума
7	11.01.2005	НП Корнієнко	маргарин сонячний	75	пачка, 250	1,10р.	82,50р.
8	15.01.2005	Стірол	масло вершкове	200	кг	12,50р.	2 500,00р.

Наведений приклад ілюструє можливості *Access* здійснювати розрахунки по окремим записам. Але досить часто потрібно підраховувати різні підсумки у таблицях. Для цього використовуються **підсумкові запити**.

**Підсумкові запити** відрізняються від звичайних. Вони формуються з полів, по яким виконується групування даних, полів, для яких виконуються обчислення, можна також включати в них поля для визначення умов запиту. По ознакам (полям) групування здійснюється добір записів, які приймають участь у підрахунку підсумків. Наприклад, для підрахунку суми всіх замовлень за день ознакою групування буде поле дати.

**Приклад 9.8.** Створити запит **Товари по датах**, у якому визначається кількість товарів замовлених по датам у розрізі найменувань.

**Виконання.** Для запиту, що створюється, необхідно підраховувати кількість товарів по кожній даті (старша ознака групування), кожному найменуванню (наступна ознака групування). Створити запит **Товари по датах** можна на основі вже існуючого запиту **Заказано**.

У вікні конструктора на підставі запиту **Замовлено** створити новий запит з полями **Дата замовлення**, **Найменування товару**, **Кількість**, **Ціна** → м. Вид → команда **Групповые операции** (у бланку запиту з'являється рядок **групповые операции**) → для всіх ознак групування у цьому рядку варто залишити операцію **группировка** → у рядку **Сортировка** для кожної ознаки, що групується можна вибрати спосіб упорядкування, наприклад, **по возрастанію**, → для поля **Кількість** у рядку **Групповые операции** вибрати **SUM** (рисунок 9.16).

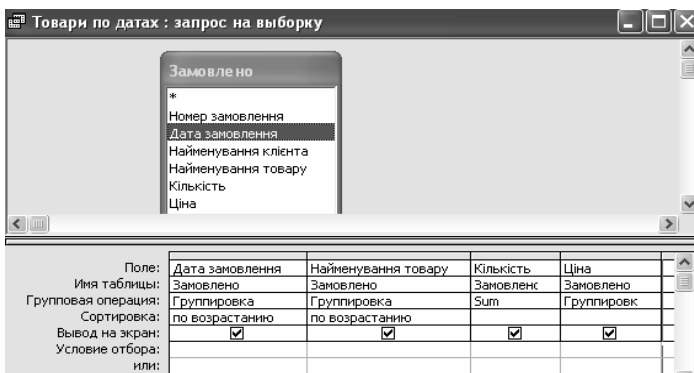


Рисунок 9.16 – Бланк створення підсумкового запиту

Залишилося зберегти запит під потрібним ім'ям: меню **Файл** → команда **Сохранить** → увести ім'я запиту **Товари по датах** → **ОК**. Результат виконання запиту показаний у таблиці 9.8.

Таблиця 9.8. – Результат роботи запиту **Товари по датах**

Дата заповнення	Найменування товару	Sum-Кількість	Ціна
10.01.2005	маргарин вершковий	100	1,50р.
11.01.2005	маргарин сонячний	150	1,10р.
12.01.2005	маргарин молочний	75	3,40р.
12.01.2005	шоколадний десерт	110	2,80р.
15.01.2005	масло вершкове	200	12,50р.
20.01.2005	маргарин молочний	50	3,40р.

Одержувані на екрані результати запитів можна переглядати й друкувати стандартним способом.

**Приклад 9.9.** Створити запит **Замовлено по датах** у сумовому вираженні, у якому по кожній даті будуть показана загальна сума замовлень, мінімальна, максимальна і середня суми замовлень.

**Виконання.** Необхідний запит можна створити на підставі запиту **Сума замовлення** з використанням статистичних функцій.



Створити новий запит → перенести в нього поля **Дата замовлення** і **Сума** з запиту **Сума замовлення** (поле **Сума** перенести в чотири поля нового запиту для виконання необхідних обчислень) → м. **Вид** → команда **Групповые операции** → виставити в рядку **групповая операция** необхідні функції (рисунок 9.17) → зберегти запит з ім'ям **Замовлено по датах**.



Рисунок 9.17 – Бланк запиту для прикладу 9.9

Результат виконання запиту показаний у таблиці 9.9.

Таблиця 9.9. – Результат роботи запиту **Замовлено по датах** з використанням статистичних функцій

Дата замовлення	Sum-Сума	Min-Сума	Max-Сума	Avg-Сума
10.01.2005	150,00р.	150,00р.	150,00р.	150,00р.
11.01.2005	165,00р.	27,50р.	82,50р.	55,00р.
12.01.2005	563,00р.	255,00р.	308,00р.	281,50р.
15.01.2005	2 500,00р.	2 500,00р.	2 500,00р.	2 500,00р.
20.01.2005	170,00р.	170,00р.	170,00р.	170,00р.

**Примітка 1.** Якщо необхідно в одному стовпці робити обчислення й вказувати умови відбору, то цей стовпець у запиті варто повторити

двічі й в кожному вказати одну з вимог. Один із цих стовпців не треба виводити на екран.

Примітка 2. Якщо в тому самому стовпці запиту необхідно розраховувати рядкові формули й підсумкову формулу, то дану роботу варто розбити на два запити – перший з підрахунком порядкових підсумків, другий – на підставі попереднього запиту підрахунок підсумку по стовпцю. Наприклад, для підрахунку загальної суми замовлень по дням необхідно створити запит **Сума замовлення**, а потім запит **Сума замовлень по дням**.

### Перехресні запити

Перехресні запити можна створювати за допомогою майстра й без нього. Застосування майстра перехресних запитів обмежено використанням однієї таблиці або одного запиту.

Створення перехресного запиту без допомоги майстра базується на створенні запиту на вибірку, для якого потім змінюється тип на перехресний. При цьому в бланку запиту (рисунок 9.18) з'являється рядок **Перекрестная таблица**, у якій варто вказати, яке (які) поля будуть заголовками рядків, а яке – заголовком стовпців. Для таких полів потрібно залишити в рядку **Групповая операция** значення **Группировка**, яке встановлене за замовчуванням. За замовчуванням, заголовки стовпців сортуються за абеткою чи за числовим значенням. Якщо потрібно розташувати їх в іншому порядку, або якщо потрібно обмежити кількість відображуваних заголовків стовпців, варто використовувати властивість **Заголовки столбцов** запиту.

Примітка. Значення **Заголовки столбцов** можна задати тільки для одного поля.

Приклад 9.10. Необхідно створити перехресний запит **Замовлення на товари по датах**, у якому по рядках повинні відбиватися суми замовлення по окремих товарах, а по стовпцях – дати.



Рисунок 9.18 – Бланк перехресного запиту

**Виконання.** Створити новий запит на вибірку з полями **Найменування товару**, **Дата замовлення**, **Сума** на підставі запиту **Сума замовлення** → на панелі інструментів розкрийте список кнопки **Тип запиту** і виберіть **Перекрестный** (або меню **Запит** → к. **Перекрестный**) → у рядку **Перекрестная таблица** для графі **Найменування товару** виберіть значення **Заголовки строк** → у рядку **Перекрестная таблица** для графі **Дата замовлення** виберіть значення **Заголовки столбцов** → для графі **Сума** в цьому ж рядку виберіть параметр **Значение** (рисунок 9.18) → у рядку **Групповая операция** для графі **Сума** виберіть функцію **Sum** → зберегти запит з ім'ям **Замовлення на товари по датах**.

**Примітка.** Параметр **Значение** може бути обраний тільки для одного поля.

Результат виконання запиту показаний у таблиці 9.10.

Таблиця 9.10. – Результат роботи запиту **Замовлення на товари по датах**

Найменування товару	10_01_2005	11_01_2005	12_01_2005	15_01_2005	20_01_2005
маргарин молочний			255,00р.		170,00р.
маргарин вершковий	150,00р.				
маргарин сонячний		165,00р.			
масло вершкове				2 500,00р.	
шоколадний десерт			308,00р.		

Щоб задати умову відбору, яка обмежує заголовки рядків, що відбирають, до виконання обчислень, уведіть вираження в рядок **Условие отбора** для поля зі значенням **Заголовки** строк у чарунку рядка **Перекрестная таблица**. Наприклад, можна вивести підсумкові суми продажів для деяких найменувань товарів.

Щоб задати умову відбору, яка обмежує записи, що відбирають, до групування заголовків рядків і до заповнення перехресної таблиці, додайте в бланк запиту поле, для якого буде задана умова відбору, виберіть для нього значення **Условие** в чарунці рядка **Групповая операция**, залишіть чарунку у рядку **Перекрестная таблица** порожньою й уведіть вираження в чарунку рядка **Условие отбора**. (Поля, що мають значення **Условие** в рядку **Групповая операция**, не виводяться в результатах запита).

### Запити на зміну

Для пошуку запису в базі даних і зміни в ній значення поля цілком достатньо режиму таблиці. Але якщо таку роботу необхідно зробити в безлічі записів, тоді використовують запити на зміни. До запитів на зміни відносяться запити на відновлення, створення нової таблиці, вставки й видалення наборів даних із таблиць.

Запит на зміну створюється шляхом конвертації простого запиту на вибірку в запит на зміну:

відкрити запит на вибірку в режимі конструктора → м. **Запрос** → к. **Создание таблицы (Обновление, Добавление, Удаление)**.

Приклад 9.11. Необхідно створити таблицю **Адреса замовника** зі списком адрес клієнтів, які зробили замовлення.

Виконання. Створити запит на вибірку **Адреса клієнта**, що містить поля **Номер замовлення**, **Дата замовлення**, **Найменування клієнта**, **Адреса клієнта** з таблиць **Замовлення** й **Довідник клієнтів**.

Після перевірки працездатності запиту необхідно відкрити його в режимі конструктора → меню **Запрос** → к. **Создание таблицы** → увести ім'я **Адреса замовника** → ОК.

Запит на вибірку буде перетворений у запит на зміну. У процесі виконання даного запиту *Access* видасть ряд попереджень про зміни, які потребують згоди користувача. Після виконання цього запиту на вкладці **Таблиця** з'явиться нова таблиця **Адреса замовника**. Щоразу при виконанні запиту на зміну у випадку змін у базі даних буде мінятися вміст таблиці, що створюється (таблиця 9.11).

Таблиця 9.11. – Результат роботи запиту на зміну **Адреса замовника**

Номер замовлення	Дата замовлення	Найменування клієнта	Адреса
1	10.01.2005	МПП Веселка	г. Горлівка
2	11.01.2005	Стірол	г. Горівка
3	11.01.2005	ТОВ Мрія	г. Харцизьк
4	12.01.2005	МПП Веселка	г. Горлівка
5	12.01.2005	ТОВ Мрія	г. Харцизьк
6	20.01.2005	ДМЗ	г. Донецьк
7	11.01.2005	НП Корнієнко	г. Донецьк
8	15.01.2005	Стірол	г. Горлівка

### Запит із параметрами

Запит із параметрами дозволяє вводити по підказці *Access* конкретні умови вибірки, які називаються параметром. Такі умови можна вводити як у запит на вибірку, так і в запит на зміну.

Приклад 9.12. Створити запит **Замовлено з параметром**, який повинний дозволяти користувачу відбирати замовлення по визначених клієнтах.

Виконання. Скопіювати запит **Замовлений** (через контекстне меню) і вставити його з ім'ям **Замовлено з параметром** → відкрити цей запит у режимі конструктора → у поле **Найменування клієнта** по рядку **Условия отбора** написати [**Уведіть найменування замовника:**] (рисунок 9.19) → зберегти змінений запит.

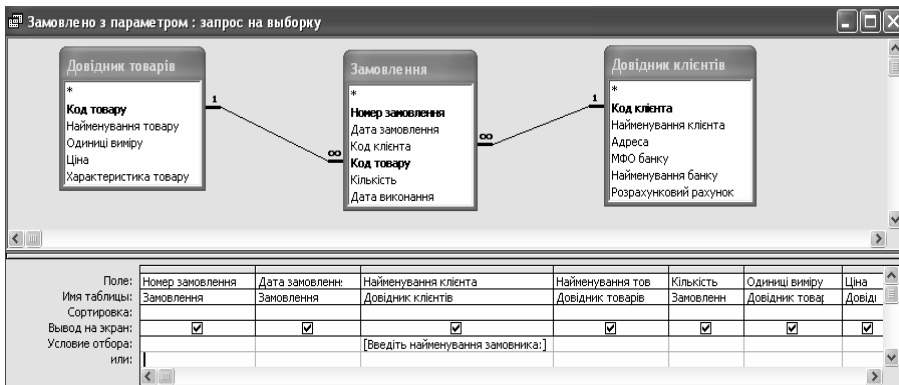



Рисунок 9.19 – Бланк запити з параметром

При виконанні даного запити (таблиця 9.12) користувачеві буде видаватися діалогове вікно для уведення найменування замовника, по якому необхідно зробити вибірку.

Аналогічно можна створити запит **Адреса клієнта** з параметром **Дата замовлення**.

На початку параграфу згадувався ще один тип запитів – запити **SQL**. Звичайно всі запити (інструкції одержання потрібних даних) зберігаються у вигляді інструкцій **SQL**. Кожен запит можна відкрити у конструкторі, на панелі інструментів розкрити список кнопки **Вид** , вибрати **Режим SQL** та побачити інструкцію на указаній мові (рисунок 9.20).

Таблиця 9.12. – Результат роботи запити **Замовлено з параметром** (параметр – **Найменування клієнта**)

Номер замовлення	Дата замовлення	Найменування клієнта	Найменування товару	Кількість	Одиниці виміру	Ціна
3	11.01.2005	ТОВ Мрія	маргарин сонячний	25	пачка, 250	1,10р.
5	12.01.2005	ТОВ Мрія	маргарин молочний	75	кг	3,40р.

Мова **SQL** розширює можливості користувача по створенню запитів, так як не усі типи запитів можна створити у режимі конструктора

(наприклад, підлеглі запити, результати яких використовуються у якості умов порівняння в других запитах). Звичайно режим SQL використовують користувачі середньої та високої кваліфікації.

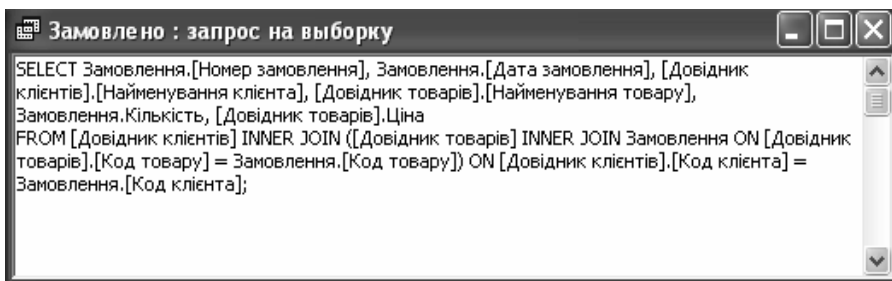


Рисунок 9.20 – Запит **Замовлено** у режимі SQL

## 9.2.4 Звіти


*Access* дозволяє створювати й друкувати звіти відповідно до потреб користувача. Можливості оформлення даних для виводу на друк майже на стільки ж гнучкі, як і можливості відображення їх на екрані.

Звіти дуже схожі на форми. Режими конструктора форм і звітів майже ідентичні. Однак між ними є істотне розходження – звіти призначені винятково для виводу даних на друк. У них відсутні елементи для уведення даних. Тому у звітах не використовують списків, полів із списками й прапорців.

В *Access* можна створити:

- прості,
- складові звіти й
- поштові наклейки.

Крім того, у звіт можна включити діаграму.

Звіти можна створювати за допомогою функції **Автоотчет**, **Мастера отчетов** або в режимі конструктора (рисунок 9.21). Повний перелік можливостей *Access* по створенню звітів викликається кнопкою **Создать** вікна бази даних  **Создать**.

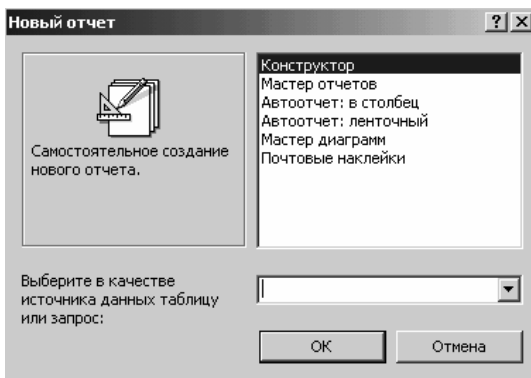


Рисунок 9.21 – Діалогове вікно для вибору способу створення звіту

Якщо вибрати **Автоотчет: у стовпець** або **Автоотчет: ленточный**, то на підставі зазначеної таблиці або запиту буде автоматично побудований звіт в один стовпець або у вигляді таблиці відповідно. При цьому всі поля джерела будуть обов'язково включені у звіт.

Більше гнучкий і в той же час простий шлях створення звіту: вкладка **Отчеты** → вибрати **Мастер отчетов** (або **Создание отчетов с помощью мастера**). Далі необхідно послідовно виконувати підказки майстра (див. приклад 9.13 і рисунки 9.22 – 9.26).

При використанні **Мастера отчетов** для формування звітів можна використати кілька таблиць або запитів і вибіркові поля з них (рисунок 9.22). При цьому не слід використовувати таблиці й запити, засновані на цих таблицях, СУБД не зможе їх зв'язати!

**Примітка** – Якщо для створення звіту використовується багато джерел, то раціональніше попередньо створити відповідний запит, а на підставі його вже друкувати звіт.

Для підрахунку підсумків у звіті необхідно вказувати ознаки для групування (наприклад, якщо потрібні підсумки по окремих товарах, то групування треба виконувати по полю «найменування товару», якщо усередині ще варто отримати підсумки по тимчасових періодах, то додаткова ознака для групування – «дата»). Можна вказати до 4-х полів, по яких буде здійснюватися групування даних (рисунок 9.23). Можна також використати інтервал групування. Якщо зазначено стандартну установку " **По полному значению** ", то автоматично виробляється



групування даних з однаковими значеннями. Для числових полів можна вказати розмір кроку. Завдання розміру кроку важливо для полів дати. У цьому випадку записи можна групувати по днях, тижнях, роках.

Усередині кожної групи можна задати порядок сортування по полях, що не беруть участь у групуванні (рисунок 9.24).

Для числових полів, для кожної групи можна виставити прапорець підрахунку підсумків по певній функції (наприклад, **Sum**).

Потім вибирають стиль оформлення звіту, орієнтацію паперу й можна задати прапорець **Расположить все поля на одной странице** (рисунок 9.25).

Звіти можна попередньо переглядати й виводити на друк стандартними для пакета Microsoft Office засобами.

Звіти можуть зберігатися в Word і Excel:

м. **Сервис** → к. **Связи с Office** → **Публикация в Ms. Word** або **Анализ в Ms. Excel**;

у текстовому форматі, форматі HTML,

у форматі SNP (як знімок для перегляду в браузері або програмі перегляду фотографій): м. **Файл** → к. **Экспорт** → указати **Тип файла**, його ім'я й папку для збереження → кнопка **Сохранить**.

Приклад 9.13. – Створити звіт **Сума замовлення**, у якому надрукувати суми по кожному найменуванню товару в розрізі дат і всього за місяць.

Виконання.

Аналіз наявних таблиць і створених запитів показує, що для формування необхідного звіту зручно використати запит **Сума замовлення**, який містить всі необхідні дані.

Для створення звіту викликати об'єкт бази даних **Отчеты** → подвійне клацання по рядку **Создание отчета с помощью мастера** → у вікні майстра (рисунок 9.22) у списку, що випадає, поля **Таблицы и запросы** вибрати **Сума замовлення** й вибрати всі поля за винятком **Дата виконання** → кнопка **Далее**.

На другому кроці майстра звітів у списку **Добавить уровни группировки** виділити **Найменування товару** → клацнути кнопку **>** → виділити **Дата замовлення** → клацнути кнопку **>** → клацнути по кнопці **Группировка ...** → задати для поля **Дата замовлення** інтервал групування **по місяцях** (рисунок 9.23) → **ОК** → кнопка **Далее**.

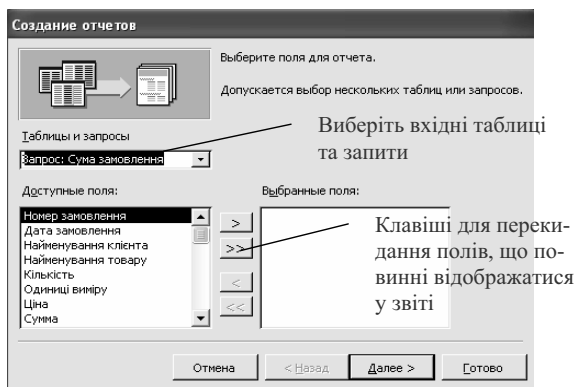


Рисунок 9.22 – Перше вікно майстра звітів

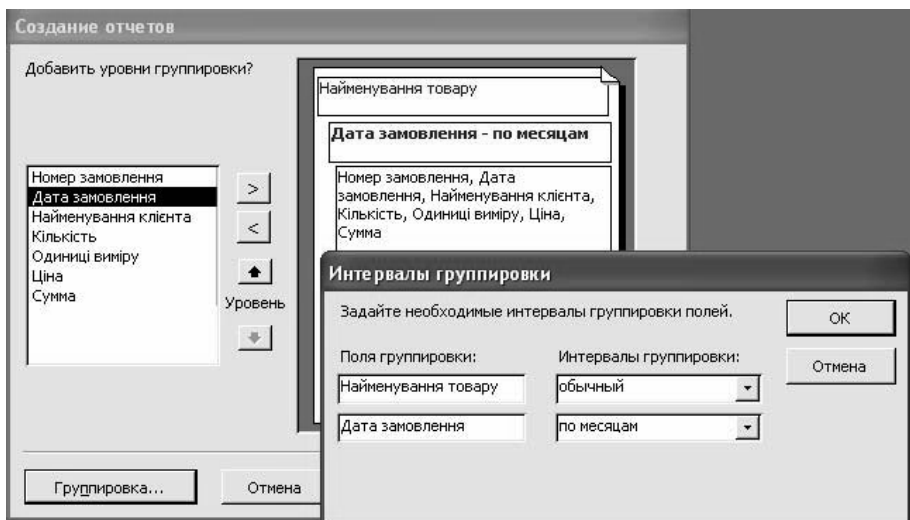


Рисунок 9.23 – Друге вікно майстра звітів

На третьому кроці необхідно вказати, що сортування записів усередині ознак для групування буде виконуватися по номеру замовлення (рисунок 9.24) → клацнути по кнопці **Итоги** → для полів **Кількість** і **Сума** виставити прапорці по стовпцю **Sum** → уключити селекторну кнопку **данные и итоги** → **ОК** → кнопка **Далее**.

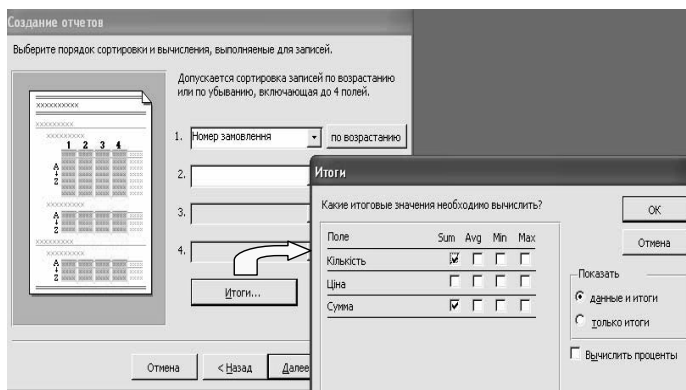


Рисунок 9.24 – Третье окно мастера звітів

У четвертому вікні **Майстра звітів** вибрати форму подання звіту **блок** (рисунок 9.25) → вибрати книжкову орієнтацію → виставити прапорець для розміщення звіту на одній сторінці по ширині → кнопка **Далее** → вибрати стиль оформлення звіту → кнопка **Далее** → увести заголовок звіту (**Сума замовлення**) → кнопка **Готово**. Одержимо звіт зазначеної форми (фрагмент звіту показаний на рисунку 9.26)

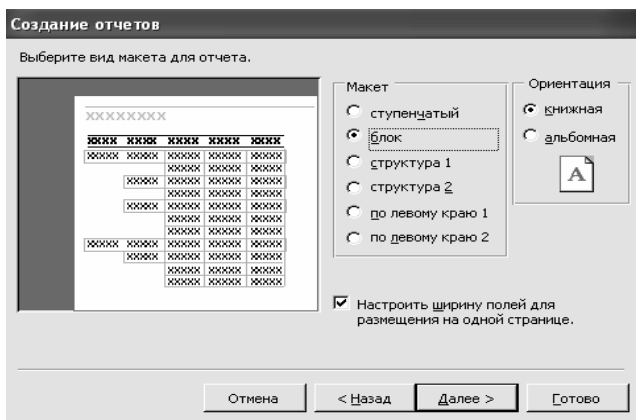


Рисунок 9.25 – Останній шаг майстра створення звітів

## Сума замовлення

Найменуван	Дата замо	вленн	мо	влення	мо	влення	Найменуван	ькість	Одосніці	вн	Ціна	сума	
маргарин вершок	Январь 2005	5	12.01.2005	ООО Мрія	100	пачка, 250					1,50р.	250	
		10	15.01.2005	Сіріоп	500	пачка, 250					1,50р.	250	
Итого для 'Дата замовлення' = 15.01.2005 (2 записей)													
<b>Sum</b>									600			500	
Итого для 'Найменування товару' = маргарин вершковий (2 записей)													
<b>Sum</b>									600			500	
маргарин молоч	Январь 2005	1	10.01.2005	МПП Веселка	100	кг					3,40р.	250	
		5	12.01.2005	ООО Мрія	75	кг					3,40р.	250	
		6	20.01.2005	ДМЗ	50	кг						3,40р.	250
		9	12.01.2005	ЧП Коркієнко	300	кг						3,40р.	250
Итого для 'Дата замовлення' = 12.01.2005 (4 записей)													
<b>Sum</b>									525			1000	
Итого для 'Найменування товару' = маргарин молочний (4 записей)													

Рисунок 9.26 – Фрагмент звіту для прикладу 9.15


При створенні звітів можна використовувати запити з параметрами, що дає можливість одержувати друкований документ у потрібному розрізі.

Майстер звітів практично повністю створює звіт. Якщо необхідно його подальше настроювання або звіт носить нестандартний характер і вимагає створення вручну, то варто використати засоби конструктора звітів *Access* (див. 9.2.5).

У СУБД *Access* передбачений спеціалізований звіт по створенню поштових наклейок, що зручно використовувати при дистанційному спілкуванні з клієнтами. Такий звіт створюється за допомогою майстра поштових наклейок, при цьому можна використовувати стандартні наклейки або створювані користувачем. Створення подібного звіту розглянемо на прикладі.

**Приклад 9.14.** – У базі даних **Замовлення** створити звіт поштових наклейок для клієнтів, що замовили товар.

Виконання. У базі даних **Замовлення** на підставі запиту на зміну створюється таблиця **Адреса замовника**, що містить необхідні дані. Саме ця таблиця й буде джерелом даних для звіту поштових наклейок.

- Для створення звіту викликати об'єкт бази даних **Отчеты** → кнопка **Создать**  → у поле джерела даних вибрати таблицю **Адреса замовника** → виділити рядок **Почтовые наклейки** → **ОК**.

2. У вікні **Создание наклеек** (рисунок 9.27 а) → указати необхідні параметри → кнопка **Далее** → вибрати параметри шрифту для наклеек (рисунок 9.27 б) → кнопка **Далее** → указати поля із джерела, які варто друкувати на наклейках, постійну адресу (для своєї фірми) можна внести вручну, сформувані зовнішній вигляд наклейки (рисунок 9.27 в) → кнопка **Далее** → можна вказати поле (наприклад, **Номер замовлення**) для сортування наклеек → кнопка **Далее** → дати ім'я звіту й кнопка **Готово**. Фрагмент звіту показаний на рисунку 9.28.

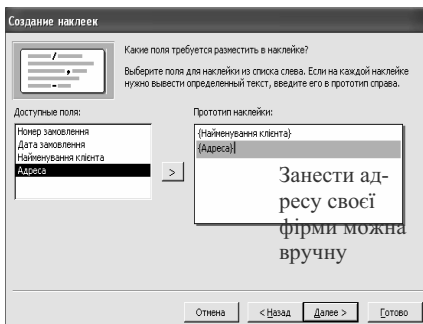
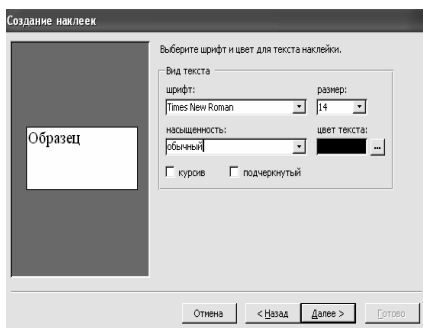
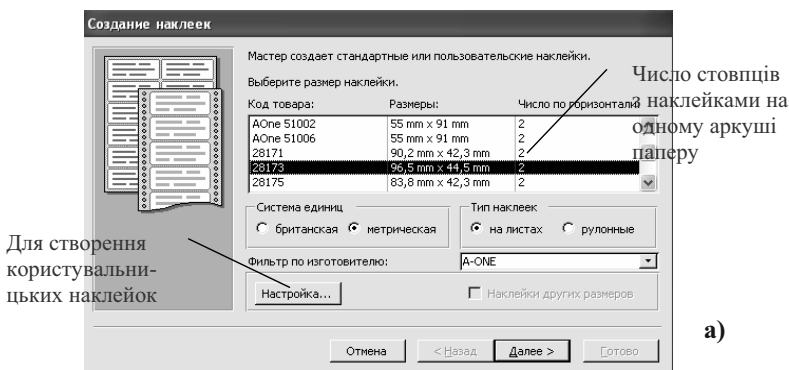


Рисунок 9.27 – Вікна Майстра поштових наклеек

ДМЗ г. Донецьк	МЧП Веселка г. Горлівка
ДонГУЕТ ім.М.Туган-Барановського 83050 м.Донецьк, вул.Щорса, 31	ДонГУЕТ ім.М.Туган-Барановського 83050 м.Донецьк, вул.Щорса, 31
МЧП Веселка г. Горлівка	ООО Мрія г.Харцизьк
ДонГУЕТ ім.М.Туган-Барановського 83050 м.Донецьк, вул.Щорса, 31	ДонГУЕТ ім.М.Туган-Барановського 83050 м.Донецьк, вул.Щорса, 31

Рисунок 9.28 – Фрагмент звіту поштових наклейок

### 9.2.5 Робота в режимі конструктора

У режимі **Конструктор** користувач одержує можливість створювати форми й звіти з порожнього листа або редагувати їх за своїм розсудом. При виконанні відповідних робіт використовуються поняття й елементи Visual Basic (об'єкти, властивості, події; написи, текстові поля, прапорці й т.д.). У видаваній на екран користувальницькій формі залежно від створюваного об'єкта можуть бути присутні різні області, перелік і призначення яких представлені в таблиці 9.13.

При створенні форми або звіту в конструкторі зручно користуватися віконцем із списком полів документа – джерела даних. У цьому випадку при перетаскуванні поля в електронну форму з'являється подвійний об'єкт – напис і відповідне текстове поле. При необхідності у формі можна працювати з кожним із цих елементів окремо або видалити який-небудь із них. Наприклад, перетаскуючи поле Номер замовлення, одержимо два об'єкти:



Якщо помістити курсор миші на чорний квадратик у лівому верхньому куті одного з об'єктів, то курсор прийме форму руки з витягнутим вказівним пальцем. У цьому випадку можна виконувати операцію переміщення одного об'єкта. Якщо курсор помістити на рамку виділеного об'єкта, то курсор прийме вид руки з випрямленими пальцями – можна виконати операцію переміщення парних об'єктів. Інші операції з

об'єктами такі ж, як і в Visual Basic. Кожний об'єкт і форма в цілому мають властивості, вікно яких можна викликати через контекстне меню. Приклад вікна властивостей для форми показаний на рисунку 9.29. У цьому ж вікні з об'єктами можна зв'язати події (вкладка **События**) і написати для них формули або підпрограми.

Таблиця 9.13. – Области режиму конструктора для створення форм і звітів

Назва області	Використовується для створення		Призначення	Примітка
	форми	звіту		
1	2	3	4	5
Заголовок	+	+	Заголовок друкується на початку форми (звіту), у розділ заголовка міститься назва об'єкта	Парні області – завжди додаються в парі
Примітка	+	+	Друкується наприкінці форми (звіту), у розділ приміток містяться підсумки по звіту в цілому	
Область даних	+	+	Відбиває зміст форми або звіту	
Верхній колонтитул	+	+	Розміщується на кожній сторінці форми (звіту), для звіту в ньому розташовують шапку таблиці	Парні області
Нижній колонтитул	+	+	Розміщається на кожній сторінці форми (звіту), для звіту в ньому розташовують номери сторінок, кількість сторінок у звіті, дату створення й ін.	
Заголовок групи		+	Розташовують поля – ознаки групування	Парні області, може бути декілька у звіті.
Примітка групи		+	Поміщають формули для підрахунку проміжних підсумків по ознаках, що групуються	



Рисунок 9.29 – Вікно властивостей області даних для електронної форми

Приклад 9.15. – У базі даних **Замовлення** відредагувати за допомогою конструктора форму **Замовлення** – відцентрувати текстові поля, збільшити поля зі списками, що випадають, щоб було видно в списках дві колонки, увести заголовок у форму **Замовлення клієнта**, вставити ілюстрацію на форму.

Виконання. 1. У базі даних викликати об'єкти **Форми** → виділити форму **Замовлення** → і клацнути по кнопці **Конструктор** → виділити клацанням поле зі списком, що випадає, **Код клієнта** й збільшити його у два рази (аналогічні дії виконати для поля **Код товару** (при натиснутій клавіші **Shift** послідовно виділити всі текстові поля форми й на панелі інструментів **Форматування** клацнути по клавіші **По центру**).

2. Для вставки заголовка форми додати область заголовка: м. **Вид** → к. **Заголовок/примечание формы** → в область заголовка додати об'єкт **напис** за допомогою кнопки **Аа** панелі інструментів → занести текст **Замовлення клієнта** → вийти з режиму занесення тексту, клацнувши за межами напису → виділити напис із заголовком і відформатувати з параметрами – **Times New Roman Cyr, розмір 14, По центру**.

3. Вставка ілюстрації на форму – за допомогою кнопки **🖼️** панелі інструментів в області даних окреслити місце для рисунка → у вікні, що відкрилося, **Вибір рисунка** вказати папку й ім'я файлу з рисунком → **ОК**. малюнок, що з'явився, може не міститися в зазначену область, необхідно змінити властивості об'єкта: контекстне меню по малюнку → к. **Свойства** → у рядку **Установка размеров** вибрати параметр **Впи-**



сать у рамку → закрити вікно властивостей → закрити вікно форми зі збереженням змін.

Вікно форми в режимі конструктора й у режимі виконання показано на рисунку 9.30.

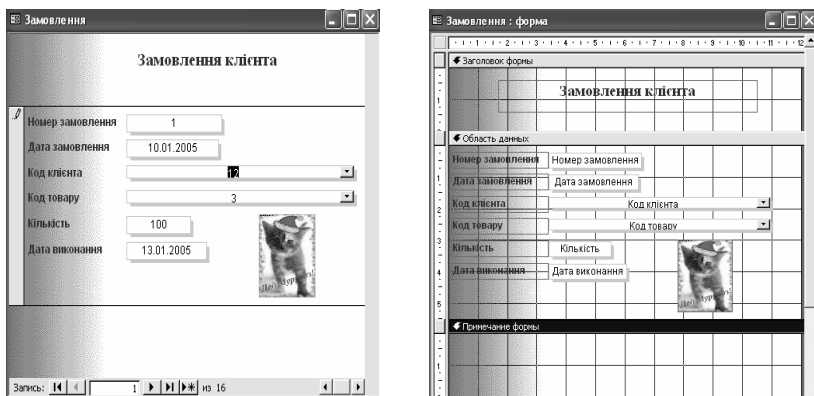


Рисунок 9.30 – Форма **Заказ** у режимі виконання та в режимі конструктора

**Приклад 9.16.** – У базі даних **Замовлення** за допомогою конструктора на підставі запиту **Сума замовлення** створити звіт **Дані про замовлення** з підсумками по кожній даті й у цілому по всіх замовленнях. У звіті проставити сторінки, наприкінці звіту виводити дату й час створення.

Виконання. 1. У базі даних викликати об'єкти **Отчеты** → кнопка **Создать** → як джерело даних указати **Сума замовлення** → виділити **Конструктор** і **ОК**.

2. В електронну форму звіту додати області заголовка й примітки: м. **Вид** → к. **Заголовок/примечание отчета** → в область заголовка додати об'єкт **напис** за допомогою кнопки **Аа** панелі інструментів → занести текст **Дані про замовлення** → вийти з режиму занесення тексту, клацнувши за межами напису → виділити напис із заголовком і відформатувати з параметрами – **Times New Roman Cyr**, розмір **14**, **По центру**.

3. Для визначення порядку формування предметної частини звіту необхідно вказати параметри групування й сортування: м. **Вид** → к. **Сортировка и группировка** або кнопка на панелі інструментів **≡** → з вікна списку полів джерела у вікно **Сортировка и группировка** перетягнути потрібні поля (**Дата замовлення** й **Найменування клієнта**) і

уточнити порядок сортування → виділити ознаку для групування **Дата замовлення** → в області **Свойства группы** по рядках **Заголовок группы**, **Примечание группы** виставити параметр **Да** → закрити вікно **Сортировка и группировка** (рисунок 9.31).

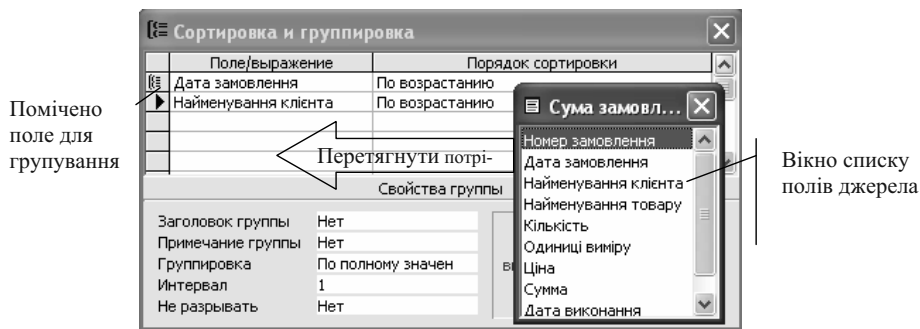


Рисунок 9.31 – Завдання полів групування у звіті

У результаті виконання третього пункту на електронній формі з'являться області **Заголовок группы** 'Дата замовлення' і **Примечание группы** 'Дата замовлення'.

4. В області **Верхний колонтитул** сформувати повторювану на кожній сторінці шапку документа з написів полів джерела (рисунок 9.32) – можна вручну або див. примітку нижче.

5. В область заголовка групи зі списку полів джерела перетягнути поле **Дата замовлення** → клацнути по об'єкту напису й видалити його → текстове поле, що залишилося перемістити в потрібну графу.

6. В **Область даних** у відповідні графи помістити текстові поля зі списку полів джерела, використовуючи той же прийом, що описано в п. 5.

Примітка 1. Прискорити створення документа можна, комбінуючи введення полів і їхніх написів в **Область даних**, у **Верхний колонтитул** і **Заголовок группы**. При цьому використовується особливість парних об'єктів – напис можна видаляти (вирізати) із пари, а при видаленні текстового поля видаляються обидва об'єкти. Таким чином, можна в область даних перетягувати поля, вирізати напис і вставляти її в область верхнього колонтитула.

**Примітка 2.** В область даних можна також поміщати нові поля, що розраховуються. Для цього у властивостях такого поля на вкладці **Данные** по рядку **Данные** внести необхідну формулу.

Заголовок отчета									
<b>Дані про замовлення</b>									
Верхний колонтитул									
Дата замовлення	Найменування клієнта	Номер замовлення	Найменування товару	Одиниці виміру	Кількість	Ціна	Сумма	Дата виконання	
Заголовок группы 'Дата замовлення'									
Дата замовлення									
Область данных									
Дата замовлення	Найменування клієнта	Номер замовлення	Найменування товару	Одиниці виміру	Кількість	Ціна	Сумма	Дата виконання	
Примечание группы 'Дата замовлення'									
="Разом по " & [Дата замовлення]							=Sum([Сума]) & "грн"		
Нижний колонтитул									
Поле2:		="Страница " & [Page]							
Примечание отчета									
="Усього за період"							=Sum([Сума])		
=Now()									

Рисунок 9.32 – Вікно **Конструктор** для звіту **Дані по закази**

7. В область примітки групи необхідно занести поля з указівкою ознаки для групування й формули для підрахунку підсумку: за допомогою інструмента **abc** вставити два текстові поля (рисунок 9.32) → для лівого поля через контекстне меню викликати к. **Свойства** → вкладка **Данные** → рядок **Данные** → кнопка з трьома крапками → у вікні **Построитель выражения**, що відкрилося занести **формулу** **"Разом по " & [Дата замовлення]** → **ОК** → закрити вікно властивостей.

Для другого поля в такий само спосіб занести формулу **=Sum([Сума]) & "грн"**.

**Примітка** – У подвійних лапках записуються текстові константи, у квадратних дужках – назва полів, значення яких використається в операції, & – операція конкатенації.

8. Для підрахунку загальних підсумків в область **Примечание отчета** скопіювати поля з області **Примечание группы ...** через буфер обміну → у ліве поле внести константу **«Усього за період»** → відформатувати поля (**Times New Roman Cyr**, розмір **14**, **полужирный**) →

вставити ще одне текстове поле ліворуч і внести в нього убудовану функцію друку поточної дати й часу =Now().


9. В область нижнього колонтитула внести номер сторінки: створити текстове поле й внести в нього з папки **Общие выражения** формулу = "Сторінка " & Page & " із " & Pages.

10. Мінімізувати відстані між областями форми для компактного розташування звіту на сторінках. Закрити вікно **Конструктора** → дати ім'я звіту **Дані про замовлення** → перевірити роботу звіту. На рисунку 9.33 показаний фрагмент готового звіту.

### Дані про замовлення

Дата замовлення:	Найменування клієнта:	Номер замовлення:	Найменування товару:	Одиниці виміру:	Кількість:	Ціна:	Сумма:	Дата виконання
Дата замовлення								
10.01.2005	МПП Веселка	1	маргарин молочний	кг	100	3,40р	20	13.01.2005
Разом по10.01.2005							20грн	
Дата замовлення								
11.01.2005	ООО Мрія	3	маргарин сонячний	пачка, 250	25	1,10р	20	14.01.2005
11.01.2005	Стірол	2	маргарин сонячний	пачка, 250	50	1,10р	20	13.01.2005
11.01.2005	ЧП Корнієнню	7	маргарин сонячний	пачка, 250	75	1,10р	20	12.01.2005
Разом по11.01.2005							60грн	

Рисунок 9.33 – Фрагмент звіту **Дані про замовлення**

Звіти в *Access* можна створювати на основі декількох зв'язаних таблиць і робити по них підсумкові обчислення. Для того щоб у головний звіт увести підлеглий треба: відкрити головний звіт у режимі конструктора → клацнути кнопку **Вікно бази даних**  панелі інструментів конструктора звітів для відображення на екрані вікна бази даних → перетягнути піктограму підлеглого звіту у відповідну область головного звіту → при необхідності можна відформатувати елементи → для зв'язку записів головного звіту й підзвіту, потрібно встановити відповідні властивості **Подчиненные поля** й **Основные поля** у вікнах властивостей головного й підлеглого звітів.

У режимі перегляду можна побачити, що *Access* розмістив під кожним полем головної таблиці відповідні поля підлеглої таблиці.

**Примітка** – Якщо у базі даних уже встановлений зв'язок між таблицями, то при створенні підлеглого звіту ці властивості будуть задані автоматично.

### 9.3 Сервісні функції пакета

СУБД *Access* надає користувачеві велику кількість сервісних функцій. Більша частина їх зосереджена в м. **Сервіс.**, частина перебуває в м. **Файл (Внешние данные, Экспорт, Свойства базы данных)**, у м. **Справка (Найти и устранить)**.

При створенні об'єктів бази даних СУБД може допомогти створити нормалізовані таблиці з таблиці з повторюваними значеннями полів: м. **Сервіс** → к. **Анализ** → п. **Таблица** → працювати з вікнами майстра.

Через м. **Файл** і к. **Внешние данные** реалізуються можливості імпорту або встановлення зв'язків з іншими базами даних або файлами.

При імпорті *Access* створює копію вхідної таблиці. Зв'язок між файлом і імпортованою таблицею губиться. Якщо потім СУБД здійснюються зміни в цій таблиці, то вони не зроблять впливу на вхідну таблицю й навпаки. При приєднанні, навпроти, вхідний файл у форматі, відмінному від *Access*, залишається унікальним. До нього можуть звертатися як програма, за допомогою якої він створений, так і *Access*.

Імпорт даних рекомендується використовувати в тих випадках, коли користувач має дані у форматі, що відрізняється від *Access*, але в подальшому він хоче управляти ними за допомогою СУБД. Для імпорту даних насамперед треба відкрити базу, у яку треба додати дані, а потім м. **Файл** → к. **Внешние данные** → п. **Импорт** → указати папку, тип файлу й файл → кнопка **Импорт** → виділити об'єкти, при необхідності уточнити параметри (кнопка **Параметры >>**) для імпорту або поспідовно заповнювати вікна майстра імпорту → **ОК**.

Зв'язування даних: м. **Файл** → к. **Внешние данные** → п. **Связь с таблицами**.

Експорт варто застосовувати в тих випадках, коли інформацію, збережену в базі даних *Access*, необхідно зробити доступною для зовнішньої програми. Для цього варто відкрити базу даних із таблицями, які потрібно експортувати → виділити таблиці → м. **Файл** → к. **Экспорт** → указати папку, тип файлу й дати ім'я файлу, що зберігається.

Office надає послуги по зв'язках із програмами Office: меню **Сервіс** → к. **Слияние с MS Word (Публикации в MS Word, Анализ в MS Excel)**, які дозволяють перекидати таблиці в *Word* і *Excel*.

Програма *Access* дозволяє створювати на Web-сторінці електронну таблицю, що відображує дані, що зберігаються в базі даних. Між таблицею й базою даних устанавлюється зв'язок. В електронній таблиці можна створювати формули, що використовують для обчислень значення з бази даних. Якщо значення в базі даних змінюються, електронна таблиця перераховується.

Після створення на Web-сторінці електронної таблиці, цю сторінку можна вдосконалювати за допомогою тієї або іншої програми-конструктора – *Microsoft Excel 2000*, *Microsoft Access 2000*, або *Microsoft FrontPage 2000*. Програмісти можуть також створювати редагувати електронні таблиці в *Microsoft Visual Basic*.

*Microsoft Access* надає користувачеві засіб по оптимізації роботи бази даних – аналізатор швидкодії: м. **Сервіс** → к. **Анализ** → п. **Быстрое действие** → Вибрати потрібну вкладку → виділити клацанням об'єкти, які потрібно оптимізувати → кнопка **Выделить** або **Выделить все** → **ОК**.

Аналізатор швидкодії видає три типи рекомендацій з оптимізації продуктивності: ради, пропозиції й думки. Ради й пропозиції *Microsoft Access* може виконувати автоматично (виділити потрібне й клацнути по кнопці **Оптимизировать**), а рекомендації типу «думка» виконуються вручну.

Одержати інформацію про властивості об'єктів бази даних можна через м. **Сервіс** → к. **Анализ** → п. **Архивариус**.

При переході на нову версію програми вже створені бази даних необхідно перетворити за допомогою м. **Сервіс** → к. **Служебные программы** → п. **Преобразовать базу данных**.

При роботі з базою даних необхідно передбачити засоби від утрати інформації та засоби її захисту.

1. Створення резервної копії за допомогою програми – архіватора.
2. Відновлення бази даних із резервної копії – скопіювати резервний файл у папку бази даних.

Варто перевірити схоронність інформації, що вводилася в момент зависання. Якщо *Access* не може успішно завершити відновлення, то рішення проблеми тільки у використанні резервної копії.

**Стиск бази даних.** У процесі роботи з Access користувач створює й обробляє нові таблиці, форми, запити або звіти. Крім того, поступово додаються нові записи, редагуються й віддаляються наявні записи. При видаленні з бази даних запис маркірується відповідним чином, але не видаляється фізично. Нові записи додаються в кінець файлу. У результаті такої організації у mdb- файлах утвориться простір, що не використовується, на обробку якого, наприклад, при пошуку запису, витрачається час. Крім того, розмір файлу стає невиправдано більшим. Тому необхідно періодично проводити стиск бази даних: м. **Сервіс** → к. **Службові програми** → п. **Сжать и восстановить базу данных.**

Після реорганізації розмір файлу бази даних зменшується.

**Шифрування бази даних.** При транспортуванні бази даних для захисту інформації її можна зашифрувати. Для цього треба відкрити вікно бази даних Access, потім:

м. **Сервіс** → к. **Защита** → п. **Шифровать/дешифровать** → вибрати потрібну базу даних → **ОК** → указати ім'я зашифрованої бази даних → **ОК**.

Шифрування приводить до того, що база даних може бути надалі прочитана тільки в Access. Перегляд даних іншими програмами виключений.

Розшифровка бази даних іде аналогічним образом, тільки вказується папка й ім'я зашифрованої бази. СУБД автоматично розпізнає зашифровану базу даних і робить її розшифровку.

## Контрольні питання

1. У чому відмінність бази даних від СУБД?
2. Чим відрізняються таблиці Access від таблиць текстових редакторів і табличних процесорів?
3. Як класифікуються СУБД?
4. Назвати об'єкти бази даних Access і дати їхню характеристику.
5. Назвіть способи створення таблиць.
6. Що таке схема даних і її призначення?
7. Які типи даних використовуються в Access?
8. Призначення ключових полів, чим вони відрізняються від індексованих полів?
9. Способи введення даних у таблиці?
10. Які типи запитів можна створювати в Access?
11. Умови відбору: призначення й правила формування?

12. Які розходження у формуванні запитів із полями, що розраховуються, і підсумкових запитів?
13. Як сформулювати запит, у якому по тому самому полю необхідно виконувати обчислення й здійснювати відбір за умовою?
14. Види форм, які можна створити в *Access*?
15. Способи захисту даних при роботі користувача з формою?
16. Які типи звітів можна створювати в *Access* ?
17. Призначення областей електронної форми для створення звітів у режимі конструктора?
18. Як змінити ширину стовпців у звіті *Access*?
19. Способи подання й друку даних із бази *Access*?

## Завдання для самостійного виконання

1. Додати в базу даних **Замовлення** таблицю **Оплата й виконання** з полями **Номер замовлення, Код клієнта, Дата виконання, Дата оплати, Сума оплати, Дата відвантаження** (фрагмент – у таблиці 9.14). Передбачити різні варіанти оплати: оплачено, не оплачено, оплачено не повністю.

Таблиця 9.14. – **Оплата й виконання**

Номер замовлення	Код клієнта	Дата виконання	Дата оплати	Сума оплати	Дата відвантаження
1	12	13.01.2005	11.01.2005	150,00р.	13.01.2005
2	15	13.01.2005	11.01.2005	55,00р.	13.01.2005
3	11	12.01.2005	12.01.2005	27,50р.	14.01.2005
5	11	12.01.2005	12.01.2005	405,00р.	14.01.2005
7	13	12.01.2005	11.01.2005	82,50р.	12.01.2005

2. Створити форму для заповнення таблиці **Оплата й виконання**, у якій поля **Номер замовлення** й **Код клієнта** заповнюються тільки на підставі дані таблиці **Замовлення**.
3. Створити запит **Разом замовлення**, у якому визначається загальна сума замовлення. Такий запит необхідний при наявності багаторядкових замовлень.
4. Створити за допомогою конструктора запит на вибірку **Виконання замовлення**, що поєднує дані запиту **Разом замовлення** й таблиці **Оплата й виконання**, зберегти й виконати його. Фрагмент результату виконання показаний у таблиці 9.15.



Таблиця 9.15. – Фрагмент результату роботи запиту **Виконання замовлення**

Номер замовлення	Дата замовлення	Найменування клієнта	Sum-Сума	Дата оплати	Сума оплати	Дата виконання	Дата відвантаження
1	10.01.2005	МЧП Веселка	150,00р.	11.01.2005	150,00р.	13.01.2005	13.01.2005
2	11.01.2005	Стирол	55,00р.	11.01.2005	55,00р.	13.01.2005	13.01.2005
3	11.01.2005	ТОВ Мрія	27,50р.	12.01.2005	27,50р.	14.01.2005	14.01.2005
5	12.01.2005	ТОВ Мрія	405,00р.	12.01.2005	405,00р.	15.01.2005	14.01.2005
7	11.01.2005	НП Корнієнко	82,50р.	11.01.2005	82,50р.	12.01.2005	12.01.2005

5. Створити звіт на підставі запиту з параметром **Замовлено з параметром по ТОВ Мрія**.

6. Створити звіти: **Реєстр оплачених замовлень, Реєстр неоплачених замовлень**.

## Література

1. Вейскас Дж. Эффективная работа с Microsoft Access 2000. СПб: Издательство «Питер», 2000. – 1040 с.
2. Дженнингс Р. Использование Microsoft Access 97: Пер. с англ. – 2-е изд. – К.; М.: СПб.: Издат.дом»Вильямс», 1998. – 944 с.
3. Чаповська Р., Жмуркевич А. Робота з базами даних Microsoft Access 2000. Навчальний посібник для студентів економічних спеціальностей. – Київ: ЦУЛ, 2003. – 324 с.

## Розділ 10. Використання Visual Basic для автоматизації робіт у Microsoft Office

У попередніх розділах користувач познайомився з основами програмування на Visual Basic, що є не просто мовою програмування, але ще і мовою програмування для більшості компонентів Microsoft Office. Різновидність Visual Basic для роботи з додатками Microsoft Office одержала назву **Visual Basic for Applications (VBA)**.

Автоматизація робіт безпосередньо в Office для підприємства або організації обумовлена необхідністю:

- забезпечити однаковість оформлення різних документів різними виконавцями;
- економії часу на створення тих самих форм документів;
- зниження вартості документообігу за рахунок економії робочого часу користувачів по створенню форм документів.

У VBA для роботи використовується те ж інтегроване середовище проектування з відомими уже вікнами: вікно редактора Microsoft Visual Basic (аналог вікна проектування), вікно проекту **Project**, вікно властивостей об'єктів **Properties**, вікно програмного коду **Code**, вікно екранної форми **Form1**, вікно інструментів **Toolbox**. Основні оператори і правила їхнього запису цілком відповідають раніше вивченим.

Розходження в основному стосуються переліку об'єктів, їхніх властивостей і використовуваних методів, що відповідають додаткам Microsoft Office.

Для автоматизації робіт у додатках Office можна використовувати **макроси (макрокоманди) і VBA**.

**Макроси** – це послідовність процедур, що автоматично виконує додаток, при натисканні визначеної комбінації клавіш або клацанні миші. За допомогою макросу Word або Excel може зберегти послідовність виконуваних вами дій. Створення макросу можна виконувати за наступною схемою:

М. Сервіс → к. Макрос → п. Начать запись → дати ім'я макросу → визначити місце його збереження → призначити кнопку або комбінацію клавіш → виконати необхідні дії у документі → клацнути по кнопці **Остановить запись**

При роботі з документом користувач клацанням миші може запустити цю послідовність.

При зазначеному варіанті роботи програмний код записується автоматично, для його зміни, усунення помилок, користувачеві прийде все рівно звернутися до **VBA**. Крім того, нижче перераховані операції також жадають від користувача знань мови програмування:

1) для створення документа необхідно запитувати інформацію в користувача;

2) при формуванні або форматуванні тексту необхідно здійснювати вибір – форматувати деякі таблиці, у залежності від умов готувати листи протилежного змісту і т.п.;

3) необхідно виконувати обробку декількох документів, працювати з іншими програмами;

4) необхідність ретельного контролю й тестування.

Таким чином, свідомо й ефективно автоматизація робіт у додатках вимагає застосування знань **Visual Basic for Applications**. У той же час, чим більше убудованих можливостей додатка використовує користувач, тим менше прийдеться програмувати.

## 10.1 Автоматизація робіт із документами у Microsoft Word за допомогою VBA

Перед тим як почати створення свого автоматизованого додатка, користувач повинен чітко визначити усі вимоги до майбутніх документів, а також оцінити переваги автоматизації їхнього формування. Наприклад, варто визначити:

- *параметри документа* – вимоги до розміру, орієнтації сторінок, полям, верхнім і нижнім колонтитулам, нумерації сторінок;

- *вимоги до форматування і стилі* – шрифти, їхні розміри, розташування різних частин документа, елементи нумерації, види міток і інші характерні риси документа;

- *наявність спеціальних елементів* – графіка, таблиці, текст, що повторюється, і т.д.;

- *інформацію, що вводиться користувачем*, – текст у довільному виді або у визначеній послідовності й у визначені місця.

Якщо досить добре відомі параметри сторінок і правила форматування документа, але не відомі специфічні вимоги до тексту або ж не відомо, де саме повинен розміщатися текст, – у такому випадку, швидше за все, потрібно створити шаблон, що визначає правила форматування. При цьому можна обійтися без або малою кількістю програмного коду VBA. З іншого боку, якщо вимоги до даних, що вводяться, досить строгі й абсолютно негнучкі, додаток тільки виграє від застосування користувальницьких форм, що дозволяють уводити дані тільки строго визначеного типу, а також від VBA-програми, що розташує їх у визначених частинах документа.

Один із ключових моментів розробки комплексного додатка в Word полягає в тім, що перш ніж приступити до написання першого рядка програми, необхідно змусити *Word* проробити якнайбільше роботи. Такий підхід дозволить сфокусувати основні зусилля по програмуванню саме на задачах, що дійсно вимагають автоматизації. При цьому зменшується обсяг програмного коду, що прийдеться написати вручну, а так само спрощується сам процес внесення яких-небудь виправлень, якщо раптом вимоги до документа зміняться.

Після формулювання усіх вимог до додатка можна приступити до процесу розробки, почавши зі створення шаблону документа. Шаблон визначає форматування й параметри сторінок у додатку, а також містить коди макрокоманд і користувальницькі форми, що ви будете застосовувати (див. розділ 3).

Велика частина додатка може бути убудована в шаблон без написання коду VBA. Розглянемо автоматизацію створення такого документа як записка у Вузi. Приклад службової записки приведений на рисунку 10.1. При створенні шаблону цього документа варто врахувати наступне:

- у верхній частині документа повинні відображатися назва Вузу й аббревіатура на задньому плані, нижче праворуч – кому (назва посадової особи і його прізвище);
- ще нижче ліворуч вказується тема записки;
- посередині вказується назва записки – службова, доповідна;
- далі повинен іти текст записки;
- унизу записки – посада й прізвище виконавця, на наступному рядку – дата створення документа;
- шрифт Times New Roman, розмір – 14, міжрядковий інтервал одинарний, текст вирівняний по лівому краю.

Як видно з приклада поля: «Кому», «ПІБ кому», «Тема», «тип записки», «підпису», «дата» розташовані у визначених місцях, але зміст їх може мінятися; назва Вузу – постійний зміст і місце розташування; текст самої записки – довільний багаторядковий текст.

До створення шаблону можна приступити, почавши зі звичайного документа. Створіть документ, наприклад такий, як Ви бачите на рисунку 10.1, щоб розташувати усі дані в потрібних місцях листа. Потім варто змусити Word поміщати дані у визначені місця. Для цього користувач повинний буде вносити ці дані в спеціальну форму, відкіля вони потраплять для збереження в змінні. Після чого програма повинна розмістити їх у зафіксовані позиції документа за допомогою кодів полів **DocVariable**.

Щоб уставити код поля, необхідно:

установити курсор у те місце шаблону, куди варто помістити код поля → м. **Вставка** → к. **Поле** → у списку **Категорії** вибрати параметр **Автоматизація документа** → у списку **Поля** вибрати параметр **DocVariable** → у текстовому полі **Код поля** після напису **DocVariable** у лапках увести ім'я змінної для відображення → **ОК**.

У документі нічого не буде видно. Щоб побачити вставлене поле, необхідно переключитися в режим відображення кодів полів: [**Alt + F9**]. Вихід із цього режиму – повторне натискання [**Alt + F9**] (рисунок 10.2). У режимі відображення кодів полів загальний вид документа може змінитися, щоб цілком показати коди. При виході з цього режиму перекручування буде виправлено.

Донецький державний університет економіки і торгівлі  
ім. М.Туган – Барановського

**ДОНДУЕТ**  
Зав. кафедри комп'ютерних  
технологій  
О.В. Оліфірову

Тема: Про створення форми записки

### Службова записка

Доводжу до Вашого відома, що в процесі поточної роботи є можливість користуватися шаблоном по створенню службових та доповідних записок, які автоматично забезпечують одноманітність форми типа документів, яка створюється.

Доцент кафедри комп'ютерних технологій

Н.М. Войтюшенко

25.10.2005

Рисунок 10.1 – Форма записки у ВНЗ

#### Примітка –

Щоб у полях «Кому» і «ПІБ кому» другий рядок тексту правильно вирівнювався ліворуч у режимі відображення кодів полів за допомогою табуляції перемістіть продовження коду в потрібну крапку.

Донецький державний університет економіки і торгівлі  
ім. М.Туган – Барановського

**ДОНДУЕТ**  
{ DOCVARIABLE"Кому" \\* MERGEFORMAT }  
{ DOCVARIABLE"ПІБ" \\* MERGEFORMAT }

Тема: { DOCVARIABLE"Тема" \\* MERGEFORMAT }

**{ DOCVARIABLE"Тип" \\* MERGEFORMAT }**  
**записка**

{ DOCVARIABLE"посада" \\* MERGEFORMAT }

{ DOCVARIABLE"Дата" \\* MERGEFORMAT }

Рисунок 10.2 – Записка в режимі відображення кодів полів

Для швидкого й безпомилкового переходу VBA-програми в потрібне місце документа (у нашому випадку це місце введення тексту) необхідно використовувати закладки. Щоб створити закладку, треба виконати наступні дії:

- указати місце для закладки – помістити туди курсор чи виділити будь-який текст;

- м. **Вставка** → к. **Закладка** → увести ім'я закладки в поле **Ім'я за-кладки** → кнопка **Добавить**.

Для записки необхідно курсор поставити на місце початку введення тексту документа й установити закладку з ім'ям **Текст**.

Шаблон варто зберегти звичайним способом (із розширенням .dot – тип файлу «Шаблон документа») з ім'ям **Шаблон записки**.

Наступний етап – *розробка користувальницького інтерфейсу*, що дозволить користувачеві водити інформацію, не задумуючись про її правильне розташування в документі. Можна також передбачити контроль введення даних і параметрів, що виставляються. Для цього використовуються *користувальницькі діалогові* вікна чи *форми*, що розробляються за допомогою редактора Visual Basic.

Як і у звичайних програмах Visual Basic, спочатку варто розробити саму форму, а потім написати необхідний код. Для нашого приклада можна використовувати форму, показану на рисунку 10.3.

Рисунок 10.3 – Електронна форма для введення інформації записки

Тип записки повинний вибиратися за допомогою списку, що розкривається. Дата формування документа повинна проставлятися автоматично по таймеру, але допускати коректування.

Створюється така форма в редакторі Visual Basic у такий спосіб:



- 1) Відкрийте шаблон записки → перейдіть у редактор Visual Basic натисканням клавіш [Alt+F11] чи п. м. **Сервіс** → к. **Макрос** → п. **Редактор Visual Basic** (рисунок 10.4).
- 2) Знайдіть на екрані вікно **Project** або відкрийте його через меню **Вид**.
- 3) Виберіть **TemplateProject**, що відповідає шаблону, із яким працюєте.
- 4) м. **(Insert) Вставка** → к. **UserForm** – нова форма буде добавлена в проект і відкрита для редагування.

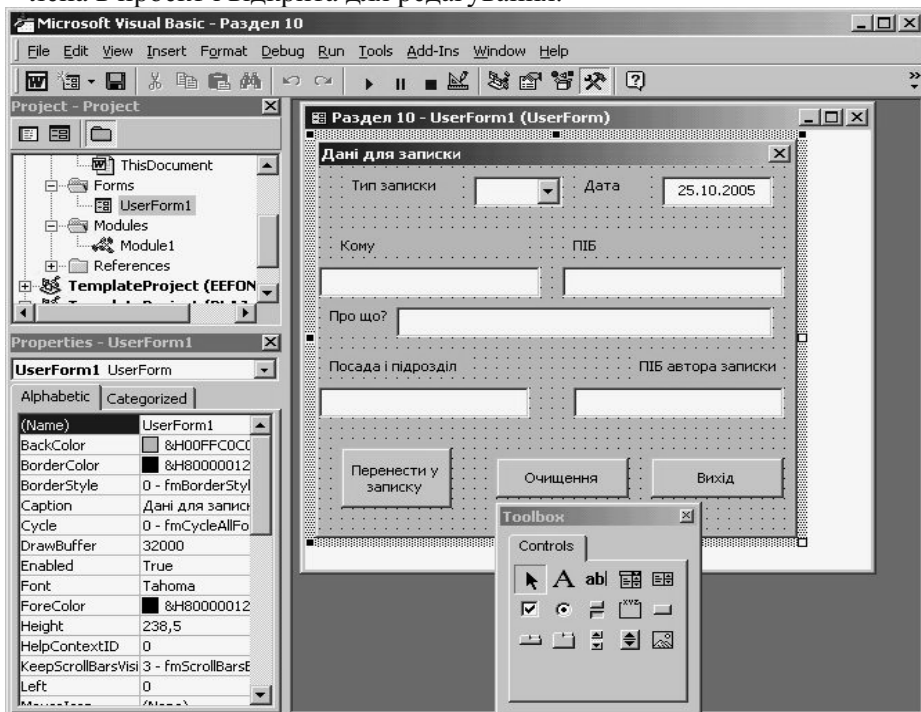



Рисунок 10.4 – Вікно редактора Visual Basic із формою для записки

Далі проектування форми виконується звичайним способом. Приласнюємо імена (властивість **Name**) формі, полю зі списком, текстовим полям і командним кнопкам, а так само вносимо текст (властивість **Caption**) в етикетки і підписуємо кнопки. При необхідності, для всіх полів установлюємо тип шрифту **Times New Roman** і необхідний розмір (властивість **Font**)

Name	Caption
1. Записка	Дані для записки
2. Tipe	Тип записки
3. Data	Дата
4 Кому	Кому
5 FIO	ПІБ
6. Tema	Про що
7. Kto	Посада й підрозділ
8. FIO_kto	ПІБ автора записки
9. У записку	Перенести у записку
10. Очищення	Очищення
11. Вихід	Вихід

Примітка – Список, що розкривається, додається за допомогою  кнопки на панелі інструментів.

*Друга частина проектування* складається з написання:

а) процедури, що буде запускатися при створенні нового документа;

б) процедури скасування форми;

в) процедури, що працює при заповненні форми.

Пункт в) створюється аналогічно раніше розглянутим прикладам **Visual Basic** з урахуванням використовуваних об'єктів і специфічних для Word процедур і методів. Так при кожній завантаженні форми (об'єкт **UserForm**) виконується процедура **Initialize**. Ця процедура використовується для виконання будь-яких попередніх дій по підготовці форми до роботи користувача.

Так у формі записки в списку, що розкривається, повинні бути показані два можливих варіанти – службова й доповідна. Додавання кожного варіанта виконується за допомогою методу:

**AddItem** <"назва варіанту" >

Поля **Кому**, **FIO**, **Tema**, **Kto**, **FIO\_kto** спочатку повинні бути порожніми (властивість **Text** у формі варто очистити).

Як початкове значення в поле **Data** повинна бути встановлена точна дата у відповідному форматі (форматований вивід даних).

Текст процедури **Initialize** наступний:

```

Private Sub UserForm_Initialize()
    Tipe.AddItem "Службова"
    Tipe.AddItem "Доповідна"
    Data.Text = Format(Date, "dd.mm.yyyy")
End Sub

```

Далі необхідно з кожною командною кнопкою зв'язати відповідний програмний код.

Клацання по кнопці **Очищення** повинне викликати знищення всіх даних у текстових полях. Код процедури наступний:

```

Private Sub Очищення_Click()
    Tipe.Text = ""
    Data.Text = ""
    Кому.Text = ""
    ФІО.Text = ""
    Тема.Text = ""
    Ктo.Text = ""
    ФІО_кto = ""
End Sub

```

Клацання по командних кнопках **Перенести у записку** або **Вихід** повинний закрити форму, викликати перенесення даних у записку й чекання введення тексту чи закриття активного документа без збереження.

Для визначення, по якій же кнопці клацнув користувач і які операції виконувати далі, необхідно оголосити глобальну перемінну **CreateZap** (ця перемінна буде використовуватися й в інших модулях) у стандартному модулі шаблону – м. **Insert** → к. **Module**:

### **Public CreateZap As Boolean**

Якщо перемінна **CreateZap** приймає значення **False** (неправда), то користувач клацнув по кнопці **Вихід**, якщо значення **True** (істина) – по кнопці **Перенести у записку**.

Закрийте вікно модуля і поверніться у вікно коду форми.

Тепер можна написати код програми для кнопок, що залишилися.

При клацанні по кнопці **Вихід** варто сховати форму і привласнити перемінної **CreateZap** значення **False**. Код програми наступний:

```

Private Sub Вихід_Click()
    Записка.Hide
    CreateZap = False
End Sub

```

При клацанні по кнопці **Перенести у записку** необхідно перевірити чи всі поля форми заповнені. Повноту заповнення полів можна фіксувати за допомогою перемінної, наприклад, **Gotovo**. Якщо хоча б одне поле не заповнене, то перемінна **Gotovo** приймає значення **False** і користувач повинний одержати повідомлення, інакше **Gotovo = True** і форму можна сховати. Код програми наступний:

```
Private Sub Y_записку_Click()  
    Dim Gotovo As Boolean  
    Gotovo = True  
    If Len(Tipe.Text) = 0 Then Gotovo = False  
    If Len(Data.Text) = 0 Then Gotovo = False  
    If Len(Кому.Text) = 0 Then Gotovo = False  
    If Len(FIO.Text) = 0 Then Gotovo = False  
    If Len(Тема.Text) = 0 Then Gotovo = False  
    If Len(Кто.Text) = 0 Then Gotovo = False  
    If Len(FIO_кто.Text) = 0 Then Gotovo = False  
    If Gotovo = True Then  
        Записка.Hide  
        CreateZap = True  
    Else  
        MsgBox "Необхідно ввести інформацію у всі поля форми", vbOKOnly +  
vbExclamation, "Помилка заповнення форми"  
    End If  
End Sub
```

Примітка – Оператор **MsgBox** ... пишеться в один рядок.

Заповнено поле чи ні перевіряє функція **Len**, що визначає довжину введеного рядка.

Процедури, зв'язані із заповненням електронної форми, не забезпечують перенесення даних у документ, та й сама форма повинна з'явитися при виклику відповідного шаблону. Для виконання зазначених операцій сам шаблон повинний містити програмний код.

Кожен шаблон містить модуль з ім'ям **ThisDocument**, що вкладений у папку **Microsoft Word Об'єкти** вікна проекту. Подвійне клацання на **ThisDocument** викликає модуль документа, куди будуть записуватися необхідні процедури.

У першу чергу, при створенні нового документа по нашому шаблоні на екрані повинна з'являтися форма для заповнення даними, що

можна зробити за допомогою методу **Show**. Потім процедура повинна перевірити значення перемінної **CreateZap** і визначити, по якій командній кнопці було клацання. Якщо по кнопці **Перенести у записку**, то кодам полів документа необхідно привласнити значення відповідних полів форми, використовуючи звертання до створюваного документа, як до активного:

**ActiveDocument.Variables (“<ім'я поля в документі”)** = **<Ім'я форми>. <ім'я поля>. <властивість>**

Усі поля в документі оновлюються й відображають поточне значення. Електронна форма повинна зникнути з екрану, а курсор – переміститися на закладку **Текст** для введення користувачем тексту записки.

Якщо клацання було виконано по кнопці **Вихід**, то документ повинний бути закритий без збереження.

Для виконання зазначених дій варто відкрити модуль **ThisDocument**, для об'єкта **Document** вибрати подію **New** і записати наступну процедуру:

```
Private Sub Document_New()  
'Оголошення перемінної для відновлення полів  
Dim fld As Field  
Записка.Show  
If CreateZap = True Then  
    Записка.Hide  
    'Присвоєння нових значень із форми полям документа  
    ActiveDocument.Variables("Тип") = Записка.Тип.Text  
    ActiveDocument.Variables("Кому") = Записка.Кому.Text  
    ActiveDocument.Variables("ПІБ") = Записка.FIO.Text  
    ActiveDocument.Variables("Тема") = Записка.Тема.Text  
    ActiveDocument.Variables("Посада") = Записка.Кто.Text  
    ActiveDocument.Variables("Підпис") = Записка.FIO_kto.Text  
    ActiveDocument.Variables("Дата") = Записка.Data.Text  
    'Відновлення всіх полів документа  
    For Each fld In ActiveDocument.Fields  
        fld.Update  
    Next  
    'Перехід на закладку для введення тексту записки  
    ActiveDocument.Bookmarks("Текст").Select  
    'Вивантаження електронної форми з пам'яті  
Unload Записка
```

```
Else
'Користувач клацнув по кнопці "Вихід"
Unload Записка
ActiveDocument.Close SaveChanges:=False
End If
End Sub
```

Після цього збережіть програмний код і створіть документ на основі розробленого шаблону:

У текстовому редакторі *Microsoft Word*

м. **Файл** → к. **Создать** → виділити позначку **Шаблон записки** → **ОК** → заповнити поля в електронній формі, що з'явилася, (клацнути по **кнопці Перенести в записку**) → увести текст записки → зберегти записку зі своїм ім'ям у потрібній папці.

Аналогічним чином можна автоматизувати створення інших документів.

## Питання й завдання для самостійної роботи

1. Які причини можуть викликати необхідність автоматизації створення документів у *Word*?
2. Що загальне і яке розходження між Visual Basic і Visual Basic for Application?
3. Критерії вибору VBA для автоматизації формування документа?
4. Послідовність робіт при автоматизації формування документа?
5. Як створити шаблон у *Word*?
6. Які основні моменти повинні бути відбиті в програмному коді шаблону для об'єкта Document, подія New?
7. Додайте в програмний код розглянутого приклада по формуванню записки додатковий тип у список, що випадає – Пояснювальна.
8. Автоматизуйте за допомогою VBA формування:
  - а) заяви про прийом на роботу;
  - б) заяви на відпустку.
9. Створіть шаблон і форму для його заповнення для документа за своїм розсудом.

## 10.2 Автоматизація робіт з таблицями у Microsoft Excel за допомогою VBA

VBA Microsoft Excel дозволяє автоматизувати роботи користувача в табличному процесорі: створення форм таблиць, вставку в них формул і виконання автоматичного перерахування, створення діаграм, створення і використання необхідних функцій, обмеження дій невмілого користувача й захист цілісності даних, виклик і виконання необхідних надбудов, роботу з файлами.

У VBA Microsoft Excel використовується більш 100 об'єктів, що знаходяться на різних рівнях ієрархії. На самому старшому рівні знаходиться сам додаток Excel. Ієрархія визначає зв'язок між об'єктами і показує шляхи доступу до них. Переглянути наявні типи об'єктів, їхні методи і властивості можна переключившись у редактор Visual Basic:

м. Сервіс → к. Макрос → п. Редактор Visual Basic → м. View → к. Object Browser.

На рисунку 10.5 наведений приклад ієрархії убудованих об'єктів VBA, а в додатку Б характеристика основних об'єктів.

Повне посилання на об'єкт складається з ряду імен вкладених послідовно один в одній об'єктів. Роздільниками імен об'єктів у цьому ряді є крапки, ряд починається з об'єкта **Application** і закінчується ім'ям самого об'єкта. Наприклад, повне посилання на чарунку **A1** робочого листа **Лист1** робочої книги з ім'ям **Архів** має вид:

```
Application.WorkBooks("Архів").Worksheets("Лист1").Range("A1")
```

Приводити щораз повне посилання на об'єкт зовсім не обов'язково. Звичайно досить обмежитися тільки неявним посиланням на об'єкт.

У неявному посиланні, на відміну від повного, об'єкти, що активні в даний момент, як правило, можна опускати. У розглянутому випадку, якщо посилання на чарунку **A1** дані в програмі, виконуваної у середовищі **Excel**, то посилання на об'єкт **Application** можуть бути опущені, тобто досить привести відносно посилання:

```
WorkBooks("Архів").Worksheets("Лист1").Range("A1")
```

Якщо робоча книга є активною, то посилання можна записати ще коротше:

```
Worksheets("Лист1").Range("A1")
```

Якщо робочий лист **Лист1** активний, то у відносному посиланні цілком достатньо обмежитися згадуванням тільки діапазону **A1:Range("A1")**.

Об'єкти в Excel також, як і в інших додатках володіють властивостями, керуються подіями і для них можна використовувати готові методи.

Властивість можна змінювати відразу у всіх об'єктів сімейства. У наведеному нижче прикладі за допомогою установки властивості **Visible** (видимість) значення **False** (неправда) усі робочі аркуші активної книги (сімейство об'єктів **Worksheets**) ховаються:

**Worksheets.Visible = False**

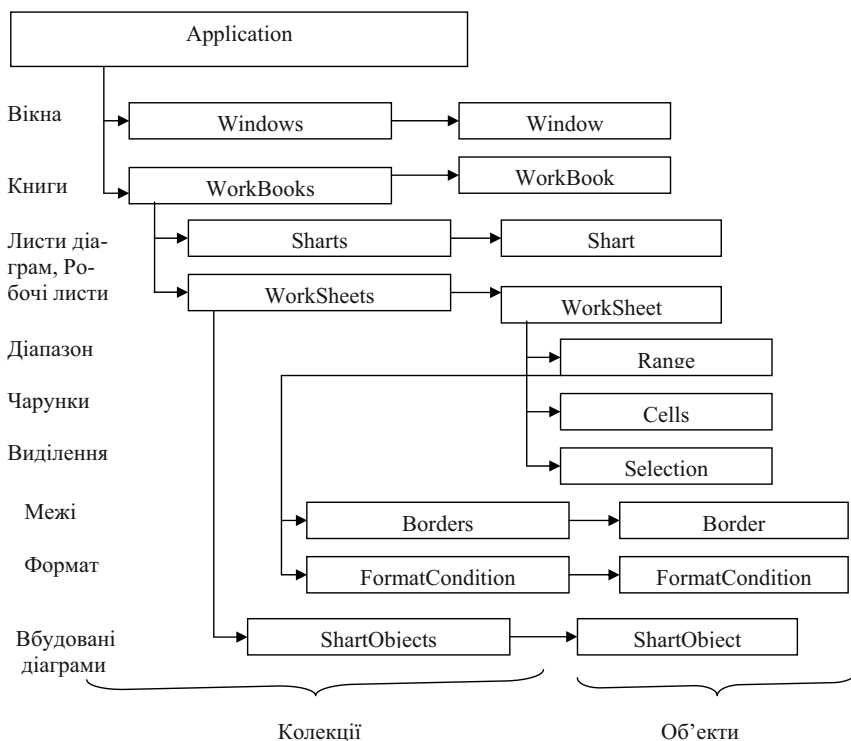


Рисунок 10.5 – Приклад ієрархії убудованих об'єктів VBA



Серед властивостей особливе місце займають властивості, що повертають об'єкт. Приведемо найбільше часто уживані подібні властивості.

**ActiveWindow** – повертає активне вікно Excel; **ActiveWorkbook** – повертає активну робочу книгу активного вікна Excel; **ActiveSheet** – повертає активний лист активної робочої книги; **ActiveDialog** – повертає активне діалогове вікно активного робочого листа; **ActiveChart** – повертає активну діаграму активного робочого листа; **ActiveCell** – повертає активну чарунку активного робочого листа.

Метод можна застосовувати до всіх об'єктів сімейства. У даному прикладі до сімейства **ChartObjects** (діаграми) робочого листа **Лист1** застосований метод **Delete** (видалити), що приводить до видалення всіх діаграм із робочого листа **Лист1**:

### **Worksheets("Лист1").ChartObjects.Delete**

Для подальшої роботи з VBA необхідно згадати методи адресації у Microsoft Excel (див. Розділ 4 даного посібника) і розібратися зі стилями посилань. У Excel використовується два стилі посилань – **A1** і **R1C1**. Одноразове їхнє використання в одній робочій книзі не можливо. У стилі **A1** у заголовках стовпців використовуються букви, а в стилі **R1C1** – цифри. Перехід до стилю **R1C1** здійснюється в такий спосіб:

м. **Сервис** → к. **Параметри** → вкладка **Общие** → виставити прапорець **Стиль ссылок R1C1** → **ОК**.

Повернутися до стилю **A1** – зняти прапорець **Стиль ссылок R1C1**.

Дотепер при роботі в Microsoft Excel використовувалися посилання стилю **A1**. У стилі **R1C1** для однозначної вказівки чарунки служать номера рядків і стовпців. Абсолютна адреса має вид **R2C1** – чарунка на перетинанні другого рядка і першого стовпця. Відносна адреса – показує положення чарунок щодо тої чарунки, у якій розташований курсор. При цьому номери у квадратних дужках показують зсув номерів рядків і стовпців щодо відповідних номерів виділеної чарунки. Зсув може бути позитивним і негативним (рисунок 10.6).



Наприклад:

```
Function Ткельвін (Тцельсій)  
    ТКельвін = ТЦельсій + 273  
End Function
```

Завдяки можливості використання в VBA україномовних імен, текст функції й призначення параметрів зрозумілі, крім того, це значно полегшує пошук потрібної функції за допомогою майстра функцій.

Створення користувальницької функції включає два етапи.

1). Спочатку створіть новий або відкрийте існуючий модуль, у якому ви хочете помістити вихідний текст програми Visual Basic, що визначає корис-тувальську функцію.

2). Потім уведіть ряд інструкцій Visual Basic для обчислення значення, що ви хочете повернути в робочий лист.

Приклад 10.1. – Для заповнення звіту про реалізацію безалкогольних напоїв (рисунок 10.7) необхідно створити користувальницьку функцію по заповненню граф залишків на кінець. Крім того, у звіті передбачена графа для розрахунку нарахувань реалізатору з урахуванням вироблення. (За кожен реалізовану пляшку реалізатор одержує 0,30 коп., якщо реалізовано більше 1000 пляшок, то премія 100 грн., більше 1500 пляшок – премія -200 грн.).

Виконання. У табличному процесорі Excel створіть таблицю звіту приведеної форми, у рядку **Разом:** проставлені формули суми.

Спочатку створимо функцію по обчисленню залишку на кінець. Дамо їй ім'я **ЗалКон**, аргументи назвемо – **ЗалНач**, **Прихід**, **Витрата**. Для цього необхідно виконати наступну послідовність дій.

1. М. **Сервіс** → к. **Макрос** → п. **Редактор Visual Basic**, у результаті чого відкриється вікно редактора VBA.

2. М. **Insert** (Вставка) → к. **Module** (Модуль) для створення листа модуля **Module1**.

3. Перейменуйте модуль – у вікні **Properties** властивості **Name** дайте ім'я **Функції** (рисунок 10.8).

	A	B	C	D	E	F	G	H	I	J	K
1	<b>Звіт про реалізацію безалкогольних напоїв</b>										
2	MBO		Сидоров	Дата		01.01.2006					Ставка
3				упаковка					грн		0,3
4	Назва товару	Залишок на початок_K	Надходження_K	Реалізовано_K	Залишок на кінець_K	Ціна	Залишок на початок_C	Надходження_C	Реалізовано_C	Залишок на кінець_C	Нараховано реалізатору
5	Фанта	2	30	9	23	2,4	4,8	72	21,6		
6	Кока-кола	5	25	5	25	3	15	75	15		
7	Миринда	4	22	4	22	3,2	12,8	70,4	12,8		
8	Байкал	1	17	7	11	2,5	2,5	42,5	17,5		
9	Монастирський квас	0	50	5	45	3,2	0	160	16		
10											
11											
12											
13	Разом:	12	144	30	126		36,1	419,9	82,9	0	0
14	За кожну реалізовану пляшку реалізатор одержує 0,30 коп., якщо реалізовано більше ніж 1000 пляшок, то премія										
15	100 грн., більше 1500 - премія 200 грн.										

Рисунок 10.7 – Форма документа до прикладу 10.1

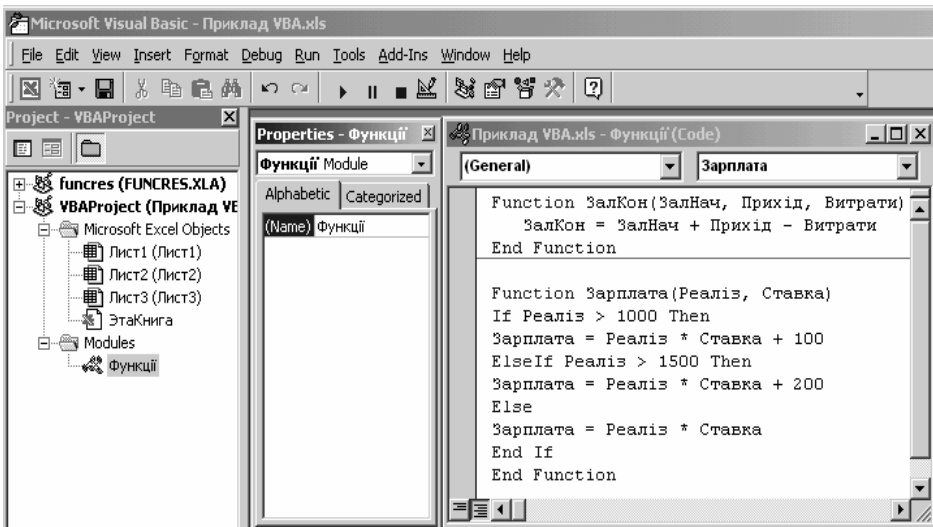


Рисунок 10.8 – Вікно VBA при створенні користувальницьких функцій

4. Наберіть у вікні редактора коду листа модуля наступну функцію:

```

Function ЗалКон(ЗалНач, Прихід, Витрати)
    ЗалКон = ЗалНач + Прихід - Витрати
End Function

```

5. Можна відразу зберегти модуль (інструментальна кнопка **Сохранить**), можна це зробити автоматично при збереженні робочої книги *Excel*.

6. М. **File** → к. **Close And Return To Microsoft Excel** для повернення в робочий лист із звітом.

7. Виділити чарунку для запису формули залишків (E5), потім викичте майстер функцій і виберіть категорію **Определенные пользователем**, у вікні функцій побачите функцію **ЗалКон**. Далі робота продовжується як із звичайною убудованою функцією. У результаті в чарунці повинна вийти наступна формула – **=ЗалКон(B5;C5;D5)**. Формулу можна скопіювати в усі необхідні чарунки.

Примітка – Зверніть увагу на роздільники параметрів у робочому листі – крапка з комою, а в кодї функції аргументи розділяються комами.

В чарунці **K10** необхідно записати функцію нарахувань реалізатору з урахуванням виплати преміальних. Оскільки реалізаторів у підприємстві може бути багато, спочатку створимо користувальницьку функцію за вище викладеною схемою (див. рисунок 10.8), а потім використовуємо її для запису формули.

Примітка – У *Excel* макроси й користувальницькі функції створюються й зберігаються в модулях і не залежать від конкретного листа, тому їх можна використовувати в різних робочих аркушах. Можна зібрати кілька макросів і користувальницьких функцій в одному модулі, як на рисунку 10.8, і розглядати його як бібліотеку. Усі користувальницькі функції будуть доступні й в інших активних книгах, якщо вони знаходяться у відкритій книзі.

Приведені приклади досить прості, але таким само способом можна створювати функції будь-якої складності.

## 10.2.2 Використання шаблонів і керуючих кнопок для створення документів

Шаблони автоматизують задачу створення робочих книг, дозволяють економити час і забезпечують відповідність всіх установок нового робочого листа побажанням користувача. Як і в Microsoft Word, шаблон застосовується всякий раз, коли запускається Microsoft Excel. Він служить як проєкт для нових робочих книг. Вихідний шаблон (Book.xls) створюється на основі установок за замовчуванням, що визначаються при інсталяції Excel. Саме завдяки цьому знову створювані робочі книги мають три порожні робочі листи. Якщо робочі книги повинні мати інші початкові установки, то можна створити користувальницькі шаблони, за допомогою яких і досягається необхідний ступінь автоматизації.

Досить часто на установку параметрів робочих книг витрачається набагато більше часу, чим на введення даних. Якщо ви відчуваєте себе непевно при визначенні параметрів робочих книг чи якщо вам приходилося копіювати робочий лист і видаляти старі дані перед уведенням нових, то ви серйозно задумаетесь про автоматизацію. Шаблони стають ще більш корисними, коли виникає необхідність у робочих аркушах, що мають гарний зовнішній вигляд і можливості по корекції. Це заощаджує значну кількість часу.

Шаблон створюється так само, як звичайна робоча книга, але його призначення полягає в тому, щоб виступати в ролі проєкту для інших робочих чи книг робочих аркушів. Шаблон включає всі характеристики робочої книги, що буде піддаватися дублюванню, наприклад, імена рядків, стовпців і заголовки. При цьому можна використовувати і кольорні користувальницькі схеми. Якщо ви застосовуєте шаблон, то вам не прийдеться знову створювати ті елементи, що є загальними для робочих книг на основі цього шаблону. За допомогою шаблонів можна відтворювати окремі робочі аркуші чи робочі книги, що включають набір робочих аркушів, отформатованих у потрібному вигляді.

Установки, що зберігаються Excel у шаблоні, визначають, як виглядають і якими властивостями володіють робочі книги і робочі аркуші, засновані на цьому шаблоні. У шаблоні зберігаються:

- інформація про параметри форматування й стилі;
- тексти заголовків і написів;
- макроси й формули робочого листа;

- графічні елементи (наприклад, логотипи);
- кольори й набори зразків для меж і чарунок, що офарблюються;
- користувальницькі панелі інструментів;
- гіперпосилання;
- зведення про захищені й сховані області.

Excel пропонує три типи шаблонів для робочих книг і аркушів:

- ✓ автошаблони для робочих книг з установками за замовчуванням;
- ✓ автошаблони для робочих аркушів з установками за замовчуванням;
- ✓ користувальницькі шаблони для робочих книг із заданими користувачем установками.

Створення шаблону починається так само, як створення звичайної робочої книги чи робочого листа. Тип шаблону, по суті, визначається його ім'ям, а також папкою, де він зберігається.

### Автошаблони

*Автошаблон* – це звичайний шаблон, що має особливий статус і використовується за замовчуванням для кожної створюваної робочої книги чи робочого листа, якщо не був обраний інший шаблон.

Для перетворення шаблону в автошаблон необхідно просто дати йому відповідне ім'я і зберегти його в спеціальній папці XlStart. Споконвічно в Excel мається один автошаблон для робочих книг (Book.xlt) і один для робочих аркушів (Sheet.xlt).

При запуску Excel шаблон Book.xlt використовується для створення робочої книги.

При виборі команди **Создать** із меню **Файл** відображається діалогове вікно **Создание документа** з обраним значком **Книга**, хоча насправді шаблону з таким ім'ям не існує і при виборі значка **Книга** використовується установки шаблону Book.xlt. Якщо обрана команда **Лист** із меню **Вставка**, то при створенні нового робочого листа застосовується шаблон **Sheet.xlt**. Excel додає новий лист в активну робочу книгу безпосередньо перед активним робочим листом, а потім робить новий лист активним.

Таким чином, при створенні користувальницьких шаблонів не можна присвоїти шаблону ім'я **Книга** (аналогічно шаблону листа – ім'я **Лист**), оскільки в *Excel* воно має спеціальне призначення.

## Користувальницькі шаблони

Шаблони, що не мають статусу автошаблону, називаються **користувальницькими**. Можна розробити скількох завгодно користувальницьких шаблонів, а потім використовувати їх у своїй роботі.

Приклад 10.2.- Створити користувальницький шаблон робочої книги зі звітами про реалізацію безалкогольних напоїв на п'ятьох аркушах. На кожному листі зробити шапку таблиці з жирною збільшеною назвою таблиці, підкреслити рядок з одиночними даними, виділити кольором і жирним шрифтом назви стовпців, ввести в чарунку **К3** ставку нарахування зарплати реалізатору (рисунок 10.9).

### Виконання.

1. Створіть нову робочу книгу → додайте в неї ще два листи (до п'яти аркушів) → об'єднаєте всі аркуші книги → на першому листі створіть шапку документа (рисунок 10.9) із відповідним форматкуванням → зніміть виділення з групи і перевірте результат роботи на аркушах книги.

Звіт про реалізацію безалкогольних напоїв										
МВО			Дата			ставка				
			Упакування			грн				
						0,3				
Найменування товару	Залишок на початок_K	Прихід_K	Реалізовано_K	Залишок на кінець_K	Ціна	Залишок на початок_C	Прихід_C	Реалізовано_C	Залишок на кінець_C	Нараховано реалізатору

Рисунок 10.9 – Шаблон звіту про реалізацію

2. М. **Файл** → к. **Сохранить как** → у діалоговому вікні **Сохранение документа** відкрити список, що випадає, **поля Тип файла** (вибрати елемент **Шаблон (\*.xls)**). При цьому в полі **Папка** автоматично відіб'ється папка **Шаблони**.

3. У поле **Имя файла** діалогового вікна **Сохранение документа** ввести ім'я файлу (**Звіт про реалізацію**) → кнопка **Сохранить**. Новий файл автоматично одержить розширення **.xls**.



Примітка 1. Перед збереженням робочої книги у виді шаблону не забудьте видалити всі зайві робочі аркуші. Якщо цього не зробити, то вони будуть присутні в кожному файлі, створеному на базі даного шаблону.

Примітка 2. Якщо у вас мається кілька робочих аркушів, що завжди використовуються разом (наприклад, звіти декількох реалізаторів чи бланк замовлення й рахунок-фактура...), розмістіть їх в одному шаблоні робочої книги.

Усі користувальницькі шаблони можна розмістити в окремій папці. Цю папку можна вказати в якості альтернативній початковій папці. При виборі команди **Создать** із меню **Файл** у діалоговому вікні **Создание документа** поряд із значками шаблонів, що знаходяться в стандартній папці автозавантаження, будуть відображатися значки користувальницьких шаблонів.

1. **М. Сервис** → к. **Параметри** → вкладка **Общие** → у поле **Каталог автозагрузки** введіть повний шлях папки з користувальницькими шаблонами.

2. Оскільки настроювання виконувалися на рівні додатка, необхідно вийти з табличного процесора, а потім завантажити його знову. Після цього шаблони з папки, зазначеної у полі **Каталог автозагрузки**, стануть доступними.

Можна створити в діалоговому вікні **Создание документа** додаткову вкладку, де будуть знаходитися всі користувальницькі шаблони. Для цього потрібно створити відповідну папку в каталозі **Windows\Application Data\Microsoft\Шаблони** і розмістити в ній свої шаблони.



Наступний рівень автоматизації роботи – створення керуючої кнопки на робочому листі, із яким можна зв'язати програму форматування, заповнення розрахунковими даними чи таблиці виконання яких-небудь інших операцій.


Приклад 10.3.- На листі із **Звітом про реалізацію безалкогольних напоїв** створити кнопку, клацання по якій повинно забезпечити розрахунок у таблиці порядкових і підсумкових показників, завершити оформлення звіту – виділити різним кольором кількісні і сумові дані, виді-

лити напівжирним збільшеним шрифтом підсумковий рядок. Таким чином користувачеві необхідно внести в таблицю тільки вхідні дані: своє прізвище, кількісні дані по залишках на початок, приходу й реалізації. Потім клацання по керуючій кнопці повинно завершити формування звіту.

#### Виконання.

1. Відкрити книгу із звітом про реалізацію на підставі створеного шаблону: м. **Файл** → к. **Создать** → виділити шаблон **Звіт про реалізацію** → **ОК**. Буде відкрита книга **Звіт про реалізацію** з уже намальованими таблицями на аркушах.

2. М. **Вид** → к. **Панели инструментов** → виставити панель **Visual Basic** → клацнути по кнопці **Элементы управления**  → клацнути по інструменту **Кнопка**  → намалювати в правому верхньому куті над таблицею прямокутну кнопку (кнопку можна малювати в будь-якій місці так, щоб вона не заважала працювати з таблицею).

3. Викликати контекстне меню намальованої кнопки → к. **Свойства** → у вікні **Properties**, що з'явилося, у властивості **Caption** замість **CommandButton1** написати **Розрахунок** → закрити вікно **Properties** (при цьому на панелі **Visual Basic** кнопка  **Конструктор** повинна залишитися утопленою).

4. Контекстне меню кнопки **Розрахунок** → к. **Исходный текст** → буде викликано вікно Microsoft Visual Basic, відкрито вікно **Code** для **CommandButton1** (кнопка **Розрахунок**) і події **Click** із відповідним записом начала процедури.

Примітка – Користувач повинний внести вихідні дані в таблицю самостійно, тобто заповнити ПІБ МВО (матеріально-відповідальної особи), графи 1-4 і 6 таблиці. Далі повинна працювати програма.

5. Оскільки кількість заповнюваних рядків точно не відомо, те введемо позначення: **k** – лічильник заповнених рядків; **i** – лічильник рядків у припущенні максимально можливого розміру предметної частини таблиці – 65 рядків.

Програма заповнення розрахункової частини й оформлення звіту з коментарями приведена нижче.

## Private Sub CommandButton1\_Click()

```
Range("F2").Select           'Виділення чарунки для дати
ActiveCell.Formula1C1 = Date 'Занесення поточної дати
k = 0                        'Лічильник фактично заповнених рядків
For i = 5 To 70              'Цикл по обробці рядків документа
                                'максимальна кількість рядків – 65
    If Cells(i, 1).Value <> 0 Then 'Перевірка наявності найменування товару
        k = k + 1                'Підрахунок рядків
                                'Заповнення чарунок по рядку формулами (графи 5,7-10)
        Cells(i, 5).Formula1C1 = "=ОстКон(RC[-3],RC[-2],RC[-1])"
        Cells(i, 7).FormulaR1C1 = "=RC[-5]*RC[-1]"
        Cells(i, 8).FormulaR1C1 = "=RC[-5]*RC[-2]"
        Cells(i, 9).FormulaR1C1 = "=RC[-5]*RC[-3]"
        Cells(i, 10).Formula1C1 = "=ОстКон(RC[-3],RC[-2],RC[-1])"
    End If
Next i                        'Підрахунок підсумкових сум по графах
Cells(k + 5, 1).Formula = "Разом:"
Cells(k + 5, 2).FormulaR1C1 = "=Sum(R5C2:R[-1]C2)"
Cells(k + 5, 3).FormulaR1C1 = "=Sum(R5C3:R[-1]C3)"
Cells(k + 5, 4).FormulaR1C1 = "=Sum(R5C4:R[-1]C4)"
Cells(k + 5, 5).FormulaR1C1 = "=Sum(R5C5:R[-1]C5)"
Cells(k + 5, 7).FormulaR1C1 = "=Sum(R5C7:R[-1]C7)"
Cells(k + 5, 8).FormulaR1C1 = "=Sum(R5C8:R[-1]C8)"
Cells(k + 5, 9).Formula1C1 = "=Sum(R5C9:R[-1]C9)"
Cells(k + 5, 10).Formula1C1 = "=Sum(R5C10:R[-1]C10)"
Cells(k + 5, 11).Formula1C1 = "=Зарплата(RC[-1],0.3)"
Cells(k + 5, 11).NumberFormat = "0.00"
                                'Виділення кольором кількісних і сумових даних
Range(Cells(5, 2), Cells(k + 5, 5)).Select           'Кількісні
Selection.Interior.ColorIndex = 35                    'колір бірюзовий
Range(Cells(5, 7), Cells(k + 5, 10)).Select          'Сумові
Selection.Interior.ColorIndex = 24                    'колір бузковий
                                'Виділення підсумкового рядка шрифтом
With Range(Cells(k + 5, 1), Cells(k + 5, 11)).Font
    .Name = "Times New Roman"
    .Size = 12
```

.FontStyle = Bold

End With

End Sub

5.Програму необхідно відлагодити. Для відлагодження програми необхідно виконати: м. **File** → **Close and return to Microsoft Excel** → заповнити декілька рядків таблиці → клацнути по кнопці **Розрахунок**. У результаті повинна вийти таблиця аналогічна рисунку 10.10.

6.Відлагоджену програму й у цілому книгу можна зберегти звичайним шляхом чи у вигляді шаблону, наприклад, з ім'ям **Звіт про реалізацію з кнопкою.xlt**, і використовувати останній як підставу для створення відповідних книг зі звітами.

У зазначеному прикладі користувач повинний нанести рамки са-мостійно, макросом вони не передбачені.

	A	B	C	D	E	F	G	H	I	J	K
1	<b>Звіт про реалізацію безалкогольних напоїв</b>										Розрахунок
2	MBO	Сидоров		Дата		01.01.2006					Ставка
3				упаковка					грн		0,3
4	Назва товару	Залишок на початок_K	Надходження_K	Реалізовано_K	Залишок на кінець_K	Ціна	Залишок на початок_C	Надходження_C	Реалізовано_C	Залишок на кінець_C	Нараховано реалізатору
5	Фанта	2	30	9	23	2,4	4,8	72	21,6	55,2	
6	Кока-кола	5	25	5	25	3	15	75	15	75	
7	Миринда	4	22	4	22	3,2	12,8	70,4	12,8	70,4	
8	Байкал	1	17	7	11	2,5	2,5	42,5	17,5	27,5	
9	Монастирський квас	0	50	5	45	3,2	0	160	16	144	
10											
11											
12											
13	Разом:	12	144	30	126		35,1	419,9	82,9	372,1	9
14	За кожну реалізовану пляшку реалізатор одержує 0,30 коп., якщо реалізовано більше ніж 1000 пляшок, то премія										
15	100 грн., більше 1500 - премія 200 грн.										

Рисунок 10.10 – Результат налагодження роботи програми кнопки **Розрахунок**

### 10.2.3 Форми як засіб взаємодії користувача і табличного процесора

Форми – це засіб розробки графічного інтерфейсу користувача в Excel. Форми являють собою вікна з набором елементів керування для спілкування з користувачем. Як і в Microsoft *Word*, форми, з одного боку, спрощують і конкретизують роботу користувача, а з іншого боку, дозволяють їй більш ретельно контролювати й захищати дані від помилок.

Форми і зв'язані з ними програмні коди, у т.ч. і користувальницькі функції, можна приєднувати до шаблонів і використовувати останні для створення окремих документів чи ряду взаємозалежних документів.

Побудова форм здійснюється також, як у всіх попередніх прикладах – необхідно увійти в Visual Basic (у нашому випадку в VBA Microsoft Excel), створити форму, задати властивості її керуючим елементам.

Приклад 10.4. – Додати в шаблон **Звіт про реалізацію** (приклад 10.2) електронну форму для заповнення таблиці вихідними даними. Заповнення звіту забезпечити через форму з контролем числової інформації і повного заповнення всіх полів. Розрахункова частина форми й остаточне заповнення звіту повинні виконуватися автоматично з використанням користувальницьких функцій. Користувачеві повинний видаватися запит про необхідність заповнення наступного листа. При позитивній відповіді на запит повинна заповнюватися наступна таблиця в такий само спосіб, як попередня.

#### Виконання.

1. Відкрити файл шаблону **Звіт про реалізацію**: м. **Файл** → к. **Відкрити** → у полі **Папки** вказати папку **Шаблони** (C:\Windows\Application Data\Microsoft\Шаблони ) → виділити **Звіт про реалізацію** → кнопка **Открыть**.

2. м. **Сервис** → к. **Макрос** → п. **Редактор Visual Basic** → м. **Insert** → к. **UserForm** → м. **View** → к. **ToolBox** → нанести на форму елементи керування необхідні для занесення вхідних даних (рисунок 10.11).

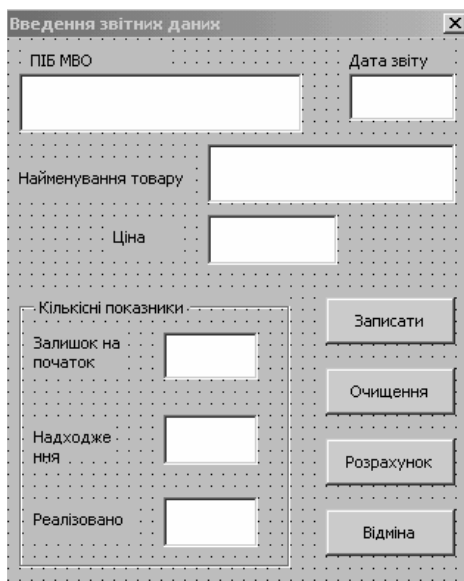


Рисунок 10.11 – Електронна форма до прикладу 10.4

Змінені властивості елементів форми проекту показані в таблиці 10.1.

Таблиця 10.1. – Властивості елементів електронної форми для прикладу 10.4

Елемент	Властивість	Значення	Примітка
1	2	3	4
UserForm1	Caption	Введення звітних даних	Назва форми
	BackColor	&H00FFC0C0&	Колір фону
Label1	Caption	ПІБ МВО	Призначення текстового поля під написом
	BackColor	&H00FFC0C0&	Колір фону
Text1	Name	FIO	Ім'я поля для введення ПІБ

## Продовження таблиці 10.1.

1	2	3	4
	Text		Порожньо
Label2	Caption	Дата звіту	Призначення тексто-вого поля під написом
	BackColor	&H00FFC0C0&	Колір фону
Text2	Name	Data	Ім'я поля для введення дати
	Text		Порожньо
Label3	Caption	Найменування товару	Призначення тексто-вого поля праворуч
	BackColor	&H00FFC0C0&	Колір фону
Text3	Name	NameTovar	Ім'я поля для введення назви товару
	Text		Порожньо
Label4	Caption	Ціна	Призначення тексто-вого поля праворуч
	BackColor	&H00FFC0C0&	Колір фону
Text4	Name	Cena	Ім'я поля для введення ціни товару
	Text		Порожньо
Frame1	Caption	Кількісні показники	Характеристика текс-тових полів у рамці
	BackColor	&H00FFC0C0&	Колір фону
Label5	Caption	Залишок на початок	Призначення тексто-вого поля праворуч
	BackColor	&H00FFC0C0&	Колір фону
Text5	Name	Ostatok	Ім'я поля для введення залишків по товару
	Text		Порожньо
Label6	Caption	Прихід	Призначення тексто-вого поля праворуч
	BackColor	&H00FFC0C0&	Колір фону
Text6	Name	Prihod	Ім'я поля для введення приходу по товару
	Text		Порожньо
Label7	Caption	Реалізовано	Призначення тексто-вого поля праворуч
	BackColor	&H00FFC0C0&	Колір фону
Text7	Name	Raxhod	Ім'я поля для введення реалізації по товару
	Text		Порожньо

Продовження таблиці 10.1.

CommandButton1	Name	Запис	Ім'я кнопки в програмі
	Caption	Записати	Назва кнопки у формі
CommandButton2	Name	Очищення	Ім'я кнопки в програмі
	Caption	Очищення	Назва кнопки у формі
CommandButton3	Name	Розрахунок	Ім'я кнопки в програмі
	Caption	Розрахунок	Назва кнопки у формі
CommandButton4	Name	Скасування	Ім'я кнопки в програмі
	Caption	Скасування	Назва кнопки у формі

Перейти до другого етапу проектування – написанню коду програми.

3. Поле дати за замовчуванням повинне заповнюватися поточною датою, що користувач може змінити при необхідності звичайним способом (виправити або стерти й написати потрібну). Заповнення поля необхідно виконати відразу з появою форми на екрані, тобто при ініціалізації форми – подія **Initialize**.

```
Private Sub UserForm_Initialize()  
    Data.Text = Format(Date, "dd.mm.yyyy")  
End Sub
```

4. Для текстових полів, у які заносяться числові дані, необхідно передбачити контроль уведення. Такі підпрограми багаторазово писалися в попередніх прикладах, тому далі буде просто приведений текст відповідних підпрограм.

Примітка – Код десяткової крапки на цифровій клавіатурі при роботі в російському алфавіті 44, а в англійському – 46.

```
Private Sub Cena_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)  
    Select Case KeyAscii  
        Case 0, 8, 44, 48 To 57  
        Case Else  
            KeyAscii = 0  
            MsgBox "Уведіть число на цифровій клавіатурі!"  
    End Select  
End Sub
```

```
Private Sub Octatok_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
```



```

Select Case KeyAscii
    Case 0, 8, 44, 48 To 57
    Case Else
        KeyAscii = 0
        MsgBox "Уведіть число на цифровій клавіатурі!"
    End Select
End Sub

```

```

Private Sub Pacxod_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Select Case KeyAscii
        Case 0, 8, 44, 48 To 57
        Case Else
            KeyAscii = 0
            MsgBox "Уведіть число на цифровій клавіатурі!"
        End Select
    End Sub

```

```

Private Sub Prixod_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Select Case KeyAscii
        Case 0, 8, 44, 48 To 57
        Case Else
            KeyAscii = 0
            MsgBox "Уведіть число на цифровій клавіатурі!"
        End Select
    End Sub

```

5. Написати коди програм для командних кнопок. Найпростіший і очевидний програмний код для кнопки **Очищення**. При клацанні по кнопці **Очищення** необхідно очищати текстові поля **NameTovar** (Найменування товару), **Cena** (Ціна), **Ostatok** (Залишок на початок), **Prixod** (Прихід), **Pacxod** (Реалізований).

```

Private Sub Очищення_Click()
    NameTovar = ""
    Cena.Text = ""
    Ostatok.Text = ""
    Prixod.Text = ""
    Pacxod.Text = ""
End Sub

```

Очищення поля **ФІО** (ПІБ МВО) повинно виконуватися при переході до заповнення наступного листа книги (див. нижче в процедурі **Otchet**).

6. Далі для запису програмного коду інших кнопок необхідно оголосити змінні: лічильник рядків із даними (**i**), змінну для збереження відповідей на запити (**n**) і логічну змінну для визначення бажання користувача розраховувати таблицю або скасувати роботу (**YesNo**). Перемінні будуть оголошені як глобальні, тому що вони будуть використовуватися в програмних кодах кнопок форми й модуля. Оголошення глобальних перемінних варто виконати в програмному модулі, що буде реалізувати всі роботи з автоматизації розрахунків.

Створіть новий програмний модуль, дайте йому ім'я **Звіт** і оголошите перемінні: м. **Insert** → к. **Module** → у вікні провідника проектів **Project** виділити модуль з назвою **Module1** → у вікні **Properties Module1**, що з'явилось, виділити властивість **Name** із значенням **Module1** → витерти **Module1** і написати **Звіт** → подвійним клацанням по імені модуля **Звіт** у вікні **Project** викликати вікно коду → для об'єкта **General** події **Declaration** записати оголошення перемінних.

**Global i, n As Integer**

**Global YesNo As Boolean**

7. Поверніться до електронної форми – вікно проектів **Project** → папка **Forms** → подвійне клацання по рядку **UserForm1**.

8. Програмний код кнопки **Записать** повинен забезпечувати запис вихідних даних у таблицю звіту в тому випадку, коли всі поля форми заповнені. Тому спочатку повинна перевірятися повнота заповнення форми, а потім виконуватися запис у таблицю.

**Private Sub Запис\_Click()**

**If FIO.Text = "" Then FIO.Text = InputBox("Уведіть ПІБ\_  
матеріально-відповідальної особи")**

**If NameTovar = "" Then NameTovar.Text = InputBox("Уведіть\_  
найменування товару")**

**If Cena.Text = "" Then Cena.Text = InputBox("Уведіть ціну  
товару")**

**If Octatok.Text = "" Then Octatok.Text = InputBox("Уведіть\_  
залишок товару")**

```

If PRIXOD.Text = "" Then PRIXOD.Text = InputBox("Уведіть_
    приход товару")
If PASXOD.Text = "" Then PASXOD.Text = InputBox("Уведіть_
    витрати товару")
If i = 5 Then 'Якщо перший запис, то записується
    Range("C2").Select 'в чарунку C2 ПІБ МБО,
    ActiveCell.Formula1C1 = FIO.Text
    Range("F2").Select 'в чарунку F2 дата звіту
    ActiveCell.Formula1C1 = Data.Text
End If
    'Заповнення предметної частини таблиці
    Cells(i, 1).FormulaR1C1 = NameTovar.Text
    Cells(i, 2).FormulaR1C1 = Octatok.Text
    Cells(i, 3).FormulaR1C1 = PRIXOD.Text
    Cells(i, 4).Formula1C1 = PASXOD.Text
    Cells(i, 6).Formula1C1 = Cena.Text
    i = i + 1 'Перехід до наступного рядка
End Sub

```

7. Для кнопок **Розрахунок** і **Скасування** програмні коди відрізняються тільки присвоєнням різного значення логічної змінної **YesNo**. Для кнопки **Розрахунок** – це **True** (Істина), а для кнопки **Скасування** – **False** (Неправда). Клацання по кожній з цих кнопок повинний сховати форму з екрана, щоб користувач бачив таблицю.

```

Private Sub Розрахунок_Click()
    YesNo = True
    UserForm1.Hide
End Sub

```

```

Private Sub Скасування_Click()
    YesNo = False
    UserForm1.Hide
End Sub

```

9. Далі у відповідності із значенням логічної змінної **YesNo** підпрограма **Otchet** модуля **Звіт** повинна або виконувати роботу з формування звіту в остаточному вигляді або очищати таблицю.

При заповненні користувачем таблиці за допомогою форми у змінній **i** було підраховане число фактично заповнених рядків (програ-

ма кнопки **Записати**). Тепер це число можна зберегти й використовувати. Для цього в процедурі **Otchet** оголосимо перемінну **k**.

Як зразок програми заповнення предметної й підсумкової частини таблиці буде використаний програмний код керуючої кнопки (приклад 10.3) з уточненим параметром кінця циклу. У попередньому прикладі це був передбачуваний максимум, а в даному прикладі змінна **k** буде містити фактично заповнену кількість рядків таблиці.

Для одержання зразка проектування виділення кольором окремих діапазонів чарунок, створення рамок можна використовувати макроси, які потім можна уточнювати. Такий прийом, при знанні елементів мови VBA, дуже зручний і прискорює процес написання програми. Для приклада запишемо макрос створення рамок у предметній частині таблиці й виділення підсумкової частини стовщеною рамкою.

**М. Сервіс** → **к. Макрос** → **п. Начать запись** → дати макросу ім'я **Рамки** → **ОК** → виконати звичайним способом роботи по створенню рамок у предметній частині таблиці → клацнути по кнопці **Остановить запись** або **м. Сервіс** → **к. Макрос** → **п. Остановить запись**.

Автоматично записаний код програми можна побачити через **м. Сервіс** → **к. Макрос** → **п. Макрос** → у діалоговому вікні виділити ім'я макросу **Рамки** → кнопка **Изменить**. У вікні коду користувач побачить наступну програму з коментарями авторів:

```
Sub Рамки()  
' Рамки Макрос  
' Макрос записаний 25.10.2005 (Войтюшенко )  
Range("A5:K16").Select ' Виділення діапазону чарунок предметної  
частини  
Selection.Borders(xlDiagonalDown).LineStyle = xlNone  
Selection.Borders(xlDiagonalUp).LineStyle = xlNone  
With Selection.Borders(xlEdgeLeft) ' Створення лівої границі діапазону  
чарунок  
.LineStyle = xlContinuous  
.Weight = xlThin  
.ColorIndex = xlAutomatic  
End With  
With Selection.Borders(xlEdgeTop) ' Створення верхньої границі діапазону  
чарунок  
.LineStyle = xlContinuous  
.Weight = xlThin  
.ColorIndex = xlAutomatic
```

```

End With
With Selection.Borders(xlEdgeBottom) ' Створення нижньої границі діапазону
чарунок
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight) ' Створення правої границі діапазону
чарунок
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical) ' Створення вертикальних ліній усередині
діапазону чарунок
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideHorizontal) ' Створення горизонтальних ліній
усередині
діапазону чарунок
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
Range("A16:K16").Select ' Виділення діапазону чарунок підсумково-
го рядка
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft) ' Створення лівої границі чарунок
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop) ' Створення верхньої границі чарунок
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom) ' Створення нижньої границі чарунок

```

```

.LineStyle = xlContinuous
.Weight = xlMedium
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)           ' Створення правої границі чару-
нок
.LineStyle = xlContinuous
.Weight = xlMedium
.ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)       ' Створення вертикальних ліній
.LineStyle = xlContinuous
.Weight = xlThin
.ColorIndex = xlAutomatic
End With
End Sub

```

Те, що було зроблено клацанням мишки по кнопці зовнішніх границь, вилилося в досить великий програмний код. Далеко не всі записані команди необхідні в нашій програмі, частина з них містить установки за замовчуванням і може бути опущена. Після внесених коректур макрос **Рамки** буде мати наступний вид:

```

' Рамки Макрос
' Макрос записаний 25.10.2005 (Войтюшенко )
Range("A5:K16").Select
Selection.Borders(xlEdgeLeft).Weight = xlThin
Selection.Borders(xlEdgeTop).Weight = xlThin
Selection.Borders(xlEdgeBottom).Weight = xlThin
Selection.Borders(xlEdgeRight).Weight = xlThin
Selection.Borders(xlInsideVertical).Weight = xlThin
Selection.Borders(xlInsideHorizontal).Weight = xlThin

Range("A16:K16").Select
Selection.Borders(xlEdgeLeft).Weight = xlMedium
Selection.Borders(xlEdgeTop).Weight = xlMedium
Selection.Borders(xlEdgeBottom).Weight = xlMedium
Selection.Borders(xlEdgeRight).Weight = xlMedium
Selection.Borders(xlInsideVertical).Weight = xlThin
End Sub

```

Отриманий у такий спосіб макрос можна взяти за зразок для запису відповідного фрагмента коду в процедурі **Otchet**.

Примітка – Поняття процедури й підпрограми синоніми.

Для створення підпрограми **Otchet** варто викликати вікно коду модуля **Звіт** – у вікні **Project VBA** подвійне клацання на назві **Звіт** у папці **Module**. У вікні коду, що з'явилося, нижче оголошення глобальних перемінних увести наступний текст програми:

**Sub Otchet()**

**Dim k As Integer** ' Перемінна для кількості заповнених рядків

**i = 5** ' Предметна частина документа починається з 5  
рядка

**YesNo = False** ' Початкова установка логічної змінної

**UserForm1.Show** ' Показати форму на екрані монітора

**k = i** ' Число фактично заповнених рядків

**If YesNo = True Then** ' Якщо клацнули по кнопці Розраху-  
нок

' Цикл по занесенню порядкових формул у документ

**For i = 5 To k**

' Заповнення чарунок по рядку формулами (графи 5,7-10)

**Cells(i, 5).Formula1C1 = "=Осткон(RC[-3],RC[-2],RC[-1])"**

**Cells(i, 7).FormulaR1C1 = "=RC[-5]\*RC[-1]"**

**Cells(i, 8).FormulaR1C1 = "=RC[-5]\*RC[-2]"**

**Cells(i, 9).FormulaR1C1 = "=RC[-5]\*RC[-3]"**

**Cells(i, 10).Formula1C1 = "=ОстКон(RC[-3],RC[-2],RC[-1])"**

**Next i**

' Підрахунок підсумкових сум по графах

**Cells(k, 1).Formula = "Разом:"**

**Cells(k, 2).FormulaR1C1 = "=Sum(R5C2:R[-1]C2)"**

**Cells(k, 3).FormulaR1C1 = "=Sum(R5C3:R[-1]C3)"**

**Cells(k, 4).FormulaR1C1 = "=Sum(R5C4:R[-1]C4)"**

**Cells(k, 5).FormulaR1C1 = "=Sum(R5C5:R[-1]C5)"**

**Cells(k, 7).FormulaR1C1 = "=Sum(R5C7:R[-1]C7)"**

**Cells(k, 8).FormulaR1C1 = "=Sum(R5C8:R[-1]C8)"**

**Cells(k, 9).Formula1C1 = "=Sum(R5C9:R[-1]C9)"**

**Cells(k, 10).Formula1C1 = "=Sum(R5C10:R[-1]C10)"**

**Cells(k, 11).Formula1C1 = "=Зарплата(RC[-1],0,3)"**

**Cells(k, 11).NumberFormat = "0.00"**

'Нанесення рамок у предметній частині

**Range(Cells(5, 1), Cells(k, 11)).Select**  
**Selection.Borders(xlEdgeLeft).Weight = xlThin**  
**Selection.Borders(xlEdgeTop).Weight = xlThin**  
**Selection.Borders(xlEdgeBottom).Weight = xlThin**  
**Selection.Borders(xlEdgeRight).Weight = xlThin**  
**Selection.Borders(xlInsideVertical).Weight = xlThin**  
**Selection.Borders(xlInsideHorizontal).Weight = xlThin**

'Виділення кольором кількісних і сумових даних

**Range(Cells(5, 2), Cells(k, 5)).Select** 'Кількісні  
**Selection.Interior.ColorIndex = 35** 'колір бірюзовий  
**Range(Cells(5, 7), Cells(k, 10)).Select** 'Сумові  
**Selection.Interior.ColorIndex = 24** 'колір бузковий

'Виділення підсумкового рядка рамкою й шрифтом

**Range(Cells(k, 1), Cells(k, 11)).Select**  
**Selection.Borders(xlEdgeLeft).Weight = xlMedium**  
**Selection.Borders(xlEdgeTop).Weight = xlMedium**  
**Selection.Borders(xlEdgeBottom).Weight = xlMedium**  
**Selection.Borders(xlEdgeRight).Weight = xlMedium**  
**Selection.Borders(xlInsideVertical).Weight = xlThin**  
**With Range(Cells(k, 1), Cells(k, 11)).Font**  
**.Name = "Times New Roman"**  
**.Size = 12**  
**.FontStyle = Bold**

**End With**

**End If** 'Кінець фрагмента для кнопки **Розрахунок**

'Очищення таблиці при клацанні по кнопці **Скасування**

**If YesNo = False Then**

**Range(Cells(5, 1), Cells(k, 11)).Select**  
**Selection.ClearContents**  
**Range("F2").Select** 'Виділення чарунки дати  
**Selection.ClearContents**  
**Range("C2").Select** 'Виділення чарунки ПІБ МВО  
**Selection.ClearContents**

**End If**

'Перехід до наступного листа



```

n = MsgBox("Будете заповнювати новий лист?", vbYesNo + _
vbQuestion, "Запит на продовження роботи")
If n = vbYes Then          ' Якщо «Так»
    UserForm1.FIO.Text = "" ' очищається ПІБ МВО
    n = InputBox("Уведіть номер заповнюваного листа", "Що_
заповнювати?")
    Worksheets(n).Activate ' Перехід на заданий лист
    Otchet                ' Поновлення роботи процедури Otchet
Else                      ' Якщо «Ні»
    UserForm1.Hide       ' сховати форму
End If
End Sub

```

10. Останнім кроком у написанні програмного коду є запис підпрограми автоматичного виклику процедури **Otchet** при створенні книги на основі шаблону **Звіт про реалізацію**. Дану підпрограму варто помістити в модуль **Звіт**.

```

Sub Auto_Open()
    Otchet
End Sub

```

11. Зберегти шаблон **Звіт про реалізацію** з усіма доповненнями.

12. Перевірити працездатність шаблону: м. **Файл** → к. **Создать** → вибрати шаблон **Звіт про реалізацію** → кнопка **Создать**.

У результаті повинна бути створена книга з п'ятьма аркушами, заповненими шапками звіту і викликана форма для введення вхідних даних користувачем. Заповніть не менш двох звітів і, при заповненні третього звіту, клацніть на кнопку **Скасування**. При правильному занесенні кодів програм у результаті дій користувача повинна вийти книга з двома заповненими звітами реалізаторів.

## Питання й завдання для самостійної роботи

1. Які роботи можна автоматизувати у Microsoft Excel?
2. Назвіть найбільш часто використовувані типи об'єктів і колекцій об'єктів у VBA Microsoft Excel?
3. Методи адресації й стилі посилань на чарунки в табличному процесорі?
4. Послідовність робіт при автоматизації формування документа?
5. Як створити шаблон у Excel? Що може входити до складу шаблону?
6. Що таке користувальницька функція й спосіб її створення?
7. Використання керуючих кнопок для автоматизації формування документів?
8. Призначення й можливості електронних форм у Microsoft Excel?
9. Як змусити програмний модуль книги виконуватися автоматично?
10. Додайте в шаблон Звіт про реалізацію лист із консолідованим звітом по всім реалізаторам.
11. Створіть шаблон і форму для його заповнення для документа за своїм розсудом.

## Література

1. Гарнаев А. Самовчитель VBA – Спб: БХВ – Санкт-Петербург, 1999 – 512с.
2. Додж М., Стинсон К. Ефективна робота з Microsoft Excel 2000 – Спб: Видавництво «Питер», 2000 – 1056с.
3. Камарда Б. Використання Microsoft Word 97: Пер.с англ. – К., М.: СПБ Издат. Будинок «Вільямс», 1998 – 900с.
4. Коттингхэм М. Excel 2000: керівництво розроблювача: Пер. с англ. – К.: Видавнича група ВНУ, 2000. – 704 с.
5. Новиков Ф., Яценко А. Microsoft Office 97 у цілому – Спб: ВНУ – Санкт-Петербург, 1998 – 624с.
6. Хэлворсон М., Янг М. Ефективна робота з Microsoft Office 2000 – Спб: Видавництво «Питер», 2000 – 1232с.

## СЛОВНИК ТЕРМІНІВ

- ASCII** – універсальний стандарт, що визначає кодування символів двійковим кодом (стор. 12).
- AUTOEXEC.BAT** – текстовий файл, який містить команди, що автоматично виконуються системою після початкового завантаження (MS-DOS або Windows).
- BIOS** (Basic Input-Output System) – базова система введення-виведення, «захита» у постійному запам'ятовуючому пристрої комп'ютера (ПЗП). Програми BIOS призначені для управління початковим завантаженням комп'ютера та взаємодією з деякими стандартними пристроями (стор. 19)
- CD-ROM** (Компакт-диск) – компакт-диск, зйомний носій інформації великого обсягу. Застосовується в мультимедіа, при поширенні програмних продуктів (стор 36).
- CMOS** – спеціальна енергонезалежна пам'ять, яка зберігає відомості про все устаткування і параметри ПК, поточну дату і час (стор 21).
- CONFIG.SYS** – файл конфігурації. За допомогою команд цього файла завантажуються драйвери, встановлюються параметри MS-DOS, оптимізується робота MS-DOS.
- dpi** (dots per inch) – одиниця виміру роздільної здатності принтера і роздільної здатності зображення. Виражає кількість крапок в одиниці довжини, у даному випадку в одному дюймі (стор 32).
- FAT 32** – формат таблиці розміщення файлів, що з'явився в останній редакції Windows 95 і Windows 98. Може використовуватися на жорстких дисках обсягом від 512 Мбайт до 4 Тбайт. Дозволяє використовувати дисковий простір більш ощадливо і ефективно, ніж формат FAT 16. Формат FAT 32 не допускає стискання диску стандартними засобами Windows 95 і Windows 98 (стор. 55).
- FTP** ( File Transfer Protocol ) – протокол передачі файлів, який регламентує процедуру пересилання файлів з одного комп'ютера на інший. Одночасно з цим прикладна програма, яка пересилає файли у відповідності до протоколу FTP (стор. 260 ).
- Gopher** – система дослідження ресурсів мережі INTERNET, яка керується за допомогою меню (стор. 269 ).
- OLE** (Object Linking and Embedding) – зв'язування та вбудовування об'єктів. Технологія Windows, яка дозволяє комбінувати при обробці документів засоби різних додатків. Основна ідея OLE – включення в документ одного додатку (OLE – клієнта) об'єкта, ство-

реного в іншому додатку (OLE – сервері), з можливістю редагування цього об'єкту засобами OLE – сервера (стор. 47).

**Plug and Play** – стандарт, відповідно до якого елемент обладнання повинен містити інформацію про модель і компанію-виготовлювача і допускати налагоджування програмними засобами. В результаті операційна система комп'ютера здатна розпізнати таке обладнання, завантажити необхідні драйвери і зробити безконфліктне налагоджування.

**TELNET** – протокол прикладного рівня, що надає користувачам терміналів інтерактивний доступ до обчислювальних ресурсів host-EOM в режимі поділу часу (стор. 269 ).

**Wave-файл** – звуковий файл у форматі, прийнятому в операційній системі Windows. Один з найбільш популярних звукових форматів. Wave-файли зазвичай мають розширення .WAV.

**WWW ( World-Wide Web )** – "всесвітня павутина" (мережа). Прикладна служба мережі INTERNET для пошуку документів та різних баз даних, заснована на гіпертекстовій логіці. Гіпертекстові документи містять ключеві слова, виконуючі роль покажчиків зв'язку з іншими документами (стор. 260 ).

**WYSIWYG (What You See Is What You Get)** – принцип відповідності екранного і друкованого представлення документа. Додатки, що дотримуються цього принципу, дозволяють переглядати документ на екрані у такому вигляді, у якому він буде отриманий при друкуванні на принтері (стор. 105).

**Абзац** – структурний елемент тексту. У процесорі *Word* – довільна послідовність символів між двома символами «Повернення каретки» (¶), а також між початком тексту і (¶) або між (¶) і кінцем тексту (стор. 104).

**Адаптери (контролери) зовнішніх пристроїв (карти розширення)** – апаратно – програмні пристрої для приєднання периферійних пристроїв (стор. 22).

**Алфавітно-цифрова інформація** – символи кодової таблиці, які можна ввести в ПК одним натисканням клавіші (букви, цифри, службові знаки) (стор. 12).

**Апаратне забезпечення** – сукупність технічних пристроїв і приладів. Необхідний елемент комп'ютерної системи.

**Арифметичний вираз (вираження)** – окремий випадок виразу. Операндами цього виразу можуть бути числа, змінні, функції. Знаки операцій: + (додавання), – (віднімання), \* (множення), / (ділення). Пі-

днесення до ступеню зазвичай позначається \*\* або ^ . Послідовність виконання операцій може бути змінена за допомогою дужок. Результат арифметичного вираження – число (стор. 316).

**Архівування файлів** – службова операція на комп'ютері, що дозволяє упакувати групу файлів в один файл-архів для економії місця на магнітних носіях (стор. 85).

**Архівація** – створення резервної копії даних. Резервну копію часто зберігають на гнучких дисках і магнітних стрічках стримера. Регулярне проведення архівації є необхідною процедурою для захисту інформації у випадку псування твердого диска або ушкодження даних (стор. 85).

**Атрибут** – деякий показник, що характеризує якийсь об'єкт і приймає для кожного екземпляра об'єкта конкретне числове, текстове або інше значення .

**Атрибути файлу** – записані на диску додаткові властивості файла. У число атрибутів входять дата і час останньої зміни файла, позначка про те, чи є файл схованим, деякі інші параметри.

**Багатозадачність** – властивість операційної системи, що полягає у можливості одночасного виконання декількох програмних додатків (стор. 47).

**База даних** – автоматизоване сховище даних. У ній за допомогою програмних комплексів – СУБД (**система управління базами даних**) накопичуються, зберігаються, обробляються і представляються в потрібному для користувача вигляді величезні обсяги інформації (стор. 14).

**База знань** – автоматизоване сховище всіх необхідних відомостей про специфіку і закони даної проблемної області, про способи розв'язання задач, які у ній виникають (стор. 14).

**Базова система введення-виведення** (BIOS, Basic Input-Output System) – див. BIOS

**Байт** – 8 бітів. Сукупність з восьми бітів, сприйнята комп'ютером як єдине ціле, наприклад, 10101110 (стор. 11).

**Біт** – найменша одиниця інформації, що виражає логічне значення «Так» або «Ні» і позначається двійковим числом 1 або 0. Значення біта – 0 або 1, які можна розглядати як альтернативу: «так-ні», «true-false», «включено-вимкнено» (стор. 11).

**Буфер обміну** (Clipboard) – частина оперативної пам'яті комп'ютера, яку Windows резервує на час своєї роботи для організації обміну

текстовими та графічними даними між документами та програмними додатками (стор.47).

**Видалення файла** – оголошення області, зайнятої файлом на диску, вільної для використання іншими файлами (стор. 59).

**Виноска** – структурний елемент тексту. Примітка до тексту, що знаходиться у нижній частині сторінки або наприкінці документа і забезпечується номером або іншою позначкою (стор. 113).

**Вираз (вираження)** – форма запису деякого висловлювання. Складається з операндів, об'єднаних між собою спеціальними знаками, які визначають семантику виразу. В інформатиці використовуються три типи виразів: арифметичні, умовні та логічні. Операнди цих виразів з'єднуються знаками арифметичних операцій, знаками відношень та знаками логічних операцій. При виконанні операцій над даними вираз отримує унікальне значення, яке є або числовим (для арифметичного виразу) або логічним (TRUE або FALSE, 0 або 1) значенням для умовного та логічного виразів. Послідовність виконання операцій може бути змінена за допомогою дужок (стор. 313).

**Відеоадаптер** – пристрій, що здійснює підготовку інформації для її відображення на екрані монітора. Іноді замість терміна «відеоадаптер» використовують спрощений варіант – відеокарта (стор. 22).

**Відеопідсистема** – груповий термін, що описує об'єднання відеоадаптера і монітора.

**Вікно** – прямокутний сегмент екрана, обмежений рамкою (стор. 49).

**Вікно додатка** – вікно, у якому виконується прикладна програма або працює програмний додаток (стор. 49).

**Вікно документа** – вікно, у якому розміщується документ – об'єкт обробки прикладної програми. Може бути частиною вікна додатка (стор. 51).

**Вінчестер** – жаргонне позначення твердого диска (стор. 29).

**Вірус** – спеціальна комп'ютерна програма, здатна «розмножуватися» і «заражати» інші програми. Вона може потрапити на комп'ютер без відома користувача через «заражену» дискету або разом з «зараженим» файлом. Багато вірусів здатні нанести серйозної шкоди: сповільнити роботу програм, знищити дані або вивести з ладу операційну систему (стор. 73).

**Вкладка** – якщо у діалоговому вікні передбачено встановлення великої кількості параметрів, їх розбивають на тематичні розділи

(підвікна). Кожне таке підвікно має вкладку (ярличок, заголовок, корінець), який видно на екрані навіть коли підвікно сховано. Для активізації схованого підвікна слід клацнути мишею на його вкладці (стор. 51).

**Властивість об'єкта** – сукупність конкретних ознак об'єкта в числовому або якісному вираженні. Наприклад, файл має такі характеристики як розмір, час і дату створення, автора, місце запису (стор. 14).

**Гарнітура** – набір шрифтів визначеного малюнка, але різних розмірів (кеглів,) і накреслень.

**Гіпертекст** – документи, що містять зв'язки з іншими документами (или зв'язки між сторінками); виділення зв'язки автоматично викликає відкриття іншого документа (стор. 267).

**Гнучкий диск** – зовнішній носій інформації у вигляді диска з магнітною поверхнею, укладеного в спеціальну оболонку. Дозволяє зберігати невеликі обсяги даних, зручний для їхнього перенесення з одного комп'ютера на інший (стор. 30).

**Головне меню** – основний елемент керування **Робочого столу** операційних систем Windows, призначений для швидкого доступу до додатків і службових функцій операційної системи. Викликається клацанням мишею на кнопці **Пуск** (стор. 47).

**Графічний редактор** – програма, призначена для створення та/або редагування графічних зображень. Окремо розглядають два класи графічних редакторів: растрові редактори і векторні редактори. Приклади: Paintbrush, CorelDRAW, Adobe Photoshop (стор. 41).

**Дані** – це інформація, яка представлена у формалізованому вигляді та призначена для обробки її технічними засобами, наприклад ЕОМ (стор. 10).

**Дефрагментація** – процес відновлення й оптимізації логічної структури диска з метою забезпечення збереження файлів у безперервних блоках дискового простору. Після дефрагментації прискорюється читання файлів і запуск програм (стор. 82).

**Джойстик** – пристрій керування комп'ютером, як правило, використовується в комп'ютерних іграх. Складається з опори, рухливої рукоятки і декількох кнопок. Джойстики бувають аналоговими і цифровими. Для комп'ютерів IBM PC використовують аналогові джойстики (стор. 31).

- Диск логічний** – частина пам'яті жорсткого диску, ідентифікується латинською буквою C:, D: і т.д. Цим терміном називають також умовні дисководи в локальній обчислювальній мережі.
- Диск** – магнітний носій інформації, представлена у вигляді файлів. Може бути зйомним (див. Гнучкий диск) і незйомним (див. Жорсткий диск) (стор. 29).
- Дискета** – див. Гнучкий диск.
- Дисковод CD-ROM** – пристрій для читання компакт -дисків (музичних, програмних) (стор. 36).
- Дисковод** – електронно-механічний пристрій для обслуговування магнітного диску (стор. 30).
- Дисплей** – Див. Монітор.
- Дистрибутивний комплект** – вихідний програмний пакет, що поставляється у вигляді набору компакт-дисків або гнучких дисків і призначений для установки додатка на комп'ютері.
- Діалогове вікно** – спеціальне вікно, яке використовується для налагоджування параметрів операційної системи або програмного додатка. Містить набір елементів керування, що можуть бути розміщені на декількох вкладишах. У діалоговому вікні користувач встановлює деякі параметри. У графічній технології – основний інструмент введення даних і керування роботою додатка (стор 51).
- Довге ім'я** – ім'я файла, що виходить за межі стандарту, прийнятого в системі MS-DOS. Довгими вважаються імена, що мають понад 8 символів у самому імені або понад 3 символи у розширенні. В імені або в розширенні дозволяється використовувати пробіли і спеціальні символи; комбінацію великих і малих літер. Довгі імена обробляються операційною системою Windows особливим чином (стор. 47).
- Документ** – об'єкт обробки прикладної програми. У широкому сенсі – об'єкт обробки Windows (стор. 47).
- Драйвер** (Від англійського «driver») – системна програма, яка управляє периферійним пристроєм. Приклади: драйвер клавіатури, драйвер миші, драйвер принтера (стор.28).
- Дюйм** – одиниця виміру довжини (1 дюйм = 2,54 см) (стор. 30).
- Екранна заставка** – спеціальна програма, що заміняє статичне зображення на екрані монітора динамічним. Використовується для зменшення навантаження на екран і приховання екранної інформації від сторонніх спостерігачів.



- Електронна таблиця** – структура даних, що забезпечує автоматичне виконання розрахунків по введеним у чарунки формулам (стор. 163).
- Елемент керування** – елемент діалогового вікна, призначений для визначення параметрів користувачем. У діалогових вікнах можуть бути присутніми як стандартні елементи керування (наприклад прапорці, списки, що розкриваються тощо), так і більш рідко уживані (палітри й інші спеціальні елементи) (стор. 51).
- ЕОМ** – електронний пристрій, призначений для автоматизації створення, введення, зберігання, обробки й виведення даних у необхідній для подальшої роботи формі (стор. 17).
- Жорсткий диск** – внутрішній пристрій комп'ютера для довгострокового збереження даних. Складається з блоку магнітних дисків у герметичному корпусі і блоку голівок для читання і запису інформації. Забезпечує можливість збереження великих обсягів інформації (декілька гігабайт) і високу швидкість її читання і запису (стор. 29).
- Значення виразу** – утворюється в результаті дій над операндами виразу відповідно до пріоритету операцій. Може бути числовим або логічним.
- Значок (позначка)** – наочне представлення об'єкта в операційних системах Windows. Див. Піктограма (стор. 47).
- Іконка** – див. Піктограма.
- Ім'я файла** – позначення файла, що складається з власного імені і розширення. Розширення може бути відсутнім, і тоді ім'я збігається з власним ім'ям. Розширення (якщо воно є) відокремлюється від власного імені крапкою. У Windows підтримуються довгі імена файлів. (Див. повне ім'я файла) (стор. 51).
- Інтерфейс користувача** – спосіб взаємодії людини з програмним додатком і програмного додатка з людиною (стор. 28).
- Інформатика** – фундаментальна природна наука, яка вивчає загальні властивості інформації, процеси, методи та засоби її обробки (збирання, збереження, перетворення, переміщення, видача), а також принципи функціонування цих засобів і методи керування ними (стор. 9).
- Інформаційна (комп'ютерна) технологія** – це сукупність методів, виробничих процесів і програмно-технічних засобів, об'єднаних в технологічний ланцюжок, що забезпечує збирання, обробку, збереження, поширення та відображення інформації з метою зниження трудомісткості процесів одержання і використання

інформаційного ресурсу, а також підвищення їх надійності та оперативності (стор. 14).

**Інформаційно-комунікаційна інфраструктура [Information and communication infrastructure]** – сукупність територіально розподілених державних і корпоративних інформаційних систем, ліній зв'язку, мереж та каналів передачі даних, засобів комутації та управління інформаційними потоками, а також організаційних структур, правових і нормативних механізмів, що забезпечують їх ефективне функціонування (стор. 15).

**Інформаційний ресурс** – це особливий вид ресурсу, який ґрунтується на ідеях та знаннях, накопичених в результаті науково-виробничої діяльності людей у визначеній предметній області, і представлений у зручному вигляді для накопичення та використання (документи і масиви документів в бібліотеках, архівах, фондах, банках даних, депозитаріях, музейних сховищах та ін.) (стор. 14).

**Інформаційні системи** – це сукупність даних, необхідних для ефективного керування економічними об'єктами, а також використувані інформаційні технології в сукупності з стосовними до них трудовими і фінансовими ресурсами організації (стор. 16).

**Каталог** – розділ файлової системи, що може містити файли і вкладені каталоги. У системах Windows замість терміна «каталог» вживається більш широкий термін папка. Інша назва – директорія (стор. 47).

**Кегль** – середній розмір (висота) шрифту (поліграфічний термін). Вимірюється в пунктах (стор. 104).

**Кількість інформації** – числова характеристика, що відбиває ту ступінь невизначеності (неповноту знань), яка зникає після одержання інформації (стор. 10).

**Клавіатура** – стандартний пристрій для уведення в ПК алфавітно-цифрової інформації і керуючих команд (стор. 27).

**Кластер** – мінімальна адресована одиниця дискового простору. Звичайно містить кілька фізичних секторів диска, запис у які здійснюється в рамках єдиної операції. Розмір кластера визначається використовуваною файловою системою і параметрами диска (стор. 74).

**Клацання** – натискання і відпускання кнопки миші. Основна операція в роботі з операційною системою Windows (стор. 28).

- Кодова таблиця символів** – внутрішнє представлення символів у машині. Кожен символ представлений десятковим числом (від 0 до 255), що займає один байт. Крім алфавітно-цифрових символів, кодова таблиця містить керуючі, псевдографічні й інші символи.
- Колонтитул** – структурний елемент документа. Знаходиться у верхній або нижній частині сторінки і містить деяку інформацію, що ідентифікує даний документ (номер сторінки, номер розділу, назву, дату, назву фірми тощо) (стор. 126).
- Командна кнопка** – елемент керування діалогового вікна. Командні кнопки використовуються для закриття діалогових вікон, відкриття нових діалогових вікон з додатковими параметрами, виконання підготовлених операцій (стор. 51).
- Компакт-диск** – один з видів оптичних дисків, призначений для зчитування даних лазерним променем. Компакт-диск може містити дані у цифровому або аналоговому вигляді. Компакт-диски, запис на які дозволяється здійснити тільки один раз, називають дисками CD-ROM. Компакт-диски, на яких можна здійснити перезапис інформації кілька разів, називають дисками CD-RW. Читання компакт-диска на комп'ютері здійснюється за допомогою дисководу CD-ROM (стор. 36).
- Комп'ютер** – пристрій для введення, обробки і відображення всілякої інформації (стор. 6).
- Комп'ютерна система** – сукупність апаратного і програмного забезпечення, що діє спільно.
- Контейнер** – об'єкт системи Windows, що може містити в собі інші об'єкти.
- Контекстна довідка** – повідомлення від операційної системи, яка містить підказку, пораду, рекомендацію, що стосується виконуваної в даний момент операції або обраного елемента керування.
- Контекстна чутливість** – властивість покажчика миші, що дозволяє йому змінювати форму залежно від того, на який об'єкт він наведений.
- Контекстне меню** – спеціальне меню, що містить набір команд, призначених для керування поточним об'єктом. В операційній системі Windows контекстне меню завжди відкривається клацанням правою кнопкою миші на об'єкті (стор. 44).
- Конфігурація** – блок інформації про налагодження, що використовуються конкретним користувачем: оформлення екрана, перелік драйверів та програм, які запускаються при завантаженні сис-

теми, елементи головного меню і робочого столу тощо (стор. 40).

**Конфігурація обладнання** – склад апаратного забезпечення комп'ютерної системи. (стор. 40).

**Конфігурація програмного забезпечення** – склад програмного забезпечення комп'ютерної системи.

**Копіювання** – це процедура, під час якої створюється новий об'єкт (наприклад, файл-копія або папка-копія), повністю тотожний оригіналу (стор. 58).

**Кореневий каталог** – каталог найвищого рівня на поточному диску. Всі інші каталоги вважаються вкладеними в нього у вигляді багаторівневої ієрархічної структури. Місце для кореневого каталога звичайно приділяється в строго визначеній області диска, тому число файлів і папок у кореневому каталозі, як правило, обмежене. Вкладені каталоги таких обмежень не мають (стор. 56).

**Кошик** – спеціальна папка системи Windows, призначена для тимчасового збереження вилучених об'єктів (стор. 47).

**Курсор** – миготлива позначка на екрані, що визначає ту позицію рядка, в яку буде введено символ при натисканні алфавітно-цифрової клавіші. Після введення символу курсор пересувається праворуч на наступну позицію рядка. У графічному режимі курсор має кілька модифікацій і може виконувати інші функції (стор. 50).

**Курсорні клавіші** – група клавіш зі стрілками на клавіатурі. Ці клавіші звичайно використовуються для переміщення курсора у відповідних програмних додатках (стор. 51).

**Кеш диску** – область оперативної пам'яті, яка спеціально приділяється під найчастіше використовувані в поточному сеансі роботи ділянки диска й під чергу на запис.

**Кеширування даних** – розміщення найбільш важливих даних у області пам'яті з більш швидким доступом (**кеш – пам'яті** – стор. 20).

**Лінійка** – знаходиться у верхній або лівій частині вікна документа (горизонтальна або вертикальна лінійка). Призначена для контролю й обробки документа (стор. 104).

**Логічний вираз** – окремий випадок виразу. Операндами цього виразу можуть бути умовні вирази або інші логічні вирази. Операнди з'єднуються знаками логічних операцій «і», «або»; операнд може мати префікс «не» (теж логічний знак). Послідо-

вність виконання операцій може бути змінена за допомогою дужок. Результат логічного виразу – логічне дане («істина» або «неправда», TRUE або FALSE, 1 або 0). Називається також булевим значенням .

**Логічний диск** – розділ фізичного жорсткого диска, який розглядається операційною системою як незалежний пристрій. Див. Диск логічний.

**Майстер** – спеціальна програма, що керує користувачем при виконанні визначеної операції. Подібне керівництво звичайно здійснюється через послідовність діалогових вікон, у кожному з яких користувач визначає необхідні параметри, після чого відбувається перехід до наступного вікна. Після збирання всіх необхідних відомостей операція виконується майстром автоматично (стор. 163).

**Материнська плата** – друкована плата, на якій здійснюється монтаж більшості компонентів комп'ютерної системи. Назва походить від англійської *motherboard*. Звичайно на материнській платі розташовуються гнізда для підключення центрального процесора, графічної плати, звукової плати, контролера жорсткого диску, оперативної пам'яті та додаткових периферійних устроїв (стор. 19).

**Машинне слово** – комбінація зв'язаних сусідніх розрядів, що обробляється спільно. Наприклад, 32-х розрядне слово. Розмір слова відповідає розрядності процесора (стор. 11).

**Меню** – елемент керування, що представляє собою набір пунктів (як правило, команд), з якого можна обрати один пункт. Пунктом меню може бути меню наступного рівня (вкладене меню) (стор. 47).

**Миша** – тип пристрою позиціонування, що забезпечує зручну роботу в графічних операційних системах. (стор. 28)

**Монітор** – пристрій, що здійснює відображення інформації на екрані. Практично всі персональні комп'ютери мають у своєму складі монітор або його аналог (наприклад, проекційну установку) (стор. 24).

**Накреслення** – текст у Windows може бути набраний будь-яким з чотирьох накреслень – звичайним шрифтом, напівжирним, курсивом і напівжирним курсивом (стор. 106).

**Об'єкт** – кожний предмет, явище, процес або стан, який сприймається як якийсь ціле, має відмітні ознаки та ім'я. Об'єкти у програмному забезпеченні – операційні системи, драйвери,

програми, файли; інші об'єкти – таблиці, малюнки, поля, чарунки і т.ін. (стор. 14)

**Об'єкт OLE (OLE-об'єкт)** – довільний елемент, створений засобами якого-небудь додатка (OLE-сервера), який можна помістити (вбудувати або зв'язати) у документ іншого додатка (OLE-клієнта) (стор. 305).

**Обчислювальна техніка** – сукупність устроїв, призначених для автоматичної або автоматизованої обробки даних (стор. 17).

**Операнд** – елемент виразу (літерал, змінна, функція, інший вираз) .

**Оперативна пам'ять (RAM)** – вид пам'яті комп'ютера, динамічно використовуваний в роботі програмних додатків. Призначається для короткочасного збереження даних під час їхньої обробки або введення/виведення. Являє собою сукупність спеціальних електронних чарунок, кожна з яких може зберігати конкретну комбінацію з нулів і одиниць – один байт. Ці чарунки нумеруються порядковими номерами, починаючи з нуля: 0,1,2,..., 32000, 32001,... Номер чарунки називається адресою того байта, який записано у ній у даний момент (стор. 19).

**Операційна система** – набір програм і драйверів, що забезпечують взаємодію програм з апаратним забезпеченням комп'ютера і базові можливості для введення команд користувачем. Розрізняють операційні системи, що використовують командний рядок для введення команд і запуску програм з використанням клавіатури, а також графічні операційні системи, у яких основним пристроєм керування є миша або інший пристрій позиціонування (стор. 40).

**Палітра** – елемент керування діалогового вікна; графічний різновид списку (стор. 70).

**Палітра, що розкривається** – елемент керування діалогового вікна; графічний різновид списку, що розкривається (стор. 318).

**Панель задач** – елемент керування Робочого столу. В операційних системах Windows забезпечує зручний доступ до усіх відкритих вікон папок і додатків (стор. 47).

**Панель інструментів** – елемент керування у вікнах папок і додатків. Являє собою панель з кнопками. Спрощує доступ до часто використовуваних команд. Один з елементів графічного інтерфейсу користувача, призначений для виконання інструментальних функцій і керування програмою. Частина засобів панелі інструментів дублює функції горизонтального меню (стор. 43).

- Панель керування** – спеціальна папка Windows, що містить значки системних засобів, використовуваних для налаштування комп'ютера й операційної системи.
- Папка** – загальний термін операційних систем Windows для позначення контейнера, що може містити інші об'єкти. Папки можуть бути вкладені одна в одну. Найчастіше папка являє собою каталог на диску, але існують і інші види папок (стор. 47).
- Пароль** – секретна комбінація символів, що служить для підтвердження права доступу до визначеного ресурсу, наприклад, до комп'ютера в мережі або до папки на диску.
- Переміщення** – операція зміни місця розташування об'єкта у файловій структурі (стор. 58).
- Перетягування** – прийом роботи з мишею, що полягає в її переміщенні при натиснутій лівій кнопці. Перетягування при натиснутій правій кнопці називається «спеціальним перетягуванням». У системі Windows використовується для переміщення і копіювання об'єктів, а також для створення ярликів (стор. 49).
- Периферійне обладнання** – зовнішні пристрої, підключені до комп'ютера (стор. 18).
- Персональний комп'ютер** – електронний прилад, призначений для автоматизації створення, збереження, обробки і передачі інформації (стор. 17).
- Підстановочний символ (шаблон)** – символ, який дозволяє визначити цілу групу символів, що могли б стояти в даному місці. Звичайно використовуються при визначенні імен файлів. Як правило, підстановочними символами є символи «\*» і «?» (стор. 60).
- Піксел (піксель)** – одиниця виміру роздільної здатності екрана. Відповідає окремій крапці на екрані, яка світиться. Кольором і яскравістю цієї крапки комп'ютер може керувати. Англійське слово «pixel» є скороченням фрази «Picture Element» (елемент зображення) (стор. 33).
- Піктограма** – графічне представлення на екрані Windows програмного елемента, програмної групи, вікна, перемикача, ярлика, інструмента і т.п. Називається також значком, іконкою, символом (стор. 47).
- ПК** – персональний комп'ютер (стор. 17).

- Повідомлення** – інформація, представлена у визначеній формі та призначена для передачі (стор. 10).
- Повне ім'я файлу** (специфікація файлу) – запис повного шляху до файлу, починаючи з устрою, на якому він (файл) розташований. Наприклад, A:\exc\less\letter.doc (стор. 56).
- Подвійне клацання** – два швидких клацання мишею. Один з основних прийомів роботи в системі Windows (стор. 10).
- Показчик миші** – позначка на екрані монітора, положення якої змінюється при використанні пристрою позиціонування. Може мати контекстну чутливість (стор. 28).
- Поле** – у документі Word – елемент тексту, що містить спеціальну інформацію (стор. 119).
- Поле зі списком** – елемент керування діалогового вікна; Відрізняється від текстового поля тим, що з ним зв'язаний список значень поля. Якщо користувача задовольняє один із запропонованих варіантів, можна клацнути на ньому мишею, і цей варіант розташується в полі. Якщо не задовольняє – можна клацнути на текстовому полі і ввести власний варіант (стор. 51).
- Постійна пам'ять (ПЗП або ROM)** – звичайно містить таку інформацію, яка не повинна мінятися в ході виконання мікропроцесором різних програм. Постійна пам'ять енергонезалежна, тобто може зберігати інформацію і при відключеному живленні (стор. 21).
- Початкове завантаження** – підготовка ПК до роботи при включенні електроживлення (стор. 47).
- Прапорець** – елемент керування діалогового вікна. Призначається для вибору або відмови від використання визначеного параметра (стор. 51).
- Принтер** – пристрій для виведення інформації на папір шляхом друкування (стор. 33).
- Пристрій позиціонування** – маніпулятор для керування у графічній операційній системі, наприклад миша.
- «Проводник»** – програмний додаток Windows, призначений для роботи з файловою структурою (стор. 52).
- Програмне забезпечення** – сукупність програм для комп'ютера. Більш строге визначення – це комплекс програм, документації та інструкцій по їх експлуатації, призначених для ефективного управління обчислювальним процесом, зменшення трудомісткості експлуатації комп'ютера, автоматизації процесу



підготовки, створення і виконання програм при різних режимах роботи машини, спрощення зв'язку користувача з ЕОМ (стор. 40).

**Протокол** – сукупність правил, згідно яким комп'ютери взаємодіють між собою при умові, що ці комп'ютери можуть бути різних виробників, але програми, що виконуються на цих комп'ютерах, досягають угоди про значення даних (стор. 39).

**Процесор** – основна мікросхема, що виконує у комп'ютері обробку інформації (стор. 19).

**Пункт** – одиниця виміру розміру (висоти) шрифту, рівного 1/72 дюйма (1 дюйм = 2,54 см) (стор. 104).

**Редагування** – внесення змін у документ (виправлення документа) (стор. 104).

**Редактор текстів** – програмний засіб для введення і модифікації текстових файлів або текстових документів (стор. 103).

**Режим MS-DOS** – режим роботи системи Windows, при якому всі ресурси комп'ютера віддані виконанню одного додатка MS-DOS. Застосовується також термін «режим змуляції MS-DOS».

**Резервна копія** – копія інформації, що зберігається на комп'ютері, виконана на окремому незалежному носії. Використовується для відновлення інформації у випадку ушкодження або втрати даних на жорсткому диску (стор. 85).

**Робочий лист** – елемент структури документа *Excel*, у якому розміщуються електронні таблиці. Входить до складу робочої книги (стор. 163).

**Робочий стіл** – об'єкт Windows, що виконує функції елемента оформлення. Служить для розміщення значків об'єктів, вікон папок і додатків, а також інших об'єктів і елементів керування операційною системою (стор. 47).

**Роздільна здатність екрана** – параметр, що визначає максимальний розмір зображення, який може бути відображений на екрані повністю. Вимірюється в пікселях. Можливі значення залежать від використовуваної карти відеоадаптера і монітора. У Windows стандартними вважаються роздільні здатності 640x480, 800x600, 1024x768, 1280x1024 і 1600x1200. Чим вище роздільна здатність екрана, тим меншу кількість кольорів може відображати відеоадаптер при фіксованих розмірах відеопам'яті (стор. 22).

- Роздільна здатність зображення** – параметр, що визначає кількість інформації, що утримується в одиниці площі графічного зображення. Визначає діапазон можливостей по обробці зображення. Вимірюється в одиницях dpi (стор. 32).
- Роздільна здатність принтера** – параметр, що визначає якість друкування друкуючого пристрою. Вимірюється кількістю крапок на одиницю довжини. Загальноприйнята одиниця виміру – dpi (стор. 34).
- Розкладка клавіатури** – система відповідності клавіш клавіатури символам національних алфавітів (стор. 26).
- Розширення імені** – комбінація символів після останньої крапки в імені файла. Розширення імені файла звичайно визначає тип файла (стор. 55).
- Сеанс MS-DOS** – спеціальне вікно системи Windows, що містить командний рядок MS-DOS. У цьому вікні можна давати команди MS-DOS і запускати додатки MS-DOS.
- Сектор** – мінімальний блок простору на диску, що дозволяє здійснювати читання або запис. Звичайно обсяг сектора дорівнює 512 байтам (стор. 29).
- Селекторна кнопка вибору** – елемент управління діалогового вікна; дозволяє вибрати один з альтернативних варіантів. Кнопки можуть бути об'єднані в групу. При виборі нової кнопки в групі, раніше включена кнопка вимикається (стор. 51).
- Сервер** – головний комп'ютер, на якому виконується програма, що дозволяє цьому комп'ютеру надавати послуги іншим комп'ютерам (стор. 240).
- Сигнал** – будь-який процес, який несе інформацію (стор. 10).
- Системне меню** – кнопка цього меню знаходиться в лівій частині рядка заголовка вікна додатка, вікна документа і діалогового вікна. Призначене для зміни варіанту представлення, розмірів і розташування вікна, а також для закриття вікна або переключення на інше вікно (стор. 50).
- Системний диск** – гнучкий диск, призначений для початкового завантаження комп'ютера при неможливості зробити завантаження зі стаціонарного жорсткого диска. Звичайно використовується при установці нової операційної системи, при пошуку несправностей або при необхідності переформатування жорсткого диска.
- Системная шина** – совокупность устройств и сигнальных линий, которые служат для связи микропроцессора с различными компонентами компьютера (стор. 21).

- Смуга гортання** – елемент керування вікна. Якщо об'єкти, що містяться у вікні, не можуть розташуватися в межах робочої області, то у вікні з'являються смуги гортання. Вони бувають вертикальні та горизонтальні, являють собою смуги з повзунком, відображаються на екрані тільки в тому випадку, коли вміст вікна не може уміститися в межах робочої області. Смуга прокручування знаходиться у нижній або правій частині вікна (горизонтальна або вертикальна смуга) і призначена для переміщення (скролінгу) по документу, якщо він не уміщається в своєму вікні (стор. 50).
- Список** – елемент керування діалогового вікно. Дозволяє обрати один параметр із наведеної множини. Якщо список не вміщується повністю у відведеному для нього місці, то з'являються смуги гортання. Список – це перелік можливих значень параметра. Один з рядків списку завжди є виділеним. На відміну від поля зі списком вводити дані до списку не можна (стор. 52).
- Список, що розкривається (випадаючий список)** – елемент керування діалогового вікна; для економії місця на екрані список може розкриватись при клацанні мишею на ньому. При виборі потрібного параметра список закривається (стор. 51).
- Стиль** – спосіб форматування структурного елемента текстового документа (послідовності символів або абзацу) (стор. 123).
- Стример** – пристрій для запису інформації на магнітну стрічку. Звичайно використовується для архівації даних. Сучасні стрімери дозволяють записати на один картридж із магнітною стрічкою весь жорсткий диск середніх розмірів (стор. 36).
- Схований файл** – файл, що за замовчуванням, не відображається при перегляді папки, у якій він розташований. Як правило, сховані файли використовують для зменшення імовірності їхнього ненавмисного знищення або зміни.
- Таблиця розміщення файлів** – таблиця, що встановлює відповідність між ім'ям файла і місцем на диску, де реально розташовані дані. Різні версії MS-DOS і Windows використовують різні версії таблиці розміщення файлів (FAT16, FAT32, NTFS) (стор. 55).
- Такт** – мінімальний інтервал часу, за який процесор може виконати якусь операцію. Багато інструкцій процесора вимагають для свого виконання декількох тактів (стор. 20).
- Тактова частота** – кількість тактів інструкцій, виконуваних процесором за одиницю часу (стор. 20).

- Текстове поле (поле введення)** – елемент керування діалогового вікна: Область введення текстової інформації з клавіатури. Для введення текстової інформації встановити мишу на позицію введення в текстовому полі та клацнути лівою кнопкою. В полі введення з'являється текстовий курсор (стор. 51).
- Текстовий редактор** – програма, призначена для створення, оформлення і форматування текстових документів. Використовується для підготовки документів, призначених для видачі на друк або, наприклад, відправлення по факсу (стор. 41).
- Текстовий файл** – файл, що містить сукупність рядків перемінної довжини (частіше від 0 до 255), причому кожен рядок – це сукупність довільних символів кодової таблиці, замкнений двома керуючими символами з кодами 13 («Повернення каретки») і 10 («Новий рядок») (стор. 363).
- Умовний вираз** – окремий випадок виразу і логічного виразу. Операндами цього виразу можуть бути арифметичні вирази і тексти. Операнди з'єднуються знаками відношень: «дорівнює», «не дорівнює», «більше», «менше», «більше або дорівнює», «менше або дорівнює». Результат умовного виразу – логічне значення («істина» або «неправда»). У деяких мовах у ролі умовного виразу може виступати арифметичний вираз, ненульовий результат якого вважається істиною (стор. 323).
- Упорядкування об'єктів** – процедура розміщення об'єктів у заданій послідовності. Як правило, те саме, що і сортування (стор. 117).
- Файл** – упорядочений набір даних, існуючих унікальне ім'я і зберігаються на одному з пристроїв (стор. 55).
- Файл ASCII** – окремий випадок текстового файла. Містить тільки символи першої половини кодової таблиці.
- Файлер** – діалогове вікно в середовищі Windows, призначене для вказівки системі адреси файла, який необхідно відкрити, зберегти або якось використовувати (стор. 72).
- Файлова система** – схема запису інформації, що міститься у файлах, на фізичний диск. Файлова система забезпечує можливість доступу до конкретного файла незалежно від того, у якому місці диска він реально записаний, і дозволяє знайти вільне місце при створенні нового файла. Системи MS-DOS і Windows 95 використовують файлову систему FAT 16. Опе-

раційна система Windows 98 використовує системи FAT 16 і FAT32, Windows 2000, XP – NTFS (стор. 55).

**Файлова структура** – структура розміщення папок (каталогів) і файлів на фізичному диску. Створюється й обслуговується засобами керування файловою системою (стор. 55).

**Фоновий малюнок** – елемент оформлення **Робочого столу** (стор. 48).

**Форматування диска** – розмітка диска і запис на нього службової інформації. Є необхідним етапом підготовки диска до запису на нього даних (стор. 30).

**Форматування документа** – оформлення документа. Операція форматування включає вибір розміру паперу для друкування документа, визначення ширини полів, встановлення шрифту, його розміру і накреслення, вибір методу вирівнювання рядків і методу оформлення абзацу, завдання величини відстані між рядками й абзацами, а також вибір методу обтікання текстом вбудованих графічних та інших об'єктів (стор. 104).

**Фрагментація** – небажана розбивка файлу на розташовані далеко один від одного блоки при запису на диск. Фрагментація призводить до зниження ефективності роботи диска. Причина фрагментації полягає у відсутності безперервної області достатнього розміру для розміщення файлу цілком (стор. 82).

**Функція** – змінна величина, значення якої залежить від значень інших величин (аргументів). Має ім'я й аргументи, що звичайно записуються через кому в дужках – слідом за ім'ям функції. Аргументом функції може бути будь-який операнд. Якщо функція використовується як операнд, вона завжди повинна приймати унікальне значення (арифметичне або логічне). У програмах бувають функції, що не приймають (не повертають) ніякого значення (стор. 119).

**Чарунка** – елемент структури електронних таблиць *Excel* або таблиць баз даних *Access*, утворений перетинанням рядка і стовпчика таблиці (стор. 118).

**Чипсет (Chipset)** – це набір мікросхем, які є сполучною ланкою між всіма компонентами материнської плати. Чипсет визначає всі основні характеристики материнської плати: тип підтримуваних процесорів, що підтримуються, тип пам'яті, стандарт шини для карт розширення і т. ін. (стор. 19).

**Шлях до файла** – позначення файлу з визначенням ланцюжка каталогів (папок), що ведуть до файла, і імені. Наприклад: `\port\excel\trf.bat` (стор. 56).

**Шлях пошуку** – послідовність вкладених каталогів (папок), однозначно визначає місце розташування файлу в ієрархічній структурі файлової системи. У MS-DOS і Windows каталоги і папки розділяються символом «\» (зворотний слеш) (стор. 60).

**Ярлик** – різновид значка в системі Windows. Ярлик, на відміну від значка, не представляє об'єкт, а тільки вказує на нього. Відрізняється від значка також наявністю стрілки в лівому нижньому куті. Для відкриття об'єкта можна двічі клацнути по його ярлику. Фізично являє собою файл-посилання із розширенням `.LNK` або `.PIF` (стор. 56).

**Ярличок** – елемент керування, призначений для переходу між робочими аркушами в робочій книзі *Excel* (стор.164 ).

## Загальноприйняті скорочення в Інтернеті

**Смайлики.** Це об'єднання самих звичайних символів, що разом нагадують знайомі усім речі. Якщо у Вас відразу не вийде розпізнати щось у цих закарлюках, спробуйте нахилити голову, краще вліво.

:-) або :0) або :)	посміхаюся
:-D	сміюся
:-o	дивуюся
:-( або :( або :o(	мені сумтно
:[ або :o[	мені дуже сумтно
:'-) або :;(	плачу
;-)	підморгну
#:-o	я в шоку
>>:-<<	я в люті
X=	схрестимо пальці
X-)	нічого не бачу
:-X	нічого нікому не скажу
:-P	я так запрацював, що язика висунув
%-)	я занадто довго сидів, уп'явшись у монітор
0:-)	янголятко
):-)	диявол
>:-)	ще один чортик
()]	пора по пиву
:-Q	покуримо
:-e	я розчарований
@}-->	трояндочка
12x@-->	дюжина трояндочок

Не будемо затверджувати, що в цей перелік увійшли всі смайлики. Хіба наша фантазія має границі? Можливо, саме Ви придумаєте щось новеньке.

Навіть у розмовній мові, особливо серед молоді, зараз зустрічаються такі комп'ютерні слівця, як софт (програмний продукт); хард софт (апаратне забезпечення); Вагя (продукт); рулез (rulez) (добре, круто, класно). А під час спілкування в Інтернеті, іноді ще вживають скорочення цілих фраз. Тому що фрази мають американське походження, то абрєвіатура, відповідно, англійська. От деякі з них:

2B or not 2B	<i>To Be Or Not To Be</i>	Бути чи не бути?
4ever	<i>Forever</i>	Назавжди
4u	<i>For You</i>	Для тебе
A/S/L	<i>Age/Sex/Location</i>	Вік/Піл/Місце знаходження
AFAIK	<i>As Far As I Know</i>	Наскільки мені відомо
AKA	<i>Also Known As</i>	Відомий також як
AMBW	<i>All My Best Wishes</i>	Мої найкращі побажання
AOTS	<i>All Of The Sudden</i>	Раптом
ASAP	<i>As Soon As Possible</i>	Якнайшвидше
B4	<i>Before</i>	Раніш
B4N	<i>Bye For Now</i>	Пока (у смислі прощання)
BBL	<i>Be Back Later</i>	Повернуся пізніше
BR	<i>Bathroom</i>	Ванна
BTW	<i>By The Way</i>	До речі
CIO	<i>Check It Out</i>	Перевір це
CUL8R	<i>See You Later</i>	Побачимося пізніше
CWYL	<i>Chat With You Later</i>	Побалакаємо пізніше
CYA	<i>See You</i>	Побачимося
DETI	<i>Don't Even Think It</i>	Навіть і не думай про це
DGT	<i>Don't Go There</i>	Туди не ходи
DLTM	<i>Don't Lie To Me</i>	Не обманюй мене
DRIB	<i>Don't Read If Busy</i>	Не читай, якщо зайнятий
EG	<i>Evil Grin</i>	Злісна усмішка
EOM	<i>End Of Message</i>	Кінець послання
ESO	<i>Equipment Smarter than Operator</i>	Устаткування розумніше оператора
F2F	<i>Face-to-Face</i>	Віч-на-віч
FOAF	<i>Friend Of A Friend</i>	Друг мого друга
FTASB	<i>Faster Than A Speeding Bullet</i>	Швидше кулі
FTL	<i>Faster Than Light</i>	Швидше світла
FYA	<i>For Your Amusement</i>	Для вашої розваги
FYI	<i>For Your Information</i>	До вашого відома
GDM8	<i>Good Day, Mate</i>	Добрий день, приятель
GL	<i>Good Luck</i>	Удачі!
GNBLFY	<i>Got Nothing But Love For You</i>	У мене для тебе нічого немає, крім любові
GR8	<i>Great</i>	Прекрасно, чудово!
GTG	<i>Got To Go</i>	Мені пора
GTGB	<i>Got To Go, Bye</i>	Мені пора, поки
GTH	<i>Go To Hell</i>	Йди до чорта
GTSY	<i>Glad To See Ya</i>	Рад вас бачити
HB	<i>Hurry Back</i>	Скоріше повертайтеся
HTH	<i>Hope This (That) Helps</i>	Сподіваюся, це допоможе



IAC	<i>In Any Case</i>	У будь-якому випадку
IDK	<i>I Don't Know</i>	Я не знаю
IDKY	<i>I Don't Know You</i>	Я не знаю вас
IDST	<i>I Didn't Say That</i>	Я цього не говорив
IDTS	<i>I Don't Think So</i>	Я так не думаю
IFAB	<i>I Found A Bug</i>	Я знайшов "глюк"
IIRC	<i>If I Remember Correctly</i>	Якщо я правильно пам'ятаю
IWM	<i>If It Were Me</i>	Якби я був на вашому місці
ILY	<i>I Love You</i>	Я тебе люблю
IMHO	<i>In My Humble Opinion</i>	По моїй скромній думці
IMO	<i>In My Opinion</i>	На мою думку
INMP	<i>It's Not My Problem</i>	Це не моя проблема
IOW	<i>In Other Words</i>	Іншими словами
IRL	<i>In Real Life</i>	У реальному житті
ISS	<i>I Said So</i>	Я так сказав
IYSS	<i>If You Say So</i>	Якщо ви так говорите
J/C	<i>Just Checking</i>	Просто перевіряю
J/K	<i>Just Kidding!</i>	Просто пустую
J/W	<i>Just Wondering</i>	Просто дивуюся
L8R	<i>Later</i>	Пізніше
LMK	<i>Let Me Know</i>	Дайте мені знати
LOL	<i>Laughing Out Loud</i>	Голосно сміятися
LTIC	<i>Laughing 'Til I Cry</i>	Сміятися до сліз
LYLAS	<i>Love You Like A Sister</i>	Люблю вас як сестру
MHOTY	<i>My Hat's Off To You</i>	Знімаю перед вами капелюх
Mor	<i>Male or Female?</i>	Чоловік чи жінка
MOTD	<i>Message Of The Day</i>	Послання дня
MTFBWY	<i>May The Force Be With You</i>	Так пребуде з Вами Сила
MYOB	<i>Mind Your Own Business</i>	Не лізьте не у свою справу
NAZ	<i>Name, Address, Zip</i>	Ім'я, адреса, поштовий код
NMP	<i>Not My Problem</i>	Це не мої проблеми
NOYB	<i>None Of Your Business</i>	Не ваша справа
NP	<i>No Problem</i>	Немає проблем
NRN	<i>No Reply Necessary</i>	Відповідь не потрібна
OIC	<i>Oh, I see</i>	Розумію
OT	<i>Off Topic</i>	Не в тему
OTOH	<i>On The Other Hand</i>	З іншої сторони
PLS	<i>Please</i>	Будь ласка
POV	<i>Point Of View</i>	Точка зору
RBTL	<i>Read Between The Lines</i>	Читати між рядків
RL	<i>Real Life</i>	Реальне життя
RMMM	<i>Read My Mail Man!</i>	Прочитай пошту від мене
RN	<i>Right Now!</i>	"Прямо шаз!"

RTK	<i>Return To Keyboard</i>	Повернися до клавіатури
SITD	<i>Still In The Dark</i>	Я як і раніше нічого не розумію
STYS	<i>Speak To You Soon</i>	Незабаром поговоримо
SWAK	<i>Sealed With A Kiss</i>	"Запечатано поцілунком"
SWDYT	<i>So What Do You Think?</i>	Ну і що ж ви (собі) думаєте?
THX, або TX, або THKS	<i>Thanks</i>	Дякую, спасибі
TIA	<i>Thanks In Advance</i>	Заздалегідь вдячний
TIAIL	<i>Think I Am In Love</i>	Думаю, що я закохався
TIC	<i>Tongue In Cheek</i>	Язик за зубами
TM	<i>Trust Me</i>	Довіряйте мені
TMI	<i>Too Much Information</i>	Занадто багато інформації
TTYL	<i>Talk To You Later</i>	Поговоримо пізніше
TY	<i>Thank You</i>	Спасибі
TYVM	<i>Thank You Very Much</i>	Велике спасибі
URYY4M	<i>You Are Too Wise For Me</i>	Ви занадто мудрі
WDYS	<i>What Did You Say?</i>	Що ви сказали?
WDYT	<i>What Do You Think?</i>	Що ви думаєте?
WOG	<i>Wise Old Guy</i>	Старий мудрець
WYP	<i>What's Your Problem?</i>	У чому ваша проблема?
WYRN	<i>What's Your Real Name?</i>	Як вас насправді звуть?
WYS	<i>Whatever You Say</i>	Що б ви ні сказали
YA	<i>Yet Another</i>	Ще один
YDKM	<i>You Don't Know Me</i>	Ти мене не знаєш
YNK	<i>You Never Know</i>	Ти ніколи не довідаєшся
YSYD	<i>Yeah, Sure You Do</i>	Так, звичайно

**Основні команди меню вікна проектування Visual Basic**

Пункт меню	Підпункт меню	Призначення
	2	3
<b>File</b>	New Project	Відкриття нового проекту
	Open Project	Відкриття вже існуючого проекту
	Add Project...	Додати проект
	Remove Project	Закрити вже існуючий проект
	Save Project	Зберегти проект під його ім'ям
	Save As Project...	Зберегти проект із запитом імені
	Save Form1	Зберегти екранну форму під її ім'ям
	Save Form1 As...	Зберегти екранну форму з запитом імені
	Save Selection	Зберегти виділене
	Save Change Script	Зберегти зміни
	Print...	Друкування тексту поточного проекту
	Print Setup...	Установка параметрів принтера
	Make Project.exe	Створення .exe файлу проекту
	Make Project Group...	Створити групу проектів
	Exit	Вихід із середовища Visual Basic
<b>Edit</b>	Can't Undo	Скасувати
	Can't Redo	Повернути
	Cut	Вирізувати
	Copy	Копіювати
	Paste	Уставити
	Remove	Переміщає виділене
	Delete	Видалити
	Delete Table from Database	Видаляє таблицю з бази даних
	Select All	Виділити все
	Select All Columns	Виділяє всі стовпчики
	Table: Set Primary Key	Визначає первинний ключ таблиці
	Table:Insert Column	Уставляє поле в таблицю
	Table:Delete Column	Видаляє поле з таблиці
	Find	Знайти
	Find Next	Шукає наступне значення, задане для пошуку

1	2	3
	Replace	Шукає значення, задане для пошуку, і заміняє його новим значенням
	Indent	Табуляція вперед
	Outdent	Зворотна табуляція
		Списки властивостей і методів, списки констант, параметри, контрольні крапки й ін.
<b>View</b>	Code	Відкрити вікно коду програми
	Object	Відкрити вікно екранної форми
	Object Browser	Відкрити вікно браузера об'єктів
	Immediate Window	Відкриває вікно для введення і безпосереднього виконання команд
	Locals Window	Відкриває вікно для перегляду значень перемінних.
	Watch Window	Відкриває вікно перегляду значень виражень.
	Project Explorer	Відкрити вікно провідника проекту
	Properties Window	Відкрити вікно властивостей
	Form Layout Window	Відкрити вікно макета форми
	Toolbox	Відкрити вікно елементів керування
	Data View Window	Відкриває вікно перегляду даних
	Color Palette	Відображає колірну палітру
	Toolbars	Відкриває меню, призначене для відображення <b>Панелі інструментів</b>
<b>Project</b>	Add Form	Додає у вікно проекту форму
	Add MDI Form	Додає MDI-форму
	Add Module	Додає програмний модуль
	Add Class Module	Додає користувальницький клас
	Add User Control	Додає користувальницький елемент керування
	Add Property Page	Додає стандартну форму налаштування властивостей
	Add WebClass	Додає Web-клас
	Add Data Report	Додає в проект звіт
	Add DHTML Page	Додає сторінку DHTML
	Add Data Environment	Додає середовище бази даних

1	2	3
	Add File	Додає файл
	Remove	Виключає файл із проекту
	References	Викликає вікно включення посилань проекту в бібліотеки
	Components	Відкриває діалогове вікно, яке дозволяє додати на панель елементів управління компоненти для їхнього подальшого включення в проект додатка
	Project Properties	Відкриває вікно властивостей проекту
<b>Format</b>	Align	Відкриває меню, що містить команди вирівнювання об'єктів
	Make Same Size	Відкриває меню, що містить команди, які керують розмірами об'єктів у формі
	Horizontal Spacing	Встановлює інтервал між об'єктами в горизонтальному напрямку
	Vertical Spacing	Встановлює інтервал між об'єктами у вертикальному напрямку
	Center in Form	Центрує об'єкти у формі
	Order	Направляє обраний об'єкт у нижній чи верхній шар форми
<b>Debug</b>	Step Into	Здійснює покрокове виконання процедури, включаючи також викликувані нею процедури
	Step Over	Здійснює покрокове виконання процедури без трасування викликуваних нею процедур
	Step Out	Виконання програми здійснюється до виходу з поточної процедури
	Add Watch	Відкриває вікно <b>Add Watch</b> для уведення виражень, за значеннями яких буде вестися спостереження
	Edit Watch	Відкриває діалогове вікно <b>Edit Watch</b> для редагування чи видалення виразів, за значеннями яких ведеться спостереження
	Quick Watch	Відкриває вікно <b>Quick Watch</b> для прогляду значення обраного вираження
	Toggle Breakpoint	Установлює крапку останова
	Clear All Breakpoints	Видаляє всі крапки останова
	Set Next Statement	Установлює наступний оператор

1	2	3
	Show Next Statement	Показує наступний оператор
<b>Run</b>	Start	Запускає додаток на виконання
	Start with Full Compile	Запускає додаток на виконання з повною компіляцією
	Break	Зупиняє виконання додатка
	End	Припиняє виконання додатка
	Restart	Перезапускає додаток
<b>Query</b>	Run	Виконує запит до бази даних
	Clear Results	Очищає результати запиту
	Verify SQL Syntax	Перевіряє синтаксис запиту
	Group By	Групує дані в запиті
	Change Type: Select	Створює запит для вибірки даних
	Change Type: Insert	Створює запит, призначений для додавання даних у таблицю, використовуючи значення інших таблиць
	Change Type: Insert Values	Створює запит, призначений для зміни даних
	Change Type: Update	Створює запит, призначений для зміни даних
	Change Type: Delete	Створює запит, призначений для видалення даних
	Change Type: Make Table	Створює нову таблицю на основі результатів вибірки
	Add to Output	Додає поля в список полів формованої вибірки
	Sort Ascending	Призначає сортування по зростанню
	Sort Descending	Призначає сортування по убаванню
	Remove Filter	Видаляє фільтр
	Select All Rows From <Table A >	Створює зовнішнє з'єднання, що вибирає всі записи з першої таблиці, навіть якщо вони відсутні в другій таблиці
	Select All Rows From <Table B >	Створює зовнішнє з'єднання, що вибирає всі записи з другої таблиці, навіть якщо вони відсутні в першій таблиці
<b>Diagram</b>	New Text Annotation	Додає пояснювальний текст

1	2	3
	Set Text Font	Установлює шрифт
	Add Related Tables	Додає зв'язану таблицю
	Show Relationship Labels	Показує найменування зв'язків
	Modify Custom View	Змінює вид відображення таблиць у діаграмі
	View Page Break	Показує розбивку на сторінки
	Recalculate Page Break	Перераховує розбивку на сторінки
	Arrange Selection	Вирівнює обрані таблиці
	Arrange Tables	Вирівнює всі таблиці
	Autosize Selected Tables	Автоматично встановлює розмір обраних таблиць
<b>Tools</b>	Add Procedure	Додати процедуру
	Procedure Attributes	Встановлює атрибути процедури
	Menu Editors	Викликає вікно редактора меню
	Options	Відкриває діалогове вікно для налаштування параметрів програми Visual Basic
	SourceSafe	Керує сховищем проектів
<b>Add-Ins</b>	Visual Data Manager	Запускає менеджера керування базою даних
	Add-In Manager	Відкриває діалогове вікно, яке дозволяє додати надбудову в Visual Basic 6
<b>Window</b>	Split	Розділяє вікно
	Tile Horizontally	Розміщає вікна горизонтально
	Tile Vertically	Розміщає вікна вертикально
	Arrange Icons	Змінює розміщення значків вікон
<b>Help</b>	Context	Викликає довідкову систему з активною вкладкою <b>Содержание</b>
	Index	Викликає довідкову систему з активною вкладкою <b>Индекс</b>
	Search	Викликає довідкову систему з активною вкладкою <b>Поиск</b>
	Technical Support	Викликає довідкову систему з інформацією про службу технічної підтримки Microsoft
	Microsoft on the Web	Викликає сторінку Microsoft у Internet
	About Visual Basic	Викликає довідкову інформацію про пакет Visual Basic 6

## Основні об'єкти VBA Microsoft Excel, їхні властивості, події і методи

Колекції і (об'єкти)	Характеристика	Події	Методи
1	2	3	4
<b>Application</b>	Має більш 120 властивостей і 40 методів для установки загальних параметрів Excel; дозволяє викликати більш 400 убудованих функцій робочого листа за допомогою конструкції: Application.Функція (Аргументи)	<b>NewWorkbook</b> – при створенні нової книги; <b>WorkbookActivate</b> – при активізації робочої книги; <b>WorkbookBeforeClose</b> – перед закриттям робочої книги; <b>WorkbookBeforePrint</b> – перед друком робочої книги; <b>WorkbookBeforeSave</b> – перед збереженням; <b>WorkbookNewSheet</b> – при додаванні нового листа в книгу; <b>WorkbookOpen</b> – при відкритті книги; <b>WorkbookDeactivate</b> – при втраті фокуса робочою книгою.	<b>Calculate</b> – примусове обчислення у всіх відкритих робочих книгах; <b>Run</b> – запускає на виконання програму або макрос; <b>Volatile</b> – викликає переобчислення функції користувача при зміні значення аргументів; <b>Wait</b> – тимчасове припинення роботи додатка; <b>OnTime</b> – призначає виконання процедури на визначений час; <b>Quit</b> закриває додаток.
<b>WorkBooks (Workbook)</b>	<b>ActiveSheet</b> – повертає активний лист книги; <b>ActiveSheet</b> – повертає активну діаграму; <b>Sheets</b> – повертає сімейство всіх аркушів книги; <b>Worksheets</b> – повертає сімейство всіх робочих аркушів книги; <b>Charts</b> –	<b>BeforeClose</b> – перед закриттям; <b>BeforePrint</b> – перед роздрукуванням; <b>BeforeSave</b> – перед збереженням; <b>Deactivate</b> – втрата фокуса робочою книгою; <b>NewSheet</b> – повертає новий лист; <b>Open</b> – відкриття існуючої робочої книги. <b>SheetActivate</b> – активізація робочого.	<b>Activate</b> – активізація робочої книги; <b>Add</b> – створює робочу книгу; <b>Close</b> – закриття робочої книги <b>Open</b> – відкриття робочої книги <b>Save (Save as)</b> – збереження робочої книги; <b>PrintPreview</b> – попередній перегляд; <b>PrintOut</b> – друк вмісту робочої книги; <b>SendMail</b> – відсилання



1	2	3	4
	<p>повертає сімейство всіх діаграм книги; <b>Saved</b> – були зміни (<b>True</b>) чи ні (<b>False</b>) – для збереження; <b>WriteReserved</b> – допускає (<b>True</b>) або ні (<b>False</b>) запис у документі.</p>	<p>листа; <b>SheetDeactivate</b> – робочий лист утрачає фокус</p>	<p>пошти.</p>
<b>Worksheets (Worksheet)</b>	<p><b>Name</b> – ім'я робочого листа; <b>Visible – True</b> – лист виводиться на екран, <b>False</b> – лист небачимо; <b>UsedRange</b> – повертає діапазон з даними; <b>ActiveCell</b> – повертає активну чарунку робочого листа; <b>Intersect</b> – повертає діапазон, що є перетинанням декількох діапазонів; <b>Union</b> – повертає діапазон, що є об'єднанням декількох діапазонів; <b>StandartHeight</b> – повертає стандартну висоту всіх рядків робочого листа.</p>	<p><b>BeforeClose</b> – перед закриттям; <b>BeforePrint</b> – перед роздрукуванням; <b>BeforeSave</b> – перед збереженням; <b>Deactivate</b> – утрата фокуса робочою книгою; <b>NewSheet</b> – при додаванні нового листа; <b>Open</b> – при відкритті робочої книги; <b>SheetActivate</b> – при активізації робочого листа; <b>SheetDeactivate</b> – коли робочий лист утрачає фокус.</p>	<p><b>Active</b> – активізує зазначений робочий лист; <b>Add</b> – створює новий робочий лист; <b>Delete</b> – видаляє робочий лист; <b>Protect</b> – захищає робочий лист від змін; <b>Unprotect</b> – знімає захист з листа; <b>Copy</b> – копіювання робочого листа; <b>Move</b> – переміщення робочого листа. <b>Evaluate</b> – перетворить вираження в об'єкт або значення. Використовується при уведенні формул і адрес чарунок з діалогових вікон.</p>
<b>Range</b>	<p><b>Value</b> – повертає значення з (або в) чарунки діапазону;</p>	<p><b>Activate</b> – активний діапазон;</p>	<p><b>Address</b> – повертає адресу чарунки; <b>AutoFit</b> – автоматично</p>

1	2	3	4
	<p><b>Name</b> – ім'я діапазону;  <b>Count</b> – повертає число об'єктів у наборі, наприклад рядків у діапазоні;  <b>CurrentRegion</b> – повертає число рядків поточного діапазону;  <b>EntireColumn</b> – повертає стовпець;  <b>EntireRow</b> – повертає рядок;  <b>Font</b> – повертає шрифт;  <b>Formula</b> – повертає формулу у форматі A1;</p>	<p><b>Address</b> – адреса діапазону;  <b>Borders</b> – границі;  <b>Cells</b> – чарунки;  <b>Column</b> – стовпчик;  <b>EntireColumn (EntireRow)</b> – поточний стовпчик (поточний рядок) і т.д.</p>	<p>набудовує ширину стовпця і висоту рядка;  <b>Clear (ClearComments, ClearContents, ClearFormats ClearNotes)</b> – очищення відповідного діапазону або типу даних;  <b>Copy</b> – копіювання;  <b>Cut</b> – вирізання;  <b>Delete</b> – видалення;  <b>Columns ( Rows)</b> – повертає кількість стовпців (рядків) з яких складається діапазон;  <b>Insert</b> – вставка чарунки або діапазону;  <b>Offset</b> – повертає діапазон, зміщений щодо даного;  <b>Select</b> – виділення діапазону;  <b>Consolidate</b> – застосовується</p>
	<p><b>Formula1C1</b> – повертає формулу у форматі R1C1;  <b>ShrinkToFit – True-</b> автоматичне розміщення тексту в чарунці,  <b>False</b> – протилежний випадок.</p>		<p>для об'єднання даних з декількох діапазонів в одну підсумкову таблицю:  <b>Об'єкт.Consolidate (Sources, _ Function, TopRow, LeftColumn, _ CreateLinks),</b>  де <b>Об'єкт</b> – діапазон для підсумкової таблиці;  <b>Sources</b> – масив посилань у R1C1-форматі на вихідні діапазони даних (повні імена);  <b>Function</b> – функція для консолідації;  <b>TopRow</b> – значення <b>True</b></p>

1	2	3	4
			<p>– категорії в заголовках стовпців, <b>False</b> – консолідація по розташуванню; <b>LeftColumn</b> – значення <b>True</b> – категорії в лівому стовпці, <b>False</b> – консолідація по розташуванню; <b>CreateLinks</b> – значення <b>True</b> задає зв'язок з вихідними таблицями.</p>
<p><b>Shats (Shat)</b></p>	<p><b>ChartArea</b> – область на листі, відведена під діаграму;  <b>ChartTitle</b> – заголовок діаграми;  <b>ChartType</b> – тип діаграми;</p>		<p><b>Gallery</b> – тип діаграми;  <b>Format</b> – формат для даного типу діаграми (до 10 форматів);  <b>Title</b> – назва діаграми;  <b>Add</b> – створює нову діаграму;  <b>Delete</b> – видаляє діаграму;  <b>CopyPicture</b> – копіює діаграму в буфер обміну як малюнок;  <b>Location</b> – пересуває діаграму на нове місце;  <b>Select</b> – вибирає діаграму.</p>

НАВЧАЛЬНЕ ВИДАННЯ

Наталія Миколаївна ВОЙТЮШЕНКО  
Алла Іванівна ОСТАПЕЦЬ

# ІНФОРМАТИКА І КОМП'ЮТЕРНА ТЕХНІКА

*2-ге видання*

Навчальний посібник

Керівник видавничих проектів – Б. А. Сладкевич  
Дизайн обкладинки – Б. В. Борисов  
Редактор – Н. П. Манойло  
Коректор – С. С. Савченко

Підписано до друку 01.10.2008. Формат 60х84 1/16.  
Друк офсетний. Гарнітура PetersburgС.  
Умовн. друк. арк. 35,5.  
Наклад 1000 прим.

Видавництво “Центр учбової літератури”  
вул. Електриків, 23  
м. Київ, 04176  
тел./факс 425-01-34, тел. 451-65-95, 425-04-47, 425-20-63  
8-800-501-68-00 (безкоштовно в межах України)  
e-mail: office@uabook.com  
сайт: WWW.CUL.COM.UA

Свідоцтво ДК №2458 від 30.03.2006