Прикарпатський національний університет імені Василя Стефаника
Факультет математики та інформатики
Кафедра інформатики

Завдання та методичні вказівки до лабораторних робіт «Структури даних»

Превисокова Н.В

Івано-Франківськ, 2010р

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних

ІС – інформаційна система

ЕОМ – електронно-обчислювальна машина

СУБД – система управління базами даних.

ВСТУП

У традиційних пакетних системах обробки інформації дані організовуються у вигляді незв’язних між собою локальних інформаційних файлів лінійної структури. Незважаючи на відносну простоту організації файлів системи мають ряд недоліків, а саме:

- надлишковість даних. Одні і ті ж дані повинні зберігатися у різних файлах. Це призводить до нераціонального використання пам’яті та збільшення часу роботи з файлами;
- неузгодженість даних. Оскільки одна і та ж інформація зберігається у різних файлах, то часто технологічно важко простежити за одночасними змінами в різних файлах;
- залежність структур даних від прикладних програм. У файлових системах логічна і фізична структура файлів повинні відповідати їх опису у прикладній програмі. Відповідно внесення змін у структури вимагає модифікації прикладних програм, які з цими даними працюють.

Тому досить актуальним постало питання створення запам’ятовуючих пристроїв прямого доступу, що стало передумовою для розв’язання проблеми надлишковості, неузгодженості та залежності даних від прикладних програм і сприяло розвитку концепції в організації інформаційного забезпечення, яка отримала назву автоматизованого банку даних. Основні переваги організації інформаційних засобів у вигляді автоматизованих банків:

- багаторазовість використання різними користувачами;
- зменшення витрат на створення і ведення баз даних;
- зменшення надлишковості даних;
- швидкість обробки непередбачуваних запитів до системи;
- простота і зручність внесення змін до даних, що досягається за рахунок єдиної системи організації та ведення баз даних;
- незалежність логічної і фізичної організації даних від прикладних

програм;

- забезпечення цілісності даних. Під цілісністю розуміють несуперечливість даних, що зберігаються в базі, для цього, як правило, використовують словник даних, який веде адміністрація.

1 СТРУКТУРИ ДАНИХ

1.1 Багаторівневе відображення та структури зберігання даних

За логікою управління та розміщення даних на носіях розрізняють логічну та фізичну структуру даних [1].

Під логічною розуміють структуру, яка враховує погляд користувача (управлінця) на дані, тобто таку, що будується на логіці управління, а не на його техніці. Як правило, вона багаторівнева і виділяти інформаційні одиниці можна як з нижчого, так і вищого рівня. Наприклад, для логічних структур даних у порядку агрегування виокремлюють такі елементи даних:

символ - реквізит - показник - масив - інформаційний потік - інформаційна база.

Символ – це елемент даних, який не має змісту. Це елементарний сигнал інформації (літера, цифра, знак).

Реквізит (атрибут) – це інформаційна сукупність найнижчого рангу ,яка не підлягає поділу на одиниці інформації. Доцільність виділення такої одиниці пояснюється тим, що потрібна однобічна характеристика конкретних об'єктів управління – або лише кількісна, або лише якісна. Тому реквізити бувають двох видів: реквізити-основи (реквізити-величини) та реквізити-ознаки. Реквізит-основа розкриває абсолютне або відносне значення реквізиту- ознаки. Реквізит-ознака відбиває якісні властивості сутності характеризує обставини, за яких відбувався той чи інший господарський процес.

Розрізняють форму і значення реквізитів. Форма реквізиту виявляється в його назві (наприклад, професія, сума), а значення реквізиту «професія» – це назва конкретної професії, наприклад, апаратник, ливарник, зварювальник тощо.

Реквізити-основи і реквізити-ознаки мають різне призначення в процесі обробки інформації: над реквізитами-основами виконуються арифметичні операції, над реквізитами-ознаками – логічні.

У спеціальній літературі трапляються синоніми реквізиту, а саме:

елемент, терм, атрибут, ознака тощо.

Сутність економічної інформації розкривається через економічний показник, що являє собою інформаційну сукупність з мінімальним складом реквізитів-ознак і реквізитів-основ, достатнім для створення елементарного документа (документорядка).

Показник є структурна одиниця, яка характеризує будь-який конкретний об'єкт управління з кількісного та якісного боку. Тому показник має назву, яка розкриває його форму, і значення, яке доповнює форму кількісно-якісними її характеристиками.

Набір взаємопов'язаних даних однієї форми (однієї назви) з усіма її значеннями являє собою масив даних. Прикладом масиву може бути сукупність даних про рух грошових коштів на підприємстві. Масив даних є основною інформаційною сукупністю, якою оперують у інформаційних процедурах.

Сукупність масивів даних, що стосуються однієї й тієї самої дільниці управлінської роботи, називають інформаційним потоком. За фізичного підходу відповідні структурні одиниці виділяються залежно від носія інформації та способу її фіксації. Наприклад, якщо за основну одиницю інформації взято паперовий документ, то можна виділити одиниці інформації вищого та нижчого рівня. Одиницями вищого рівня є стос документів, документаційне господарство об'єкта управління. Одиницями нижчого рівня є зона документа, рядок, графа, позиція.

При створенні інформаційних систем обробки даних великого значення набувають машинні структури даних. Це пов'язано з розміщенням масивів даних у пам'яті ЕОМ.

При внутрішній структуризації даних виділяють такі одиниці інформації:

символ - поле - агрегат даних - запис - файл - база даних

Поле – поєднання символів, яке приводить до створення мінімального семантичного елемента масиву (дата, цех, дільниця).

Агрегат даних – це пойменована сукупність двох і більше елементів нижчого рівня. Загалом до агрегату даних можуть належати як елементи, так і

інші агрегати даних. Прикладом агрегату даних можуть бути групи елементів, які утворюють адресу або дату народження.

Запис – пойменована сукупність полів, об'єднаних за змістовним принципом, яка є об'єктом та результатом одного кроку обробки даних. Прикладом запису можуть бути відомості про робітника.

Файл – поіменована сукупність записів для об'єктів одного типу. Як правило, записи, що входять до файла, мають однакову структуру. Прикладом файла можуть бути відомості про всіх робітників.

Агрегати даних і записи реалізуються на практиці організацією списків, черг, стеків, таблиць.

База даних – поіменована сукупність взаємопов'язаних файлів з мінімальною надмірністю, яка призначена для одночасного користування багатьма користувачами. Прикладом бази даних може бути гіпотетична база ЦЕХ, яка об'єднує файли РОБІТНИКИ, ВЕРСТАТИ, ВИРОБИ. Ці файли містять різноманітті відомості відповідно про робітників, обладнання цеху та виготовлювану в ньому продукцію, а між записами цих файлів існують зв'язки типу РОБІТНИК - ПРАЦЮЄ НА - ВЕРСТАТ; РОБІТНИК - ВИПУСКАЄ - ВИРІБ.

Для різних схем побудови програм можна виділити спільні рівні зображення даних. Одна зі схем, яка відповідає принципу розділення логічного і фізичного представлення даних, зображена на рис.1.

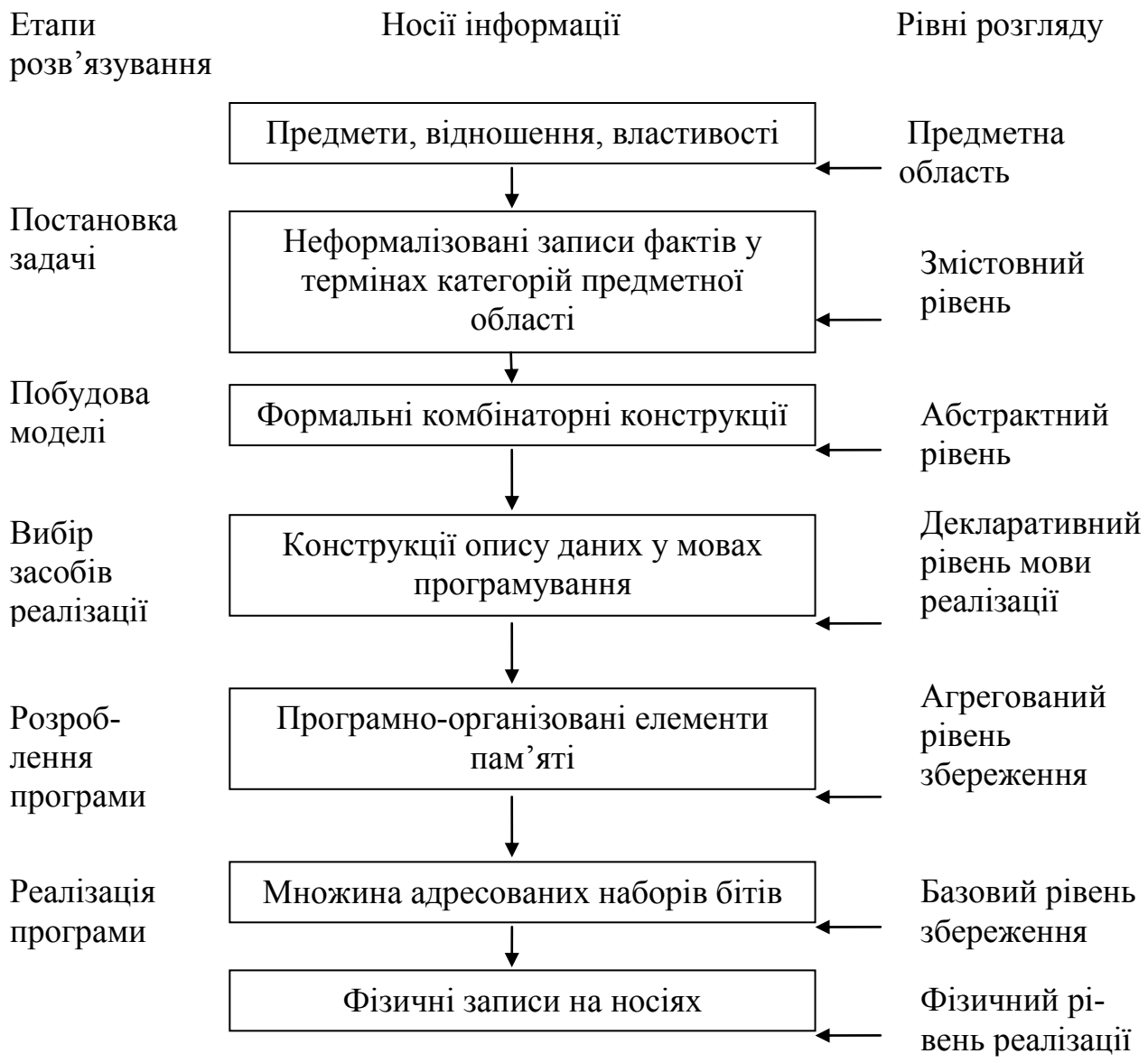

Рис. 1 Рівні відображення даних

Кожний з пропонованих рівнів розгляду даних відповідає певному етапу розв'язування задачі. Змістовний рівень відповідає постановці задачі, бо дає змогу описувати її неформалізовано у термінах предметної області. Побудова моделі передбачає вибір абстрактних структур даних, а реалізація програми використовує базовий рівень збереження даних.

У багаторівневому відображенні інформації про об'єкти реального світу на машинні носії виділено п'ять основних рівнів: змістовний, абстрактний, декларативний, агрегований, базовий. За своїм цільовим призначенням вони

стосуються проблемного (змістовний та абстрактний), процедурного (декларативний) або машинного (агрегований і базовий) середовища існування даних [2].

Залежно від застосовуваної системи програмування і організації даних у програмі, базова структура даних може бути використана по-різному. Дані можуть зберігатись лише в сусідніх одиницях пам'яті або одиницях, розкиданих по пам'яті і з'єднаних за допомогою спеціальних адрес зв’язку (вказівників). Інший вид доступу до розкиданих одиниць - обчислення адрес за деякою формулою. Відповідно до цих трьох типів відношень на сукупності одиниць пам'яті, які відводяться для збереження ДСД, розрізняють неперервне (послідовне), зв'язане (зчеплене) та розсіяне розміщення даних у пам'яті.

Який би не був спосіб розміщення даних, відповідна сукупність одиниць пам'яті організована за допомогою певних програмних засобів і моделює структуру пам'яті вищого рівня, ніж базова. Такі програмно-організовані "надбудови" над базовою структурою пам'яті називатимемо агрегатами пам'яті, а структури даних, які зберігаються у них віртуальних надбудовах - структурами зберігання даних (СЗД).

Залежно від того, який тип відношення між одиницями в агрегаті пам'яті, що містить СЗД, розрізняють неперервні, зв'язані та розсіяні СЗД (рис.2).

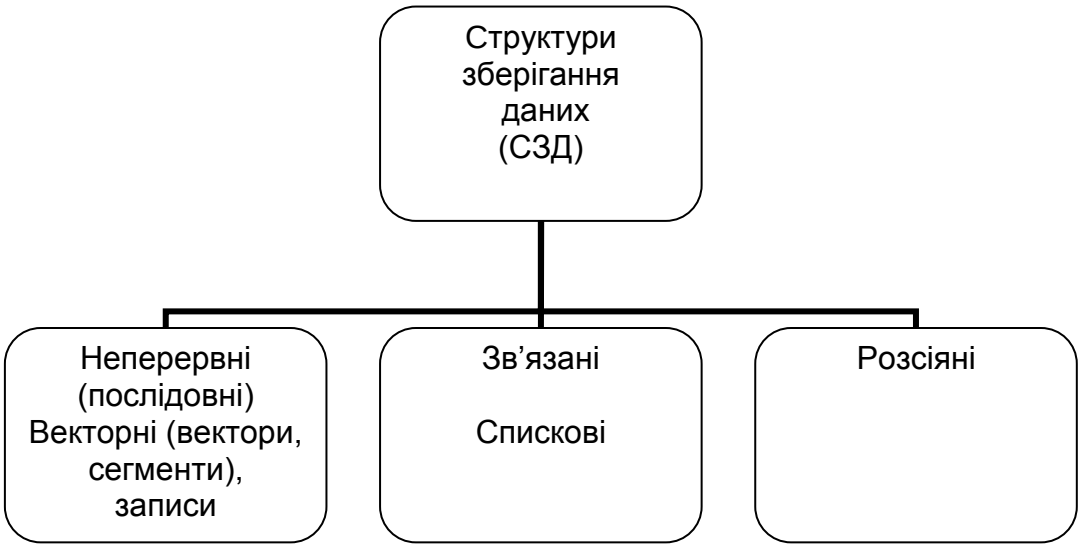

Рис. 2. Структури зберігання даних

Отже, довільні різновидності СЗД – це програмно-організовані

сукупності елементів доступу, що утворюють програмно-агрегований рівень збереження даних, який виникає на етапі розміщення ДСД у машинному середовищі віртуальних агрегатів пам'яті.

1.2 Квантування пам’яті

Для опису різновидів структур зберігання даних зручно користуватися поняттям квант пам'яті. Квант пам'яті – це сукупність сусідніх одиниць пам'яті, вміст яких інтерпретується як єдине ціле, наприклад, як двійковий код для зображення символу, число з фіксованою чи плаваючою крапкою, адреса, елемент даних разом з його дескриптором, який описує, наприклад, розмір елемента. Тоді внутрішню структуру агрегатів пам'яті можна буде описувати не в термінах одиниць пам'яті-базового рівня, а враховуючи об'єкти вищого рівня - кванти пам'яті, які містять елементи даних або вказівники зв'язку [3].

Адресація одиниць пам'яті визначає її структуру і доступ до неї. Використання адреси кванта пам'яті (адреси його першої одиниці) дає змогу реалізувати прямий доступ до пам'яті. Якщо задано адресу, то можна додавати до неї деяке фіксоване число, яке залежить від розміру кванта, тобто користуватися послідовним доступом до пам'яті. Довільний доступ не вводить жодних структурних зв'язків між квантами, а послідовний доступ накладає на пам'ять лінійну структуру. Якщо сусідні кванти об'єднано у сукупність, то адресу сукупності називають базовою, а адресу кванта щодо початку сукупності - зміщенням і тоді адреса кванта = база + зміщення.

Лабораторний практикум «Структури даних»

Лабораторна робота №1. Векторні величини

Тема. Основні структури даних. Робота з векторними величинами в системі MatLab.

Мета. Ознайомитися з видами структур даних. Навчитись створювати вектори в MatLab, виробити навики здійснення основних операцій над елементами вектора.

Теоретичні відомості

Розглянемо основні структури зберігання даних.

Послідовні структури зберігання. Об'єднання сусідніх квантів у єдиному блоці пам'яті без пропусків приводить до структури пам'яті послідовного типу, яку називають вектором, якщо всі кванти мають однаковий розмір, і неоднорідним вектором у протилежному випадку. Для неоднорідного вектора використовують термін "сегмент пам'яті".

Спискові структури. Об'єднання квантів, розташованих по базовій пам'яті довільно, за допомогою посилань від одного кванта до іншого приводить до агрегатів пам'яті зв'язного типу – спискових структур.

Спискова структура – це сукупність взаємозв'язаних ланок, кожна з яких складається з двох полів: поля даних та поля зв'язку. У полі зв'язку містяться адреси ланок, які визначають операцію прямування на структурі. Поле даних призначено для збереження елементів інформації або вказівника на інформацію. Зв'язані структури класифікують за кількістю елементів у полі зв'язку та за напрямом зв'язку. Іноді ще використовують фіксатори-вказівники на певні ланки структури, які дають змогу реалізувати прямий доступ до цих ланок. За кількістю елементів у полі зв'язку спискові структури можна поділити на одноелементні та багатоелементні.

Розсіяні структури. Третій тип структур зберігання даних – розсіяні структури – виникає у результаті використання спеціальних механізмів

розташування квантів у базовій пам'яті. Переважно це пов'язане з моделюванням асоціативної пам'яті, яка на відміну від традиційної, адресується не за фізичним розташуванням даних, а за вмістом ключового поля даних. Ключ, як послідовність бітів, інтерпретується не лише як частина елемента даних, а й як число, над яким виконують арифметичні дії, зазначені у функції розстановки (її називають також функцією хешування, розсіювання, адресною функцією). У результаті обчислень визначають відносну позицію елемента з заданим ключем у блоці пам'яті, відведеному під агрегат, забезпечуючи швидкий пошук даних із заданим вмістом.

Абстрактні структури. Абстрактні структури даних – це організовані певним чином сукупності даних. Кожну структуру визначають, правила і обмеження, за якими визначено зв'язки між різними частинами даних [2].

Важливим поняттям для визначення прямого доступу до окремих елементів даних є поняття індексації.

- Масив – це структура з прямим доступом; всі його компоненти можна довільно вибрати і вони однаково доступні.
- Таблиця – це набір елементів, з кожним з яких пов'язано ознаку, названу ключем, яка одночасно визначає позицію елемента і забезпечує прямий доступ до нього.
- Лінійна структура, доступ до елементів якої не змінює її, – це рядок.

Динамічні структури. Є три динамічні структури – стеки, черги та деки. Особливістю всіх динамічних структур є те, що елементи перебувають у них до першого звертання. Будь-яке звертання до такої структури змінює її вміст.

Розглянемо детальніше таку структуру даних, як вектор. Вектори застосовують для зберігання однорідних структур даних, які мають один дескриптор для всіх елементів. Дескриптор – це вектор, який містить додаткову інформацію, потрібну для декодування ланцюжка бітів. Такою може бути інформація про тип елементів вектора, довжину окремого елемента і т.д. Сегменти використовують для неоднорідних структур з різними дескрипторами для різних елементів. Структури зберігання даних, розташовані у векторах та

сегментах, називають векторними структурами даних і записами відповідно [7].

Вектор визначає:

1) база – адреса першого елемента;

2) розмір елемента – кількість одиниць пам'яті (квантів), яку займає один елемент даних;

3) довжина – кількість елементів даних.

Послідовний спосіб збереження структур даних компактніший стосовно розташування у пам’яті, проте явно не відображає складних відношень у структурі [5].

Для створення вектора в MatLab використовують наступний синтаксис:

```
>> e=[3 4 5 6]
```

Результат виведення матиме вигляд:

```
e =
  3    4    5    6
```

Вивести розмір вектора можна за допомогою функції length(<вектор>).

Над елементами вектора можна виконувати різні операції(множення чи ділення на число, застосування функцій до вектора), наприклад:

```
>> e*2
ans =
   6    8   10   12
```

Для того щоб заданий вектор-рядок представити у вигляді вектора-стовпця необхідно ввести його наступним чином:

```
>> e'
```

Результат виведення матиме вигляд:

```
ans =
   3
   4
   5
   6
```

Для звернення до елементів вектора використовують круглі дужки:

```
> e(3)
ans =
```

5

За допомогою функцій min(max) є можливість виведення найменшого (найбільшого) елемента вектора. Вивести середнє арифметичне елементів вектора можна, використовуючи функцію mean(<вектор>). MatLab дає можливість знаходити суму та добуток елементів вектора(функції sum(<вектор>) – сума, prod(<вектор>) – добуток). Сортування елементів вектора можна здійснити за допомогою функції sort(<вектор>), причому сортування здійснюватиметься по зростанню. Але коли виникає потреба сортування елементів за спаданням, то можна здійснити дзеркальне відображення посортованого за зростанням вектора за допомогою функції fliplr(<вектор>). В MatLab передбачено створення векторів, кожен елемент яких відрізняється від попереднього на однакову величину. Для цього використовують двокрапку, наприклад, x = 5:0.2:6.

Розглянемо ще деякі функції, що призначені для роботи з векторними величинами.

std – оцінка середньоквадратичного відхилення.

var – оцінка дисперсії.

cumsum – обчислення частинних сум елементів вектора (інтегрування методом прямокутників).

cumprod – обчислення частинних добутків елементів вектора.

diff – обчислення різниць між сусідніми елементами.

fft – одномірне пряме дискретне перетворення Фур’є.

fft2 – двомірне пряме дискретне перетворення Фур’є.

fftn – багатомірне пряме дискретне перетворення Фур’є.

ifft – одномірне зворотне дискретне перетворення Фур’є.

ifft2 – двомірне зворотне дискретне перетворення Фур’є.

ifftn – багатомірне зворотне дискретне перетворення Фур’є.

fftshift – перестановка половин вектора.

Ifftshift – перестановка, зворотня по відношенню до fftshift.

Для вектора в програмі MatLab при необхідності можна розраховувати та

виводити гістограми. Для цього використовують функцію hist(<вектор>), причому гістограма буде виводитись в окремому вікні [7].

Хід роботи

1. Завантажити MatLab. Створити вектор-рядок V: (3,2,1,4,5,5,7,8,9). Знайти та вивести його розмір.

2. Збільшити кожен елемент даного вектора в 2 рази, використовуючи операцію множення.

3. Застосувати функцію cos до вектора V.

4. Створити вектор-стовбець V1 з елементами попереднього вектора.

5. Знайти добуток V*V1.

6. Вивести перший та третій елемент вектора V .

7. Обчислити найбільший та найменший елемент вектора V та вивести його на екран.

8. Знайти середнє арифметичне елементів вектора V.

9. Знайти оцінку середньоквадратичного відхилення вектора V.

10. Посортувати елементи вектора V за зростанням та за спаданням.

11. Знайти суму та добуток елементів вектора V.

12. Обчислити оцінку дисперсії вектора V.

13. Обчислити та вивести частинні суми елементів вектора V.

14. Обчислити та вивести частинні добутки елементів вектора V.

15. Вивести різниці між сусідніми елементами.

16. Здійснити над даним вектором одномірне пряме дискретне перетворення Фур'є.

17. Виконати над вектором V двомірне та багатомірне пряме дискретне перетворення Фур'є.

18. Здійснити одно-, дво- та багатомірне зворотне дискретне перетворення Фур'є над вектором.

19. Створити вектор X=(1 2 3 4 5 6 7 8) та виконати в ньому перестановку половин вектора. Результат присвоїти змінній X1.

20. Здійснити зворотню перестановку половин вектора.

21. Розрахувати та вивести гістограму даного вектора.

22. Ввести вектор t = [0.2 0.3 0.5 0.8 1.3 1.7 2.5]. Обчислити значення функції y:

y = sin(t).^2./(1 + cos(t)) + exp(-t).*log(t)

23. Створити вектор P, елементи якого утворюють арифметичну прогресію чисел від 1 до 2, з кроком 0.1.

24. Створити вектор P1, елементи якого утворюють спадну арифметичну прогресію чисел від 6 до 5, з кроком -0.1.

25. Зберегти файл у своїй папці та закрити MatLab.

Запитання для самоконтролю:

- Які існують структури даних ?
- Що таке вектор?
- За допомогою якої функції можна вивести середнє арифметичне елементів вектора?
- Проаналізувати роботу функції cumsum.
- Як в MatLab вивести гістограму вектора?

Лабораторна робота №2. Масиви

Тема. Масиви.

Мета. Навчитись створювати масиви та здійснювати над ними основні операції в системі MatLab.

Теоретичні відомості

Масив – це, напевно, найвідоміша структура даних, оскільки у багатьох мовах це структура, яка існує в явному вигляді. Масив – це регулярна структура; всі його компоненти належать до одного типу, який називають базовим.

Масив – це структура з прямим доступом; всі його компоненти можна довільно вибрати і вони однаково доступні.

Масив – це набір елементів, з кожним з яких пов'язана послідовність цілих чисел, які називаються індексами. Індекси однозначно визначають місце елемента у масиві і забезпечують прямий доступ до нього. Кількість індексів визначає розмірність масиву. Кожний індекс має певно визначений діапазон зміни. Кількість елементів у масиві визначає його розмір. Розмір та розмірність масиву фіксовані для конкретного масиву.

Локалізувати елемент у масиві – означає зазначити місце його розташування, визначити його індекси [2].

Масив називають прямокутним , якщо кожен з його індексів змінюється у своєму діапазоні з постійним кроком від нижнього значення до верхнього, тобто належить лінійній ортогональній послідовності.

Двовимірний масив – матриця, де кожний елемент належить одночасно двом лінійним ортогональним послідовностям: і-му рядку та j-му стовпцю.

Прямокутні масиви, напевно, найчастіше використовують для реалізації обробки однотипних об'єктів.

Найпоширенішими операціями над масивами є:

1) пошук елемента за індексом (чи сукупністю індексів);

2) локалізація елемента у масиві;

3) зміна значення елемента у масиві (запис елемента);

4) сортування масиву;

5) копіювання масиву.

Відповідно до потреб під час використання масивів можуть бути введені й інші операції. Проте треба відзначити, що новий елемент не можна ні включити в масив, ні видалити з нього.

Операція локалізації елемента – визначення розміщення по відношенню до першого елемента масиву. Крім того, ця операція лежить в основі відображення масиву у послідовну структуру зберігання даних [2].

Всі об’єкти, якими оперує MatLab є масивами. Для того, щоб переконатися в цьому достатньо створити змінну і вивести її розмір [4].

```
>>x=3;
>>size(x)
ans
1
```

Отже , ми бачимо, що змінна трактується в MatLab як масив, розміром 1x1. Основні функції створення та опрацювання масивів в системі MatLab:

eye(m,n) – створення одиничної матриці ,розміром mxn;

ones(m,n) – повертає матрицю, елементами якої є одиниці;

zeros(m,n) – повертає матрицю ,елементами якої є нулі;

repmat(r,m,n) – створює матрицю, розміром mxn, заповнену числом r;

rand(m,n) – генерування випадкових чисел з рівномірним розподілом;

randn(m,n) – генерування випадкових чисел з нормальним розподілом;

ndims – виведення числа розмінностей масиву;

isempty – перевірка чи масив є порожнім;

isequal – перевірка масивів на рівність;

isnumeric – перевірка, чи масив є числовим;

islogical – перевірка, чи масив є логічним;

logical – перетворення числового масиву в логічний;

tril – виділення нижнього трикутного блоку матриці;

triu – виділення верхнього трикутного блоку матриці;

cat(dim,Al,A2,A3,A4,...) – здійснює конкатенацію масивів, в залежності від параметра dim(1 – вертикальна , 2 – горизонтальна);

Для створення масиву використовують квадратні дужки ,в яких через пропуски записують елементи рядків, рядки відокремлюють ‘;’

```
>> o=[1 2 3;1 2 3]
o =
   1   2   3
   1   2   3
```

Звернення до елементів масиву відбувається із вказанням індексу елементу:

```
>> o(1,2)
ans =
 2
```

sum(A,dim) – здійснює сумування елементів масиву А в залежності від параметру dim (dim =1 – по стовпцям, dim =2 – по рядкам).

prod(A,dim) – здійснює множення елементів масиву А в залежності від параметру dim (dim =1 – по стовпцям, dim =2 – по рядкам). Для здійснення поелементних операції необхідно перед знаком операції поставити крапку, наприклад: x./y

rot90(A,k) – здійснює поворот матриці А на 90x k градусів, де k – ціле число.

[i,j]=find(A=0) – виведення індексів нульових елементів вектора.

Для зменшення розміру масиву необхідно видалити рядки чи стовпці, шляхом присвоєння їм порожнього місця. Для прикладу видалимо з нашого масиву о другий рядок:

```
>> o(2,:)=[]
o =
 1   2   3
```

Для виділення прямокутного блоку елементів з масиву, необхідно вказати потрібні рядки і стовпці. Наприклад, створимо масив r і виділимо із нього блок, що охоплює 1-ий та 2-ий рядки і 1-ий та 2-ий стовпці:

```
>> r=[1 2 3;3 4 5]
```

```
r =
    1     2     3
    3     4     5
>> r(1:2,1:2)
ans =
  1     2
  3     4
```

Хід роботи

1. Завантажити MatLab.Створити змінну x=5 .За допомогою функції size переконатись в тому що змінна є масивом.

2. Створити одиничну матрицю A розміром 4x4.

3. Створити масив B, заповнений нулями, розміром 5x4.

4. Створити масив C , заповнений одиницями, розміром 5x5.

5. Створити матрицю, розміром 3x3, заповнену числом 7.

5. Здійснити вертикальну конкатенацію масивів A та B.

6. Виконати горизонтальну конкатенацію масивів B та C.

7. Згенерувати матрицю випадкових чисел, розміром 6x7, з рівномірним розподілом.

8. Згенерувати матрицю випадкових чисел, розміром 6x7, з нормальним розподілом.

9. Створити масив M:

```
M =
    1     2     3
    4     5     6
    7     8     9
```

10. За допомогою функції prod вивести результат перемноження всіх елементів масиву M по стовпцях.

11. Просумувати елементи матриці M по стовпцях.

12. Здійснити поворот матриці M на 90 градусів.

13. Виконати виведення числа розмірностей масиву M.

14. Створити масиви E=[5 6 7; 8 9 5] та E1=[5 6 7;8 9 5]. За допомогою

вбудованої в MatLab функції переконатися, що ці масиви є тотожніми.

15. Перевірити чи масив M є логічним.

16. Вивести верхній та нижній трикутні блоки матриці M.

17. Вивести масив N, який міститиме 1-ий та 2-ий рядки , 1-ий та 2-ий стовпці масиву M.

18. Вивести результат множення елементів масиву N на число 5.

19. Видалити другий рядок масиву M.

20. Перетворити масив M в логічний.

21. Вивести поелементний добуток двох масивів S та S1:

S(1,2,3,4), S1(4,5,7,3).

22. Створити масив P(-1 -2 3; 1 2 3; 4 5 -2). Вивести індекси елементів, які менші за нуль.

23. Вивести індекси елементів масиву P, які більші за нуль.

24. Розв’язати систему лінійних рівнянь методом Крамера, Гауса, матричним :

$$1{,}2\cdot x1+0{,}3\cdot x2-0{,}2\cdot x3=1{,}3;$$

$$0{,}5\cdot x1+2{,}1\cdot x2+1{,}3\cdot x3=3{,}9;$$

$$-0{,}9\cdot x1+0{,}7\cdot x2+5{,}6\cdot x3=5{,}4.$$

25. Зберегти файл у своїй папці та закрити MatLab.

Запитання для самоконтролю

- Яка структура даних називається масивом?
- Які ви знаєте найпоширеніші операції над масивами?
- Для чого використовують функцію rot90(A,k)?
- Яким чином можна вивести по елементний добуток масивів?
- Як виділити верхній трикутний блок матриці?

Лабораторна робота №3. Статистична обробка даних

Тема. Статистична обробка даних у масиві.

Мета. Вивчити основні операції опрацювання даних в масиві.

Теоретичні відомості

Статистична обробка даних масиву включає знаходження мінімального (максимального) елементу масиву, середнього значення, коефіцієнтів кореляції та коваріації тощо [4].

«Магічним квадратом» називається матриця, суми елементів якої по стовпцям і рядкам рівні. Функція magic(n) призначена для формування такого квадрату і включена в систему MatLab(n-розмір «магічного квадрату»).

Опишемо основні функції, призначені для статистичної обробки даних масиву:

max(A) – повертає вектор максимальних елементів масиву A по стовпцям;

min(A) – повертає вектор мінімальних елементів масиву A по стовпцям;

mean(A,dim) – повертає вектор середніх значень масиву A (dim=1 –по стовпцям; dim=2-по рядкам);

max(A,B) – повертає масив того ж розміру, що й A і B, кожен елемент якого є максимальний із відповідних елементів в цих масивах;

min(A,B) – повертає масив того ж розміру, що й A і B, кожен елемент якого є мінімальний із відповідних елементів в цих масивах;

[C,I]=max(A) – крім максимальних значень виводить ще й вектор індексів цих елементів;

[C,I]=min(A) – крім мінімальних значень виводить ще й вектор індексів цих елементів;

median(A) – повертає вектор-рядок медіан для кожного стовпця;

std(X) – виводить вектор-рядок стандартних відхилень елементів кожного стовпця;

sort(A,dim) – здійснює сортування елементів масиву A(dim=1–по стовпцям; dim=2-по рядкам);

sortrows(A) – сортування рядів масиву по зростанню;

[B,index]=sortrows(A) – сортування рядів масиву по зростанню, а також вивід вектора індексів;

fliplr – дзеркальне відображення матриці відносно вертикальної осі;

flipud – дзеркальне відображення матриці відносно горизонтальної осі;

corrcoef(A) – повертає матрицю коефіцієнтів кореляції;

cov(A) – повертає матрицю коефіцієнтів коваріації масиву A.

Хід роботи

1. Завантажити MatLab. Створити «магічний квадрат» A з розмірністю 7.

2. За допомогою подвійного використання функції max вивести найбільший елемент масиву A.

3. За допомогою подвійного використання функції min вивести найменший елемент масиву A.

4. Вивести середнє арифметичне значення масиву A, використовуючи функцію mean.

5. Створити масив A1, розміром 7x7 і заповнити його випадковими числами.

6. Повернути масив того ж розміру, що й A і A1, кожен елемент якого є максимальний із відповідних елементів в цих масивах;

7. Повернути масив того ж розміру, що й A і A1, кожен елемент якого є мінімальний із відповідних елементів в цих масивах;

8. Вивести максимальні елементи стовпців матриці A, а також їх індекси.

9. Вивести мінімальні елементи стовпців матриці A, а також їх індекси.

10. Вивести вектор-рядок медіан для кожного стовпця матриці A;

11. Вивести вектор-рядок стандартних відхилень елементів кожного стовпця матриці A;

12. Здійснити сортування рядів масиву A.

13. Здійснити сортування рядів масиву за зростанням, а також вивести вектор індексів;

14. Виконати дзеркальне відображення матриці А відносно вертикальної осі, здійснити сортування елементів за спаданням;

15. Виконати дзеркальне відображення матриці А відносно горизонтальної осі;

16. Посортувати елементи масиву А по стовпцях.

17. Посортувати елементи масиву А по рядках.

18. Розрахувати матрицю коефіцієнтів кореляції масиву А.

19. Розрахувати матрицю коефіцієнтів коваріації масиву А.

20. Створити масив Р (-1,2,3 ; -5,5,6 ; 3,0,-4).

21. Знайти кількість додатніх елементів масиву Р, використовуючи функції MatLab.

22. Знайти суму додатніх елементів масиву Р, використовуючи функції MatLab.

23. Переставте стовпці матриці Р в порядку зростання сум елементів стовпців, використовуючи функції MatLab.

24. Розрахувати матриці коефіцієнтів кореляції та коваріації масиву Р.

25. Зберегти файл у своїй папці та закрити MatLab.

Запитання для самоконтролю

- Яка матриця називається «магічним квадратом»?
- Як можна створити «магічний квадрат» в системі MatLab?
- Яким чином можна здійснити сортування елементів масиву по стовпцям?
- Як вивести максимальний та мінімальні елементи масиву?
- За допомогою якої функції можна вивести середнє значення елементів масиву?
- Яким чином можна виконати дзеркальне відображення матриці А відносно горизонтальної осі?

Лабораторна робота №4. Багатовимірні масиви

Тема. Багатовимірні масиви

Мета**.** Навчитись створювати багатовимірні масиви та здійснювати над ними основні операції в системі MatLab.

Теоретичні відомості

В MatLab двовимірний масив є частинним випадком багатовимірного масиву. Багатовимірні масиви характеризуються розмірністю, яка є більше 2. Таким масивам можна надати наочну інтерпретацію. Так ,матрицю (двовимірний масив) можна записати на одному листі паперу в вигляді рядків і стовпців, які складаються з елементів матриці. Тоді блокнот з такими листами можна вважати тривимірним масивом, полицю в шафі з блокнотами – чотиривимірним масивом, шафу з такими полицями – п’ятивимірним масивом і т. д.

Під розмірністю масиву розумітимемо число вимірів в просторовому представленні масиву, а під розміром – число рядків і стовпців в кожній розмірності масиву [7].

Наведемо приклад створення тривимірного масиву.

Спочатку створимо звичайний масив:

```
» M=[1 2 3; 4 5 6; 7 8 9]
M =
1 2 3
4 5 6
7 8 9
```

Для добавлення нового шару масиву з тим же розміром можна розширити M наступним чином:

```
» M(:,:,2)=[10 11 12; 13 14 15; 16 17 18]
M(:,:,1) =
1 2 3
4 5 6
7 8 9
```

```
M(:,:,2) =
10 11 12
13 14 15
16 17 18
```

Тепер масив M при явному його заданні матиме наступний вигляд:

```
» M
M(:,:,1)=
1 2 3
4 5 6
7 8 9
M(:,:,2) =
10 11 12
13 14 15
16 17 18
```

Можна легко помітити , що числа в M(:,:, 1) і M(:,: ,2) означають номер шару. Доступ до елементів такого масиву здійснюється аналогічно як і в одно- та двовимірних масивах: M(<номер рядка>,<номер стовпця>,<номер шару>), наприклад,

```
>> M(2,2,1)
ans =
5
```

Завдяки цьому , будь якому елементу можемо присвоїти певне значення:

M(<номер рядка>,<номер стовпця>,<номер шару>)=<потрібне значення>

Видалити з масиву непотрібний шар можна ,присвоївши йому []:

M(:,:,<номер шару>)=[];

M(:,:,2)=1 – створює багатовимірний масив, в якому другий шар повністю заповнений одиницями;

E=ones(m,n,r) – створення багатовимірного масиву, заповненого одиницями;

Z=zeros(m,n,r) – створення багатовимірного масиву, заповненого нулями;

R=randn(m,n,r) – створення багатовимірного масиву, заповненого випадковими числами;

До багатовимірних масивів є можливість застосування операції конкатенації. При цьому перший шар одного масиву буде об’єднуватись з першим шаром другого масиву ,другий шар з другим і так далі. Для даної операції застосовують функцію cat(dim,<масив1>,<масив2>,…,<масивN>).Тип конкатенації залежатиме від значення параметра dim.При dim=1 – вертикальна конкатенація, при dim=2 – горизонтальна. Для визначення розміру та розмірності багатовимірного масиву використовують функцію size(<масив>), яка поверне 3 значення : перше – кількість рядків, друге – кількість стовпців, третє – розмірність масиву [5].

Наприклад застосуємо цю функцію до нашого масиву M:

```
>> size(M)
ans =
3   3    2
```

permute (A, ORDER) – переставляє размірності масиву A в порядку, визначеному вектором перестановок ORDER. Елементи вектора перестановок – числа від I до N, де N – розмірність масиву;

ipermute (A, ORDER) – робить теж саме, тільки в зворотньому порядку.

Хід роботи

1.Завантажити MatLab .Створити двовимірний масив M розміром 3x3.

M =

1	4	7
2	5	8
3	6	9

2. Додати другий шар до даного масиву з елементами:

10	13	16
11	14	17
12	15	18

3. Вивести утворений багатовимірний масив M.

4. Вивести перші елементи кожного шару масиву M.

5. Присвоїти останньому елементу першого шару масиву значення 0.

6. Добавити до масиву третій шар заповнений довільними числами (розмір обов’язково 3x3).

7. Видалити з масиву M перший шар. Прослідкувати за змінами в нумерації шарів .

8. Створити багатовимірний масив, в якому перший шар повністю заповнений одиницями, другий- нулями.

9. Створити багатовимірний масив E повністю заповнений одиницями;

10. Створити багатовимірний масив Z повністю заповнений нулями;

11. Створити багатовимірний масив R повністю заповнений випадковими числами;

12. Створити масив N з 2 шарами:

N(:,:,1)=

4	2	8
4	5	6
0	7	6

N(:,:,2)=

5	2	9
2	5	6
7	8	1

13. Виконати горизонтальну конкатенацію масивів M і N, використовуючи функцію cat.

14. Виконати вертикальну конкатенацію масивів M і N, використовуючи функцію cat.

15. Визначити розмір масиву M(кількість рядків, стовпців, шарів) за допомогою функції size.

16. Створити масиви A, B, C:

A=[1 2; 3 4] ; B=[5 6; 7 8] ; C=[9 10;11 12]

17. Обєднати їх в масив D за допомогою функції cat(3,A,B,C).

18. Вивести розмірність масиву D.

19. Переставити розмірності масиву D у порядку що визначається вектором [3,2,1]. Прослідкувати зміни у розмірності утвореного багатовимірного масиву.

20. Зберегти файл у своїй папці та закрити MatLab.

Запитання для самоконтролю

- Чим характеризуються багатовимірні масиви?
- Яка відмінність між поняттями «розмір» та «розмірність» масиву?
- Яким чином можна здійснити конкатенацію масивів?
- Яким способом можна видалити з багатовимірного масиву непотрібні шари?

Лабораторна робота № 5. Рядкові величини

Тема. Робота з рядковими величинами в системі MatLab.

Мета. Навчитись створювати рядки та здійснювати над ними основні операції в системі MatLab.

Теоретичні відомості

Лінійна структура, доступ до елементів якої не змінює її, - це рядок . Конкатенацією називають операцію приєднання символів. Ця операція не комутативна. Рядком над алфавітом V(нагадаємо, що алфавітом називають скінчену непорожню множину символів) називається або символ з V, або послідовність символів, яку одержали в результаті конкатенації елементів з алфавіту V. Важливою характеристикою рядка є його довжина, яку визначають за кількістю елементів у рядку. Наприклад , довжина рядка ‘123’ дорівнює 3. Рядок нульової довжини називається порожнім. Під час роботи з рядками ,дуже важливим є поняття підрядка [2]. Рядок b є підрядком рядка d ,якщо є такі рядки a та c (можливо порожні) , що d =abc.

В основі представлення символів в рядках лежить їх кодування за допомогою таблиць кодів. Такі таблиці ставлять в однозначній відповідності кожному символу деякий код з значенням від 0 до 255. Вектор, який містить рядок символів, в системі MatLab задається наступним чином:

>>S= 'Any Characters' – вектор ,компонентами якого є числові коди, які відповідають символам.

Конкатенацію рядків можна здійснити за допомогою функції strcat(‘рядок1’,’рядок2’,…,’рядокN’) або утворити масив : [рядок1,’ ’,рядок2], де як бачимо можна добавляти при необхідності пропуски.

Виконати пошук початкових індексів рядка S2 в рядку S1 можна використовуючи функцію findstr(S1,S2). Функція upper(<рядок>) здійснює перетворення малих символів рядка в великі, а великі залишає незмінними.(lower(<рядок>)- навпаки).

Функція strrep(<рядок1>,<рядок2>,<рядок3>) заміняє всі рядки2, який є

підрядками рядка1 на рядок3.

blanks – створення рядка, заповненого пропусками.

deblank – видалення кінцевих пропусків з рядка.

Порівняння рядків можна виконати за допомогою функції strcmp(<рядок1>,<рядок2>), функція поверне значення 1, якщо вони ідентичні і значення 0 в протилежному випадку.

strncmp – порівняння перших n символів рядків.

strcmpi – порівняння рядків без врахувавання регістру.

strncmpi – порівняння перших n символів рядків без врахувавання регістру символів.

Здійснити перетворення числа в рядок можна за допомогою функції num2str(<число>).

MatLab дає можливість перетворення чисел поданих в рядках в інші системи числення. Здійснити перетворення рядка з двійковими цифрами в ціле число десяткової системи можна використовуючи функцію bin2dec (‘число’).

Виконати перетворення числа з десяткової системи в двійковий рядок , можна використавши функцію dec2bin(число) [6].Рядкові вирази MatLab не обчислює, так, наприклад, вивід рядка ' 2+3' просто повторює рядок:

```
» '2+3'
ans =
2+3
```

Однак за допомогою функції eval ('рядковий вираз') рядок, що представляє собою математичний вираз, може бути обчислений:

```
» eval ('2+3')
ans = 5
```

Є можливість створення масиву рядків. При цьому рядки повинні бути однакової довжини. Синтаксис наступний:

```
>>masyv=[‘ryadok1’;’ryadok2’;…;ryadok n’]
```

Коли рядки різної довжини, можна скористатися функцією char, яка зрівняє рядки, додавши до них пробіли:

masyv=char(‘ryadok1’,’ryadok2’,…,ryadok n’)

Аргументами char можуть бути і масиви рядків, що дозволяє поміщати рядки в початок чи кінець уже існуючого масиву.

masyv=char(masyv,’ryadok k’)

Пошук в масиві рядків здійснюється функцією strmatch. Причому будуть виведені номера рядків масиву.Приклад:

```
mas=char(‘липень’,’ листопад’,’ березень’)
>> ind=strmatch('ли',mas)
ind =
    1
    2
```

Хід роботи:

1.Завантажити MatLab. Створити рядки:

S1-Прикарпатський

S2-національний

S3-університет

2. Здійснити конкатенацію рядків, утворивши новий рядок S4.

3. Виконати пошук початкових індексів підрядка S2 в рядку S4, використовуючи функцію findstr.

4.Застосувати функцію lower до рядка S1, функцію upper('str') до рядка S3. Прослідкувати зміни.

5. В рядку S4 замінити слово Прикарпатський на Львівський.

6. Виконати порівняння рядків S2 та S3, використовуючи функцію strcmp.

7. Створити рядок К, заповнений 6-ма пропусками.

8. Створити рядок К1: ‘Студент__’ та вивести його розмір.

9. Видалити з рядка К1 кінцеві пропуски та вивести розмір утвореного рядка.

10. Створити рядок С: ‘величини’ та рядок С1: ‘ВЕЛИЧИНИ’.

11. Виконати порівняння рядків С та С1 з урахуванням регістру символів, та без урахування регістру символів.

12. Здійснити перетворення числа 12345 в рядок(функція num2str).

13. Здійснити перетворення рядка 101 з двійкового рядка в ціле число десяткової системи, використовуючи функцію bin2dec (‘число’).

14. Виконати перетворення числа 12 з десяткової системи в двійковий рядок , використавши функцію dec2bin(число).

15. Створити рядок ‘1+2+3’.Обчислити значення виразу в даному рядку, використовуючи функцію eval .

16. Створити масив рядків names із рядків: ‘Іван’,’Олег’,’Ігор’.

17. Вивести другий рядок даного масиву рядків.

18. Створити масив рядків surnames : ‘Петров’,’Іванов’,’Сидорів’.

19. Вивести даний масив прізвищ, у першому та другому рядках видалити лишні пропуски.

20. Додати до масиву прізвищ своє прізвище.

21.За допомогою вбудованої функції MatLab знайти в масиві прізвище ’Петров’.

22. Зберегти файл у своїй папці та закрити MatLab.

Запитання для самоконтролю

- Яка структура даних називається рядком?
- Як здійснити конкатенацію двох рядків?
- За допомогою якої функції можна вивести довжину рядка?
- Яким чином можна обчислити математичний вираз, записаний в формі рядка?
- Як змінити регістр символів рядка?

Лабораторна робота №6. Структури

Тема. Структури

Мета. Навчитись створювати структури та здійснювати над ними основні операції в системі MatLab.

Теоретичні відомості

Структури належать до складних типів даних(на мові Pascal даний тип організації даних називається записом). Структури дозволяють об’єднувати різнорідні дані в одній змінній і здійснювати доступ до них по іменам. В MatLab для створення структури достатньо присвоїти певному полю потрібне значення. Імена полів записуються після імені структури через крапку.

Наприклад:

```
» man.name='Іван';
» man.date=1956;
```

Вивести цю структуру можна просто, вказавши її ім’я:

```
» man
man =
name: 'Іван'
date: 1956
```

Виведення конкретного компоненту структури здійснюється наступним чином:

```
» man.date
ans =
1956
```

Для створення масиву структур використовують індексацію структур:

```
»man(2).name='Петро';
```

При виведенні структури men MatLab покаже тільки розміри масиву і

список полів.

Число структур в масиві дозволяє знайти функція length(S), де S-назва структури. Структуру також можна створити , використовуючи функцію struct('fieldl' ,'values1', 'field2' ,'values2',...), де в дужках вказані імена полів та їх значення. За допомогою функції isfield(S, 'field') є можливість перевірити чи належить поле структурі S. Вивести імена полів структури S можна за допомогою функції fieldnames (S).Функція getfield(S, 'field'), виводить значення конкретного поля. Функція setfielc(S. 'field' ,V) присвоює полю структури S значення V. За допомогою функції rmfield(S, 'field')можна видалити поле [5].

Хід роботи:

1.Завантажити MatLab. Створити структуру student, використовуючи оператори присвоєння, та викликати цю структуру:

Student:

поле	коментар
name	Ім'я
surname	Прізвище
date	Рік народження
group	Група
height	ріст

2. Викликати компонент student.data та присвоїти йому значення місяця народження. Вивести структуру на екран. Прослідкувати зміни в структурі.

3. Утворити аналогічну структуру student(2), але без двох останніх полів, заповнити її даними про свого одногрупника. Вивести на екран student(2) і student.

4. Вивести кількість структур в даному масиві структур, використовуючи функцію length.

5. Самостійно придумати структуру та створити її, використовуючи

функцію struct('fieldl' ,’values1’, 'field2' ,’values2’,...).

6. За допомогою функції isfield(S, 'field') перевірити чи належить поле name структурі student.

7. Вивести імена полів структури student, за допомогою функції fieldnames (S).

8. Використовуючи функцію getfield(S, 'field'), вивести значення поля surname.

9. Використовуючи функцію setfielc(S. 'field' ,V) присвоїти полю name будь яке інше ім’я.

10. За допомогою функції rmfield(S, 'field') видалити поле height.

11. Описати структуру даних EKZAMENU, яка містить таку інформацію: прізвище студента, група, рік народження, результати складання п'яти іспитів. Заповнити з клавіатури 5 змінних описаного типу. Здійснити необхідну обробку даних та вивести результат на екран.

12. Вивести список студентів групи, назва якої вводиться з клавіатури.

13. Вивести список студентів, які мають трійки.

14. Вивести список студентів, віком від 19 до 22 років.

15. Вивести середній бал здачі сесії студентами.

16.За даними, що вводяться з клавіатури – номер іспиту та оцінка, вивести список студентів, які мають аналогічні дані.

17. Вивести перелік студентів, які склали всі іспити на п'ятірки.

18. Вивести предмет, по яким середній бал здачі найвищий.

19. Зберегти файл у своїй папці та закрити MatLab.

Запитання для самоконтролю

- Який тип даних називається структурою?
- Як можна створити масив структур?
- Як вивести кількість структур в масиві?
- Яким чином можна вивести імена полів структури?
- Як видалити непотрібне поле з структури?

Лабораторна робота №7. Масиви комірок

Тема. Масиви комірок.

Мета. Навчитись створювати масиви комірок та здійснювати над ними основні операції в системі MatLab.

Теоретичні відомості

Масиви комірок, так як і структури, дозволяють об’єднати в одній змінній різнорідні дані. Проте доступ до цих даних здійснюється не по іменам полів, а по числовим індексам. Для того, щоб відрізняти масив комірок від звичайного масиву, його індекси записують не в круглих дужках, а в фігурних. Масиви комірок є найбільш складним типом даних в системі MatLab. Це масив, елементами якого є комірки, що можуть вміщувати будь-які типи масивів, в тому числі і масиви комірок [7].

Функція cell(M,N) створює пустий масив комірок, розміром MxN, який можна заповнити шляхом присвоювання коміркам певних значень. Внесення в комірку певної структури відбувається наступним чином:

```
EXPER{3, 1}.Family = 'Іванов';
EXPER{3, l}.Name= 'Олексій';
EXPER{3, 1}.Group = 201;
```

Функція cellplot(A) дозволяє графічно в окремому вікні відобразити масив комірок з його елементами.

Для створення із масиву символів S строкового масиву комірок використовується функція cellstr(S). Для того щоб перевірити чи даний масив є масивом комірок використовують функцію iscell(A) .

```
» t=iscell(A)
t =
1
```

Поверне значення 1, якщо так і 0 в протилежному випадку.

MatLab дає можливість створення багатовимірних масивів комірок. Для цього можна створити 2 масиви комірок, однакового розміру та об’єднати їх за допомогою функції cat(3,A,B). Трійка в дужках означає, що масив буде тривимірним. Елементом масиву комірок може бути і сам масив комірок. Такі масиви називаються вкладеними [5].

Хід роботи

1.Завантажити MatLab. Створити наступний масив комірок:

A{1,1}=‘Інформатика';

A{1,2}=[1 2;3 4];

A{2,1}=2+3i;

A{2,2}=0:0.1:1

2. Вивести розмір масиву комірок A.

3. Створити попередній масив комірок, використовуючи функцію cell, та заповнити його відповідними значеннями.

4. Відобразити даний масив комірок за допомогою функції celldisp(A).

5. Графічно відобразити даний масив комірок за допомогою функції cellplot(A).

6. Створити рядковий масив комірок, за допомогою функції cellstr(S), використовуючи наступний масив символів.

```
» S={'Факультет.'; 'математики'; 'інформатики'};
```

7. Створити 2 масиви комірок B та B1 розміром 2x2 та заповнити їх довільними даними.

8. Сформувати з них за допомогою функції cat тривимірний масив та вивести його на екран.

9. Вивести перший елемент утвореного масиву.

10. Створити вкладений масив комірок

```
» clear A;
```

```
» A{1,1}=7;

»A{1,2}=B;
```

11. Графічно відобразити даний вкладений масив комірок A за допомогою функції cellplot(A).

12. Створити масив комірок B з наступною структурою

Доц. Петров	Доц. Гринишин	Асс. Зінин
2.2 1.3 0.7	Варіант2	2.0 1.2 0.4
Family Іванов Name Олексій Group 201	Family Сергіев Name Антон Group 202	Family Пашин Name Антон I n f o Вечірн. ф.-т
-1.33 0.35 1.74 0.99 0.98 0.78	-1.43 0.24 1.88 0.90 0.91 0.59	рез. не получений

13. Вивести даний масив комірок за допомогою функції celldisp.

14. Зберегти файл у своїй папці та закрити MatLab.

Запитання для самоконтролю

- Дайте визначення поняттю «масиви комірок».
- Яким чином здійснюється доступ до даних в масивах комірок?
- Як створити порожній масив комірок?
- Як створити багатовимірний масив комірок?
- Яким чином можна графічно відобразити масив комірок?

Лабораторна робота №8. Розріджені матриці

Тема**.** Розріджені матриці

Мета. Вивчити функції роботи з розрідженими матрицями в системі MatLab.

Теоретичні відомості

Матриці з достатньо великою кількістю нулів називаються розрідженими. Для початку розглянемо елементарні розріджені матриці і функції MatLab, що до них відносяться [7].

[B,d] = spdiags(A) – виводить всі ненульові діагоналі з матриці A, розміру mxn. B –матриця розміру min(m,n)xp, стовпці якої p є ненульовими діагоналями A. d –вектор, довжини p, цілочисельні елементи якого точно визначають номера діагоналей матриці A(додатні номера – вище головної діагоналі, від'ємні – нище).

S = speye(m,n) – повертає розріджену матрицю розміру mxn з одиницями на головній діагоналі і нульовими недіагональними елементами.

R = sprandn(S) – повертає матрицю зі структурою розрідженої матриці S, але з елементами, розподіленими по нормальному закону з нульовим середнім і дисперсією, рівною 1.

R = sprandn(m,n,density) – повертає випадкову розріджену матрицю розміру mxn, що містить приблизно densityxmxn нормально розподілених ненульових елементів (0<density<l).

Розглянемо функції перетворення розріджених матриць.

k = find(X) – повертає індекси вектора x для його ненульових елементів. Якщо таких елементів немає, то функція повертає порожній вектор, find(X>100) повертає індекси елементів вектора з X>100.

[i,j] = find(X) – повертає індекси рядка і стовпця для ненульових елементів матриці X.

full(S) – перетворює розріджену матрицю S в повну.

S=sparse(A) – перетворює повну матрицю в розріджену, видаляючи нульові елементи.

S=sparse(i,j,s,m,n,nzmax) – використовуючи вектори i, j, s, генерує розріджену матрицю розміру mxn з ненульовими елементами, кількість яких не перевищує nzmax. Вектори i, j задають позиції елементів і є цілочисельними, а вектор s визначає числове значення елемента матриці, яке може бути дійсним чи комплексним. Вектори i, j, s повинні бути однієї і тієї ж довжини [5].

Розглянемо функції для роботи з ненульовими елементами розріджених матриць.

nnz(X) – повертає число ненульових елементів матриці X.

nnz(X)/numel (X) – щільність розрідженої матриці.

nonzeros(A) – повертає повний вектор-стовбець ненульових елементів матриці A, вибираючи їх послідовно по стовпцям.

S=spalloc(m,n,nzmax) – створює масив для розрідженої матриці S, розміру mxn з простором для розміщення nzmax ненульових елементів, після чого матриця може бути заповнена по стовпцям.

R = spones(S) – генерує матрицю R тої ж розрідженості, що й S, але заміняє на 1 всі ненульові елементи.

Розглянемо функції візуалізації розріджених матриць.

spy(S) – графічно відображає розрідженість матриці S.

spy(S, 'LineSpec' ,markersize) – використовує точно визначені тип, колір і розмір графічного маркера.

Розглянемо алгоритми впорядкування розріджених матриць.

p = colmmd(S) – повертає вектор упорядкованості стовпців розрідженої матриці S.

j = colperm(S) – повертає вектор перестановок j, такий що стовпці матриці S будуть впорядковані по зростанню числа ненульових елементів.

r = symrcm(S) – повертає вектор упорядкованості для симетричної матриці S і називається упорядкуванням Катхілла-Макки.

Наочну інформацію про співвідношення величин елементів матриці дає

функція imagesc, яка інтерпретує матрицю як прямокутне зображення. Кожен елемент матриці представляється у вигляді квадратика,
колір якого відповідає величині елементу.

Хід роботи

1.Завантажити MatLab. Створити матрицю A та вивести всі її ненульові діагоналі.

A =

1	3	4	6	8	0	0
7	8	0	7	0	0	5
0	0	0	0	0	9	8
7	6	54	32	0	9	6

2. Створити розріджену матрицю C, розміру 5x5, з одиницями на головній діагоналі і нульовими недіагональними елементами.

3. Вивести матрицю C з тією ж структурою, але з елементами розподіленими по нормальному закону (з нульовим середнім і дисперсією, рівною 1). Результат присвоїти змінній D.

4. Вивести випадкову розріджену матрицю S, розміру 5x6, що містить приблизно densityxmxn нормально розподілених ненульових елементів (density=0.3).

5. Створити матрицю X і вивести індекси ненульових елементів цієї матриці .

X =

1	2	0
0	0	2
3	4	5

6. Задати вектор У і повернути індекси його ненульових елементів.

7. Виконати попереднє завдання, задавши параметр У>5. Прослідкувати зміни.

8. Перетворити розріджену матрицю C в повну. Результат присвоїти змінній C1.

9. Перетворити матрицю C1 в розріджену, видаливши з неї нульові елементи.

10. Задати вектори i=[1 2], j=[3 4], s=[2 5] і згенерувати розріджену матрицю F, розміром 4x4 з ненульовими елементами, кількість яких не перевищує 5.

11. Вивести число ненульових елементів матриці X, створеної в 5-му завданні.

12. Знайти щільність матриці X.

13. З матриці X згенерувати розріджену матрицю X1, замінивши всі ненульові елементи на 1.

14. Вивести вектор-стовбець ненульових елементів матриці A(матриця A була створена в 1-му завданні).

15. Створити масив для розрідженої матриці G, розміру 6x6, з простором для розміщення 5 ненульових елементів.

16. Графічно відобразити розрідженість матриці F.

17. Вивести вектор упорядкованості стовпців матриці F.

18. Повернути вектор перестановок розрідженої матриці F, такий що стовпці матриці будуть впорядковані по зростанню числа ненульових елементів.

19. Створити симетричну матрицю V і вивести для неї вектор упорядкованості Катхіла-Макки.

V=1 2 3

2 4 5

3 5 6

20. Створити матрицю G=2*eye(7)+diag(ones(1,6),1)+diag(ones(1,6),-1).

21. Графічно відобразити розташування ненульових елементів матриці G.

22. Відобразити матрицю G як прямокутне зображення.

23. Зберегти файл у своїй папці та закрити MatLab.

Запитання для самоконтролю:

- Яка матриця називається розрідженою ?
- Які є функції створення розріджених матриць?
- Яким чином можна здійснити візуалізацію розрідженої матриці?
- За допомогою якої функції можна визначити щільність розрідженої матриці?
- Що здійснює функція sparse(i,j,s,m,n,nzmax) ?

Лабораторні роботи №9-10

Тема: Структура та програмна реалізація бази даних

Мета: закріплення умінь практичного застосування масивів структур, при розробці інформаційних систем на основі баз даних для зберігання інформації.

Завдання: розробити інформаційну систему на основі баз даних для зберігання інформації про wav-файли, яка містить наступні поля:

name \\ назва звуку;

opusZvyky \\ характер звуку;

zminna \\ змінна, якій присвоєний звук;

kilkistZapusiv \\ кількість рядків масиву звуку;

kilkistKanaliv \\ кількість каналів;

chastotaDuskr \\ частота дискретизації;

chusloBitNaRyadok \\ число біт на рядок;

truvalist \\ тривалість в секундах;

masuvZvyky \\ масив звуку;

grafik \\ згенерований графік;

filtrButtervordaZminna \\ змінна, якій присвоєно результат примінення фільтра Баттерворда до звуку;

filtrButtervordaMasuv \\ масив результату процедури фільтрації;

filtrButtervordaGrafik \\ згенерований графік результату фільтрації Баттерворда;

PonuzhChastotaZminna \\ змінна, якій присвоєно результат процедури пониження частоти звуку;

PonuzhChastotaMasuv \\ масив результату процедури пониження частоти звуку;

PonuzhChastotaRozmir \\ розмір масиву звуку після процедури пониження

частоти;

PonuzhChastotaGrafik \\ згенерований графік в результаті процедури пониження частоти;

ZbilwChastotaZminna \\ змінна, якій присвоєно результат процедури збільшення частоти звуку;

ZbilwChastotaMasuv \\ масив результату процедури збільшення частоти звуку;

ZbilwChastotaRozmir \\ розмір масиву звуку після процедури збільшення частоти;

ZbilwChastotaGrafik \\ згенерований графік в результаті процедури збільшення частоти;

BilujShymZminna \\ змінна, якій присвоєно результат накладання білого шуму на звук;

BilujShymMasuv \\ масив результату процедури накладання білого шуму на звук;

BilujShymGrafik \\ згенерований графік в результаті процедури накладання білого шуму на звук;

Основні операції формування та опрацювання структур

При реалізації процедур обробки даних в середовищі MatLab усі вхідні, проміжні та вихідні дані зберігаються у формі масивів структур, що і зумовило необхідність огляду основних операцій.

Структури дозволяють об’єднати в одній змінній дані різних типів і здійснювати доступ до них за іменами. В MatLab для створення структури непотрібно спеціальних оголошень, достатньо присвоїти значення потрібному полю. Імена полів вказуються після імені змінної через крапку.

Масиви структур – це різновид нечислових масивів, елементами яких є значення записів.

Структура - це складний тип даних, що представляє собою сукупність полів.

Поле – це ім’я параметра, що описує об’єкт, ім’ям може бути і назва

деякого масиву. Значення поля – це значення відповідного параметра, значеннями можуть бути і значення елементів відповідного масиву [5].

Значення структури – це сукупність полів і присвоєних їм значень. Масив структур доцільно використовувати для організації бази даних по певних М об'єктах, коли кожен із них характеризується N значеннями параметрів різних типів. Термінологія в цьому випадку узгоджена наступним чином:

- і-й об'єкт = і-му елементу масиву структур = і-й структурі(i = 1,2,…,M);
- кількість об'єктів М = кількості М елементів масиву = кількості М структур;
- і-та структура = сукупності з N полів;
- кількість полів N = кількості параметрів N, що характеризують об'єкт;
- n-е поле = n- му параметру(n=1,2,…,N);
- значення n-го поля = значенню n-го параметра;
- значення і-тої структури = сукупності з N полів і присвоєних їм значень;

Значення кожної і-тої структури формується окремо по кожному N-му полю наступним чином:

<назва масиву (<індекс і >)>.<ім'я n-го поля >= <значення для і -того елемента> , де <назва масиву (<індекс і >)> – і-та структура.

Аналогічні дії необхідно виконати для всіх елементів масиву, для кожного поля. Всього М х N разів. Також можна зарезервувати поле під якийсь параметр, але значення цьому параметру до певного часу не присвоювати. В цьому випадку вказують:

<назва масиву (<індекс і >)>.<ім'я n-го поля >=[].

При створенні даного масиву структур використовуються наступні функції MatLab:

y= wavread('filename') - зчитування wav-файлів;

wavesize=wavread('filename','size') – число каналів і рядків запису;

[y,Fs,bits]=wavread(‘filename’,1) – частота дискретизації і число біт на рядок;

wavesize(1)/Fs – тривалість звуку в секундах;

size(y) – розмір масиву звуку;

plot(y) – побудова графіку звукового сигналу;

sound(y) – прослуховування звукового сигналу;

[b,a]=butter(5,0.2) – генерування фільтру Баттерворда;

s=filter(b,a,y) – накладання фільтру Баттерворда на сигнал;

d=decimate(y,2) – функція пониження частоти дискретизації звуку;

c=interp(y2,2) – функція збільшення частоти дискретизації звукового сигналу;

t=awgn(y,3) – накладання «білого шуму» на сигналу.

Формування масиву структур для одного з звукових сигналів :

```
>> Zvyku(20).name='Door';
>> Zvyku(20).opusZvyky='skrupDverey';
>> Zvyku(20).zminna='y20';
>> Zvyku(20).kilkistZapusiv=32873;
>> Zvyku(20).kilkistKanaliv=1;
>> Zvyku(20).chastotaDuskr=11025;
>> Zvyku(20).chusloBitNaRyadok=8;
>> Zvyku(20).truvalist=2.9817;
>> Zvyku(20).masuvZvyky=y20;
>> Zvyku(20).grafik='D:\Grafiku\Door';
>> Zvyku(20).filtrButtervordaZminna='s20';
>> Zvyku(20).filtrButtervordaMasuv=s20;
>> Zvyku(20).filtrButtervordaGrafik='D:\Grafiku\Door1';
>> Zvyku(20).PonuzhChastotaZminna='d20';
>> Zvyku(20).PonuzhChastotaMasuv=d20;
>> Zvyku(20).PonuzhChastotaRozmir=size(d20);
>> Zvyku(20).PonuzhChastotaGrafik='D:\Grafiku\Door2';
>> Zvyku(20).ZbilwChastotaZminna='c20';
>> Zvyku(20).ZbilwChastotaMasuv=c20;
>> Zvyku(20).ZbilwChastotaRozmir=size(c20);
>> Zvyku(20).ZbilwChastotaGrafik='D:\Grafiku\Door3';
>> Zvyku(20).BilujShymZminna='t20';
>> Zvyku(20).BilujShymMasuv=t20;
```

```
>> Zvyku(20).BilujShymGrafik='D:\Grafiku\Door4';
```

Виведення переліку імен полів:

>> Zvyku

Одночасно з переліком назв полів виводиться розмір масиву структур.

Вивід значеннь і-тої структури:

>> Zvyku (i)

Виведення значень певного поля

>> Zvyku.<назва поля>

Виведення значення певного поля і-тої структури:

>> Zvyku(i).<назва поля>

Побудова графіку :

>>plot(<змінна звуку>)

Існує можливість видалення при необхідності певного поля. Наприклад, для видалення поля з назвами звуків необхідно ввести:
rmfield(Zvyku,’name’). Після чого наша база даних матиме наступну структуру:

```
>> rmfield(Zvyku,'name')
ans =
1x25 struct array with fields:
    opusZvyky
    zminna
    kilkistZapusiv
    kilkistKanaliv
    chastotaDuskr
    chusloBitNaRyadok
    truvalist
    masuvZvyky
    grafik
    filtrButtervordaZminna
    filtrButtervordaMasuv
    filtrButtervordaGrafik
    PonuzhChastotaZminna
    PonuzhChastotaMasuv
    PonuzhChastotaRozmir
```

PonuzhChastotaGrafik

ZbilwChastotaZminna

ZbilwChastotaMasuv

ZbilwChastotaRozmir

ZbilwChastotaGrafik

BilujShymZminna

BilujShymMasuv

ЛІТЕРАТУРА

1. Ситник В.Д., Писаревська Т.А. Основи інформаційних систем: Навч. Посібник. – К.: КНЕУ, 1997.– 252с.
2. Костів О. Структури даних. – Львів: ВЦЛНУ., 2000.
3. Кожевникова Г.П. Структури даних і проектування ефективного обчислювального середовища. – Львів: Вища шк.,1986.
4. Курбатова К. А. MATLAB 7. Самоучитель. – М.: «Диалектика», 2005.
5. Сайт http://atomas.ru/mat/Matlab/
6. "Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем. – М.: Финансы и статистика, 1989.
7. Ануфриев И. Е., Смирнов А. Б., Смирнова Е. MATLAB 7. – СПб.: БХВ-Петербург, 2005.
8. Сергиенко А.Б. Цифровая обработка сигналов – Спб.:Питер, 2002.