

Державний вищий навчальний заклад
«Прикарпатський національний університет імені Василя Стефаника»
Кафедра комп'ютерних наук та інформаційних систем

Матеріали для самостійного вивчення дисципліни
«Теорія програмування»
для студентів спеціальності
122 «Комп'ютерні науки»

Матеріали для самостійного вивчення дисципліни «Теорія програмування»
для студентів спеціальності 122 «Комп'ютерні науки» – 2020. – 17 с.

Розробник:

*Ізмайлов Артем Вікторович, магістр, асистент кафедри комп'ютерних
наук та інформаційних систем.*

© Ізмайлов А. В., 2020 р.

ВСТУП

Серед наукових, однією з найважливіших, є проблема створення ефективних методів розробки програмних систем. Такі методи мають базуватися на розвиненій теорії програмування, яка має на меті дослідження програм та методів їх аналізу і синтезу. У теорії програмування термін «програма» тлумачиться у широкому смислі, охоплюючи, зокрема, як програми конкретних мов програмування, так і великі програмні системи.

Навчальна дисципліна "Теорія програмування" є однією з фундаментальних дисциплін галузі інформаційних технологій і формує важливі навички практичної і наукової діяльності бакалавра напряму підготовки "Інформатика". При вивченні цієї навчальної дисципліни використовуються поняття і методи основ елементарної математики, дискретної математики, алгебри, математичної логіки та теорії алгоритмів, а також основи теорії множин, основи загальної алгебри, теорії алгоритмів, пропозиційної логіки та логіки предикатів.

Теорія програмування є теоретичною основою вивчення курсів «Системне програмування», «Бази даних та інформаційні системи», «Інтелектуальні системи», «Штучний інтелект», «Формальні методи розробки програмних систем», «Інформаційні технології», низки спецкурсів відповідного напряму.

Мета викладання навчальної дисципліни полягає в:

- оволодінні фундаментальними теоретичними поняттями теорії програмування: синтаксис мови програмування, дерево синтаксичного виводу програми, семантична ієрархія Дані – Функції – Композиції – Програмні алгебри, семантичний терм програми, програма, процес програмування;
- формуванні практичних навичок оперування елементами програмної алгебри, складання програм простою мовою програмування, синтаксичного та семантичного аналізу програм, побудови дерева синтаксичного виводу програми, побудови семантичного терму програми та обчислення його значень, доведення часткової та повної коректності програм;
- встановленні предметних зв'язків навчальної дисципліни з різними розділами системного програмування, баз даних та інформаційних систем, штучного інтелекту та інтелектуальних систем, формальних методів розробки програмних систем;
- формуванні практичних навичок застосування теорії програмування для розв'язування прикладних задач природничих, економічних, соціальних та інших наук;
- отриманні студентами теоретичної підготовки і практичних навичок для успішного засвоєння низки інших фундаментальних та спеціальних дисциплін навчального плану, а також для можливості вивчення спеціальної літератури.

Завдання викладання навчальної дисципліни: навчити студентів оперувати елементами програмної алгебри, складати програми простою мовою програмування, проводити синтаксичний та семантичний аналіз програм, будувати дерево синтаксичного виводу програми, будувати семантичний терм програми та обчислювати його значення, доводити часткову та повну коректність програм.

У результаті вивчення навчальної дисципліни студент повинен

знати:

- основні методи синтаксичного та семантичного аналізу програми, побудови програмних алгебр, а також методи побудови дерев синтаксичного виводу і семантичних термів;
- методологічні принципи визначення понять, методи подання синтаксису та семантики мови програмування, а також основні поняття програмування та їх загальні характеристики;
- фундаментальні теоретичні поняття: синтаксис мови програмування, дерево синтаксичного виводу програми, семантична ієрархія Дані – Функції – Композиції – Програмні алгебри, семантичний терм програми, програма, процес програмування;
- основні принципи та методи доведення часткової та повної коректності програм.

вміти:

- оперувати елементами програмної алгебри;
- складати програми простою мовою програмування;
- проводити синтаксичний та семантичний аналіз програми;
- будувати дерево синтаксичного виводу програми;
- будувати семантичний терм програми та обчислювати його значення;
- доводити часткову та повну коректність програм.

ЗМІСТ ЛЕКЦІЙНОГО КУРСУ

Тема 1. Вступ. Теоретичні та прикладні аспекти програмування.

- 1) Предмет теорії програмування. Мета курсу. Зв'язок з іншими дисциплінами. Структура курсу.
- 2) Прикладний та теоретичний аспект програмування, їх взаємозв'язок.
- 3) Чинники, що обґрунтовують важливість теорії програмування: помилки в програмному забезпеченні та їх наслідки, складність програмних систем та необхідність автоматизації їх побудови.

Тема 2. Проста мова програмування SIPL: синтаксис.

- 1) Неформальний опис простої мови програмування.
- 2) Формальний опис синтаксису мови SIPL: БНФ та її властивості.
- 3) Древа синтаксичного виводу.
- 4) Індуктивні визначення синтаксису.

Тема 3. Проста мова програмування SIPL: семантика.

- 1) Методи подання семантики.
- 2) Алгебри даних та функцій.
- 3) Композиції програм.
- 4) Семантичні терми.
- 5) Обчислення значень семантичних термів.

Тема 4. Властивості програм мови програмування SIPL.

- 1) Властивості композицій послідовного виконання та циклу.
- 2) Еквітонність та монотонність програм.
- 3) Часткова та повна коректність програм.

Тема 5. Розвиток основних понять програмування.

- 1) Аналіз словникових визначень поняття програми.
- 2) Розвиток поняття програми з гносеологічної точки зору.
- 3) Початкова тріада понять програмування.
- 4) Тріада прагматичності програм.
- 5) Тріада основних понять програмування.
- 6) Пентада основних понять програмування.

Тема 6. Уточнення основних понять програмування. Основні аспекти програм.

- 1) Пентада програмних понять процесного типу.
- 2) Основні аспекти програм: семіотичні та сутнісні.

Тема 7. Методології програмування.

- 1) Основні методології.
- 2) Імперативне програмування.
- 3) Структурне програмування.
- 4) Процедурне програмування.
- 5) Об'єктно-орієнтоване програмування.
- 6) Декларативне програмування.
- 7) Функціональне програмування.
- 8) Логічне програмування.

ЗМІСТ ТЕМ ЛАБОРАТОРНИХ ЗАНЯТЬ КУРСУ

Лабораторне заняття 1

Побудова і синтаксичний аналіз програм на мові SIPL для задач реалізації елементарних математичних операцій операціями інкременту та декременту

Мета роботи: Навчитись будувати та аналізувати програми на мові SIPL, реалізовувати елементарні математичні операції через операції інкременту та декременту у вигляді програм на мові SIPL.

Хід роботи

- 1 Розробити програму, яка обчислює результат додавання `res` двох невід'ємних цілих чисел використовуючи операції інкременту та/або декременту:
 - 1.1 побудувати програму для розв'язку задачі на мові SIPL,
 - 1.2 побудувати дерево синтаксичного виводу програми,
 - 1.3 перевірити правильність побудованого дерева та зробити відповідні висновки.

Контрольні запитання

1. Як працює операція інкременту?
2. Як працює операція декременту?
3. Яким чином задається синтаксис мови SIPL?
4. Що є деревом синтаксичного виводу програми?
5. Для чого вводиться пріоритет операцій?
6. Які пріоритети введено для операцій мови SIPL?
7. Які синтаксичні категорії та метазмінні введені в мові SIPL?

Лабораторне заняття 2

Побудова і семантичний аналіз програм на мові SIPL для задач реалізації елементарних математичних операцій операціями інкременту та декременту

Мета роботи: Навчитись будувати та аналізувати програми на мові SIPL, реалізовувати елементарні математичні операції через операції інкременту та декременту у вигляді програм на мові SIPL.

Хід роботи

- 1 Здійснити семантичний аналіз розробленої у лабораторній роботі №1 програми, яка обчислює результат додавання `res` двох невід'ємних цілих чисел використовуючи операції інкременту та декременту:
 - 1.1 побудувати композиційний семантичний терм програми,
 - 1.2 на основі побудованого терму зробити висновки щодо адекватності програми поставленому завданню.

Контрольні запитання

1. Як працює операція інкременту?
2. Як працює операція декременту?
3. Для чого вводиться пріоритет операцій?
4. Які пріоритети введено для операцій мови SIPL?
5. Які програмні алгебри пов'язані з мовою SIPL?
6. Як будується семантичний терм програми?

Лабораторне заняття 3

Побудова і тестування програм на мові SIPL для задач реалізації елементарних математичних операцій операціями інкременту та декременту

Мета роботи: Навчитись будувати та аналізувати програми на мові SIPL, реалізовувати елементарні математичні операції через операції інкременту та декременту у вигляді програм на мові SIPL.

Хід роботи

- 1 Здійснити тестування розробленої у лабораторній роботі №1 програми, яка обчислює результат додавання res двох невід'ємних цілих чисел використовуючи операції інкременту та декременту:
 - 1.1 обчислити значення композиційного семантичного терму програми на вхідному даному $[A \rightarrow 14, B \rightarrow 6]$. Літерні позначення змінних у вхідному стані можна замінити тими, які використані у тексті програми з попередніх лабораторних робіт.
 - 1.2 Зробити висновки щодо правильності роботи програми та її адекватності поставленому завданню.

Контрольні запитання

1. Як працює операція інкременту?
2. Як працює операція декременту?
3. Яким чином задається синтаксис мови SIPL?
4. Для чого вводиться пріоритет операцій?
5. Які пріоритети введено для операцій мови SIPL?
6. Які програмні алгебри пов'язані з мовою SIPL?
7. Як будується семантичний терм програми?
8. Як обчислюється значення семантичного терму на вхідному даному?

Лабораторне заняття 4

Побудова і аналіз програм на мові SIPL для задач реалізації алгоритмів виконання типових процедур програмування

Мета роботи: Навчитись будувати та аналізувати програми на мові SIPL, реалізовувати алгоритми виконання типових процедур програмування у вигляді програм на мові SIPL.

Хід роботи

- 1 Розробити програму, яка обчислює значення $N!$ за вхідним цілим N з використанням одного циклу. Результат записувати у змінну `res`:
 - 1.1 побудувати програму для розв'язку задачі на мові SIPL,
 - 1.2 побудувати дерево синтаксичного виводу програми,
 - 1.3 побудувати композиційний семантичний терм програми,
 - 1.4 обчислити значення композиційного семантичного терму на вхідному даному $[N \rightarrow (\text{date} \bmod 3 + \text{month} \bmod 2 + 5)]$. Тут `date` – дата народження, `mod` – операція взяття остачі від ділення, `month` – місяць народження.

Контрольні запитання

1. Що таке факторіал числа n ?
2. Яка формула обчислення факторіалу числа n ?
3. Яким чином задається синтаксис мови SIPL?
4. Що є деревом синтаксичного виводу програми?
5. Для чого вводиться пріоритет операцій?
6. Які пріоритети введено для операцій мови SIPL?
7. Які синтаксичні категорії та метазмінні введені в мові SIPL?
8. Які програмні алгебри пов'язані з мовою SIPL?
9. Як будується семантичний терм програми?
10. Як обчислюється значення семантичного терму на вхідному даному?

Лабораторне заняття 5

Аналіз коректності програм на мові SIPL для задач реалізації алгоритмів виконання типових процедур програмування

Мета роботи: Навчитись аналізувати коректність програми на мові SIPL, реалізовувати алгоритми виконання типових процедур програмування у вигляді програм на мові SIPL.

Хід роботи

- 1 Довести часткову та повну коректність розробленої у лабораторній роботі №4 програми, яка обчислює значення $N!$ за вхідним цілим N з використанням одного циклу і результат записує у змінну `res`.

Контрольні запитання

1. Що таке факторіал числа n ?
2. Яка формула обчислення факторіалу числа n ?
3. Яким чином задається синтаксис мови SIPL?
4. Що є деревом синтаксичного виводу програми?
5. Для чого вводиться пріоритет операцій?
6. Які пріоритети введено для операцій мови SIPL?
7. Які синтаксичні категорії та метазмінні введені в мові SIPL?
8. Які програмні алгебри пов'язані з мовою SIPL?
9. Як будується семантичний терм програми?
10. Дайте визначення частковій коректності програм.
11. Дайте визначення повній коректності програм.

Лабораторне заняття 6

Побудова і аналіз програм на мові SIPL для задач реалізації простих і елементарних функцій

Мета роботи: Навчитись будувати та аналізувати програми на мові SIPL, реалізовувати прості та елементарні функції у вигляді програм на мові SIPL.

Хід роботи

- 1 За допомогою операцій додавання і віднімання розробити програму, яка обчислює результат ділення *res* невід'ємного числа *a* на додатне число *b* та остачу від цього ділення – *ost*:
 - 1.1 побудувати програму для розв'язку задачі на мові SIPL,
 - 1.2 побудувати дерево синтаксичного виводу програми,
 - 1.3 побудувати композиційний семантичний терм програми,
 - 1.4 довести часткову та повну коректність програми.

Контрольні запитання

1. За якою формулою обчислюється ділення з остачею цілого числа?
2. Яким чином задається синтаксис мови SIPL?
3. Що є деревом синтаксичного виводу програми?
4. Для чого вводиться пріоритет операцій?
5. Які пріоритети введено для операцій мови SIPL?
6. Які синтаксичні категорії та метазмінні введені в мові SIPL?
7. Які програмні алгебри пов'язані з мовою SIPL?
8. Як будується семантичний терм програми?
9. Дайте визначення частковій коректності програм.
10. Дайте визначення повній коректності програм.

Лабораторне заняття 7

Побудова і аналіз програм на мові SIPL для задач реалізації логічних предикатів за допомогою базових (Ч.1)

Мета роботи: Навчитись будувати та аналізувати програми на мові SIPL, реалізовувати логічні предикати за допомогою базових у вигляді програм на мові SIPL.

Хід роботи

- 1 Припустимо, що у *SIPL* введено лише деякі базові необхідні предикати. Виразимо решту предикатів через базові: у змінну *res* записати результат обчислення відповідного предикату над значеннями вхідних змінних *A* та *B*. Булеві значення *true* та *false* інтерпретувати як числові «1» та «0» відповідно.

Виконати завдання за номером, отриманим за формулою (процес обчислення навести у звіті): $\text{Num} \bmod 3 + 1$. Тут *Num* – порядковий номер студента у журналі групи, *mod* – операція взяття остачі від ділення.

Завдання:

- 1) Розписати $\dot{\cup}$ через \emptyset і $\dot{\cup}$
- 2) Розписати $=$ через $\dot{\cup}$ і \in
- 3) Розписати $>$ через \emptyset і \in

До отриманого варіанту виконати пункти 1.1 – 1.4:

- 1.1 побудувати програму для розв'язку задачі на мові SIPL,
- 1.2 побудувати дерево синтаксичного виводу програми,
- 1.3 побудувати композиційний семантичний терм програми,
- 1.4 довести часткову та повну коректність програми.

Контрольні запитання

1. Назвіть одну з найголовніших властивостей повної системи логічних функцій.
2. Яким чином задається синтаксис мови SIPL?
3. Що є деревом синтаксичного виводу програми?
4. Для чого вводиться пріоритет операцій?
5. Які пріоритети введено для операцій мови SIPL?
6. Які синтаксичні категорії та метазмінні введені в мові SIPL?
7. Які програмні алгебри пов'язані з мовою SIPL?
8. Як будуються семантичний терм програми?

9. Дайте визначення частковій коректності програм.
10. Дайте визначення повній коректності програм.

Лабораторне заняття 8

Побудова і аналіз програм на мові SIPL для задач реалізації логічних предикатів за допомогою базових (Ч.2)

Мета роботи: Навчитись будувати та аналізувати програми на мові SIPL, реалізовувати логічні предикати за допомогою базових у вигляді програм на мові SIPL.

Хід роботи

- 1 Припустимо, що у *SIPL* введено лише деякі базові необхідні предикати. Виразимо решту предикатів через базові: у змінну *res* записати результат обчислення відповідного предикату над значеннями вхідних змінних *A* та *B*. Булеві значення *true* та *false* інтерпретувати як числові «1» та «0» відповідно.

Виконати завдання за номером, отриманим за формулою (процес обчислення навести у звіті): $\text{Num} \bmod 3 + 1$. Тут *Num* – порядковий номер студента у журналі групи, *mod* – операція взяття остачі від ділення.

Завдання:

- 1) Розписати \neq через \emptyset і $=$
- 2) Розписати $<$ через \dot{U} , 1 і \in
- 3) Розписати \geq через $>$, \dot{U} і $=$

До отриманого варіанту виконати пункти 1.1 – 1.4:

- 1.1 побудувати програму для розв’язку задачі на мові SIPL,
- 1.2 побудувати дерево синтаксичного виводу програми,
- 1.3 побудувати композиційний семантичний терм програми,
- 1.4 довести часткову та повну коректність програми.

Контрольні запитання

1. Назвіть одну з найголовніших властивостей повної системи логічних функцій.
2. Яким чином задається синтаксис мови SIPL?
3. Що є деревом синтаксичного виводу програми?
4. Для чого вводиться пріоритет операцій?
5. Які пріоритети введено для операцій мови SIPL?
6. Які синтаксичні категорії та метазмінні введені в мові SIPL?
7. Які програмні алгебри пов’язані з мовою SIPL?
8. Як будується семантичний терм програми?

9. Дайте визначення частковій коректності програм.
10. Дайте визначення повній коректності програм.

Рекомендована література

Базова література

1. М.С. Нікітченко. Курс лекцій з теорії програмування. Електронний посібник. – Київ, 2008.
2. И.А. Басараб, Н.С.Никитченко, В.Н. Редько. Композиционные базы данных. - К., Либідь, 1992.– 182 с.
3. Дж. Хопкрофт, Р. Мотвани, Д. Ульман. Введение в теорию автоматов, языков и вычислений.– М.: Вильямс, 2002.– 528 с.
4. С. Лавров. Программирование. Математические основы, средства, теория.– СПб.: БХВ-Петербург, 2001.– 320 с.
5. В.М. Глушков, Г.Е. Цейтлин, Е.Л. Ющенко. Алгебра. Языки. Программирование.– К.: Наукова думка, 1974.– 328 с.
6. М. Гросс, А. Лантен. Теория формальных грамматик.– М.: Мир, 1971.–294 с.
7. Э. Дейкстра. Дисциплина программирования. - М., Мир, 1976.
8. Д. Грис. Наука программирования. - М., Мир, 1982.

Допоміжна література

1. Капітонова Ю.В., Кривий С.Л., Летичевський О.А. та ін. Основи дискретної математики. – К., 2002.
2. Катленд Н. Вычислимость. Введение в теорию рекурсивных функций. – М., 1983.
3. Мальцев А.И. Алгебраические системы. – М.: Наука, 1970.
4. Нікітченко М.С., Шкільняк С.С. Математична логіка та теорія алгоритмів: підручник. – К., 2008.– 528 с.
5. The RAISE specification language. Prentice Hall Int.– 1992.– 397 p.

Інформаційні ресурси

www.scientific-library.net – Електронна бібліотека науково-технічної літератури

www.elibrary.ru – Наукова електронна бібліотека науково-технічної літератури