

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«___» _____ 2018 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ТЕОРІЯ ПРОГРАМУВАННЯ**

для студентів

галузь знань **12 «Інформаційні технології»**
(шифр і назва)

спеціальність **122 «Комп'ютерні науки»**
(шифр і назва спеціальності)

освітній рівень **бакалавр**
(молодший бакалавр, бакалавр, магістр)

освітня програма **«Інформатика»**
(назва освітньої програми)

вид дисципліни **обов'язкова**

Форма навчання

денна

Навчальний рік

2018/2019

Семестр

5

Кількість кредитів ECTS

4

Мова викладання, навчання

та оцінювання

українська

Форма заключного контролю

екзамен

Викладачі: **д.ф.-м.н., проф. Нікітченко М.С.** (лекції)

к. ф.-м.н., асистент Федорова М.В. (практичні заняття)

к. ф.-м.н., асистент Криволап А.В. (практичні заняття)

асистент Шишацька О.В. (практичні заняття)

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» ____ 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» ____ 20__ р.
(підпис, ПІБ, дата)

КИЇВ – 2018

Розробник: Нікітченко Микола Степанович, д.ф.-м.н., професор кафедри «Теорії та технології програмування»

ЗАТВЕРДЖЕНО

В.о. зав. кафедри «Теорії та технології програмування»

_____ Панченко Т.В.
(підпис) (прізвище та ініціали)

Протокол № ____ від «____» _____ 20__ р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «____» _____ 20__ року № ____

Голова науково-методичної комісії _____ Хусаїнов Д.Я.
(підпис) (прізвище та ініціали)

Затверджено Вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «____» _____ 20__ року № ____

Голова Вченої ради _____ Анісімов А.В.
(підпис) (прізвище та ініціали)

ВСТУП

1. Мета дисципліни – засвоєння основних теоретичних концепцій, принципів та понять сучасного, зокрема композиційного, програмування; методів формалізації мов програмування та доведення коректності програм.

2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):

1. *Знати:* основні поняття, засоби і методи програмування, їх застосування в інформатиці; знати мови програмування та логіки 1-го порядку, їх можливості для опису предметних областей.
2. *Вміти:* описувати на формальних мовах 1-го порядку твердження стосовно тих чи інших предметних областей та властивостей програм; встановлювати істинність пропозиційних формул, формул 1-го порядку.
3. *Володіти елементарними навичками:* програмування в сучасних мовах, тестування програм, перевірки виконуваності формул.

3. Анотація навчальної дисципліни:

Навчальна дисципліна «Теорія програмування» є складовою освітньо-професійної програми підготовки спеціалістів за освітньо-кваліфікаційним рівнем «магістр» галузі 12 „Інформаційні технології” зі спеціальності 122 „Комп’ютерні науки”, освітньо-професійної програми – „Інформатика”.

Дана дисципліна є обов’язковою навчальною дисципліною за *програмою “Інформатика”*.

Викладається в 5 семестрі 3 курсу бакалаврату в обсязі 120 годин.

(4 кредити ECTS) зокрема: *лекції – 28 год., консультації – 2 год., практичних занять – 26 год., самостійна робота – 64 год.* У курсі передбачено **2 частини** та **2 контрольні роботи**. Завершується дисципліна – **екзаменом в 5 семестрі**.

В результаті вивчення навчальної дисципліни студент повинен:

знати: основні поняття теорії програмування, методи формалізації мов програмування, зокрема формалізації та аналізу семантики та синтаксису програм; теорію найменших нерухомих точок;

вміти: формалізувати синтаксис мов програмування за допомогою БНФ та граматик, робити синтаксичний аналіз програм, будувати семантичний терм програми в алгебрі програм, доводити коректність програм. Формалізувати та досліджувати рекурсивні програми.

.

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні, відповідно освітньої кваліфікації «Бакалавр з комп’ютерних наук».

Зокрема:

- розвивати здатність до абстрактного мислення, аналізу та синтезу;
- здатність до математичного та логічного мислення, формулювання та досліджування математичних моделей, зокрема дискретних математичних моделей, обґрунтування вибору методів і підходів для розв’язування теоретичних і прикладних задач у галузі комп’ютерних наук, аналізу та інтерпретування;
- здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об’єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
RH1.1	Знати основні поняття програмування.	Лекція, практичні заняття	Контрольна робота 60% правильних відповідей, екзамен	20%
RH1.2	Знати методи формалізації мов програмування.	Лекція, практичні заняття	Контрольна робота 60% правильних відповідей, екзамен	20%
RH1.3	Знати методи моделювання предметних областей.	Лекція, практичні заняття	Контрольна робота 60% правильних відповідей, екзамен	20%
RH2.1	Вміти формалізувати мови програмування, моделювати предметні області за допомогою відповідних мов, застосувати засоби аналізу програм.	Лекція, практичні заняття, самостійна робота	Поточне оцінювання, екзамен	20%
RH3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування, розробки специфікацій та програм.	Лекція, практичні заняття	Поточне оцінювання, екзамен	10%
RH4.1	Організовувати свою самостійну роботу для досягнення результату.	Екзамен, самостійна робота	Поточне оцінювання, екзамен	10%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	RH 1.1	RH 1.2	RH 1.3	RH 2.1	RH 3.1	RH 4.1
Програмні результати навчання						
(з опису освітньої програми)						
Застосовувати ґрунтовні знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.	+	+	+	+	+	
Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.				+	+	+

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота 1: РН 1.1., РН 1.2 — 30 балів/18 балів.

2. Контрольна робота 2: РН1.3 - 30 балів/18 балів.

- підсумкове оцінювання (у формі екзамену):

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;

- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН2.1, РН3.1, РН4.1;

- форма проведення і види завдань: письмова.

Види завдань: 4 письмових питання.

- 1 питання: РН1.1, РН3.1, РН4.1;
- 2 питання: РН1.2, РН3.1, РН4.1;
- 3 питання: РН1.3, РН3.1, РН4.1;
- 4 питання: РН2.1, РН3.1, РН4.1;

За розгорнуту відповідь на кожне завдання студент може отримати від 1 до 8 балів.

Критерії оцінювання відповіді студента на питання:

- повнота розкриття питання 1-3 бали;
- логіка викладення 2 бал;
- аналітичні міркування 1-3 бали.

Типове завдання контрольної роботи 1:

1. Описати основні поняття програмування та їх відношення.
2. Побудувати програму на мові *SIPL* для однієї із наступних задач (x, y, n – додатні цілі числа):
 - обчислення $x * y$, використовуючи функції $+$, $-$,
 - обчислення $n!$,
 - обчислення $x - y$, використовуючи функцію віднімання одиниці (-1),
 - перевірки парності числа n ,
 - обчислення $x \div y$, використовуючи функції $+$, $-$,
 - обчислення x^y , використовуючи функції $*$, $+$, $-$,
 - обчислення $[lg n]$, використовуючи функції \div , mod , $+$, $-$,
 - обчислення $x \bmod y$, використовуючи функції $+$, $-$,
 - обчислення 3^x , використовуючи функції $*$, $+$, $-$,
3. Перевірити синтаксичну правильність створеної програми, побудувавши їх дерева синтаксичного виводу. Побудувати семантичні терми цих програм, та застосувати їх до певних вхідних даних.
4. Довести часткову та повну коректність створеної програми.

Теоретичні питання до контрольної роботи 1:

1. Властивості основної пентади програмування.

2. Властивості програмної пентади.
3. Що таке часткова коректність програм? Яким чином доводиться часткова коректність програм?
4. Що таке повна коректність програм? Яким чином доводиться повна коректність програм?
5. Розкрийте зміст формалізації поняття програми.
6. Визначте різні класи функцій.
7. Визначте програмні системи різного рівня абстракції.
8. Дайте визначення класу номінативних даних.

Контрольні запитання до частини 1:

1. Які недоліки неформального опису мов програмування?
2. Яким чином задається синтаксис мови *SIPL*?
3. Що є деревом синтаксичного виводу програми?
4. Коли програма є синтаксично правильною?
5. Що таке неоднозначна БНФ?
6. Для чого вводиться пріоритет операцій?
7. Які пріоритети введено для операцій мови *SIPL*?
8. Які синтаксичні категорії та метазмінні введені в мові *SIPL*?
9. Які типи даних використовуються у мові *SIPL*?
10. Які алгебри даних пов'язані з мовою *SIPL*?
11. Як визначаються операції іменування та розіменування?
12. Які класу функцій використовуються для формалізації семантики мови *SIPL*?
13. Які типи номінативних функцій використовуються для формалізації семантики мови *SIPL*?
14. Як визначається суперпозиція в n -арну функцію?
15. Як визначається композиція присвоєння?
16. Як визначається композиція послідовного виконання?
17. Як визначається композиція розгалуження?
18. Як визначається композиція циклу?
19. Які програмні алгебри пов'язані з мовою *SIPL*?
20. За якими критеріями в перелік композицій включають функції?
21. Як визначається підалгебра алгебри A_Prog , породжена мовою *SIPL*?
22. Як частковість функцій впливає на визначення композицій?
23. Як визначається семантичний терм програми?
24. Як будується семантичний терм програми?
25. Які властивості виконуються для програмних алгебр?
26. Як визначаються монотонні функції?
27. Як визначаються еквітонні функції?
28. Які визначають програми мови *SIPL*?
29. Що таке часткова коректність програм?
30. Що таке повна коректність програм?
31. Розкрийте зміст формалізації поняття програми.
32. Визначте різні класи функцій.
33. Визначте програмні системи різного рівня абстракції.
34. Дайте визначення класу номінативних даних.

Рекомендована література: [1,2,4,5,9,12].

Типове завдання контрольної роботи 2:

1. Побудувати породжуючу граматику G для мови $L = \{ a b^{3n} d c^{2n} e a^n \mid n \geq 0 \}$.
Довести, що $L(G)=L$ та що мова L не є КВ мовою.
2. Побудувати нормальну форму без ε -правил для КВ-граматики з множиною правил:
 $S \rightarrow BbB, S \rightarrow cCc, S \rightarrow AaA, S \rightarrow dDd,$
 $A \rightarrow \varepsilon \mid aA, B \rightarrow \varepsilon \mid bB, C \rightarrow \varepsilon \mid cC, D \rightarrow dD$
3. Побудувати КВ граматику G та систему рівнянь E для мови $L = \{ c^n a^2 b^n \mid n \geq 0 \}$.
Довести незалежно, що $L(G)=L$ та $R(E)=L$ ($R(E)$ – розв’язок системи рівнянь E).
4. Побудувати рекурсивну програму на мові *SIPL* для однієї із наступних задач (x, y, n – додатні цілі числа):
 - обчислення $x * y$, використовуючи функції $+, -$,
 - обчислення $n!$,
 - обчислення $x - y$, використовуючи функцію віднімання одиниці (-1) ,
 - перевірки парності числа n ,
 - обчислення $x \div y$, використовуючи функції $+, -$,
 - обчислення x^y , використовуючи функції $*, +, -$,
 - обчислення $\lfloor \lg n \rfloor$, використовуючи функції $\div, \text{mod}, +, -$,
 - обчислення $x \text{ mod } y$, використовуючи функції $+, -$,
 - обчислення 3^x , використовуючи функції $*, +, -$,Побудувати семантичні терми цих програм, та застосувати їх до певних вхідних даних.
Довести часткову та повну коректність створеної рекурсивної програми.

Контрольні запитання до частини 2:

1. Як визначається синтаксичний аспект мов?
2. Як формулюється принцип відокремлення і єдності синтаксичного та семантичного аспектів.
3. Охарактеризуйте основні моделі формальних мов.
4. Якими методами подається синтаксис мов програмування?
5. Що таке БНФ?
6. Які є модифікації БНФ?
7. Що таке формальна мова?
8. Визначте основні поняття теорії формальних мов.
9. Що таке породжуюча граматики?
10. Розкрийте інтенціональні та екстенціональні аспекти визначення породжуючих граматики.
11. Дайте визначення виводу в граматиці та мови, породженою граматикою.
12. Яким чином доводиться правильність граматики для породження певної мови?
13. Як визначається ієрархія Хомського?
14. Дайте визначення загально-контекстних граматики.
15. Які алгебраїчні операції задані для формальних мов?
16. Які типи автоматів використовують для подання синтаксису мов?
17. Що таке контекстно-вільні граматики?
18. Які властивості контекстно-вільних граматики?
19. Відносно яких операцій клас контекстно-вільних мов є замкненим?

20. Відносно яких операцій клас контекстно-вільних мов не є замкненим?
Дайте визначення різним нормальним формам контекстно-вільних граматик.
21. Сформулюйте розв'язні проблеми для контекстно-вільних граматик.
22. Сформулюйте нерозв'язні проблеми для контекстно-вільних граматик.
23. Як визначається розв'язок системи рівнянь в алгебрах формальних мов?
24. Побудуйте граматики, які породжують мови
 - а) $\{a^{n^*n} \mid n \geq 1\}$;
 - б) $\{w \mid w \in \{a, b, c\}^*, |w|_a = |w|_b = |w|_c\}$;
 - в) $\{ww \mid w \in \{a, b\}^*\}$;
 - г) $\{a^m b^n a^m b^n \mid m \geq 1, n \geq 1\}$.
 Доведіть правильність побудованих граматик. Якщо граматика не є контекстно вільною, доведіть, що контекстно-вільної граматики для вибраної мови не існує.
25. Визначення семантики.
26. Сформулюйте принципи функціональної семантики.
27. Сформулюйте принципи композиційної семантики.
28. Сформулюйте принципи натуральної семантики.
29. Як розуміється еквівалентність композиційної та натуральної семантик?
30. Що таке рекурсивні визначення?
31. Як визначаються повні впорядковані множини?
32. Які властивості мають повні впорядковані множини?
33. Що таке неперервний оператор?
34. Як визначаються нерухомі точки оператора?
35. Сформулюйте теорему Кнастера-Тарського.
36. Яку структуру має множина нерухомих точок неперервного оператора?
37. Як визначається семантика рекурсивних програм?
38. Як можна доводити властивості рекурсивних програм?
39. Сформулюйте принципи денотаційної семантики.
40. Сформулюйте принципи аксіоматичної семантики.
41. Як доводиться коректність програм в аксіоматичній семантиці?
42. Сформулюйте визначення абстрактних типів даних.
43. Сформулюйте принципи іменування.
44. Що таке номінативні дані?
45. Визначте різні типи структур даних як конкретизації номінативних даних.
46. Сформулюйте принципи абстрактних типів даних.
47. Наведіть приклади абстрактних типів даних.
48. Сформулюйте принципи побудови мови VDM.
49. Сформулюйте принципи побудови мови RAISE.
50. Сформулюйте принципи побудови мови B.
51. Сформулюйте принципи побудови мови Z.
52. Що таке уточнення даних?
53. Як доводяться семантичні властивості програм?
54. Які властивості досліджуються при семантичному аналізі програм?

Рекомендована література: [1–5, 7–13].

Запитання для підготовки до екзамену

1. Основні поняття програмування та відношення між ними
2. Основні програмні поняття та відношення між ними
3. Сутнісні та семіотичні аспекти програм
4. Програмні системи та рівні їх абстракції
5. Номінативні дані. Структури даних мов програмування як конкретизації номінативних даних
6. Основні композиції програм
7. Методи подання синтаксису мов програмування
8. Формальні граматики та мови (визначення та класифікація)
9. Визначення та властивості ω -областей
10. Визначення та властивості неперервних відображень
11. Теорема про найменшу нерухому точку (Кнастер-Тарський-Кліні)
12. Теорема про неперервність оператора нерухомої точки
13. Системи рівнянь в формальних мовах
14. Композиційна семантика мов програмування
15. Натуральна семантика мов програмування
16. Денотативна семантика мов програмування
17. Функціональне програмування
18. Аксиоматична семантика
19. Формальні методи розробки програм.
20. Розробка систем в RAISE, та доведення їх семантичних властивостей.
21. Розробка систем в B та доведення їх семантичних властивостей.
22. Розробка систем в Z та доведення їх семантичних властивостей.

Студент не допускається до екзамену, якщо під час семестру набрав менше ніж 24 балів.

7.1 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота 1: до 5 тижня семестру.
2. Контрольна робота 2: до 12 тижня семестру.

Студент має право на одне перескладання кожної контрольної роботи із можливістю отримання максимально 80% початково визначених за цю контрольну роботу балів. Термін перескладання визначається викладачем.

У випадку відсутності студента з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до „Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу” від 1 жовтня 2010 року.

7.2 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59
Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій

№ лекції	Назва лекції	Кількість годин		
		Лекції	Практ. заняття	Сам. р-та
	Частина 1. Основні поняття теорії програмування та їх формалізація			
1	Тема 1. Вступ до предмету. Основні методологічні принципи побудови формальних моделей програм. Мова SIPL.	2		4
2	Тема 2. Формалізація мови SIPL.	2	2	6
3	Тема 3. Алгебри даних та програм.	2	2	4
4	Тема 4. Часткова та повна коректність програм.	2	2	4
5	Тема 5. Предметні області та методи їх опису. Розвиток основних понять програмування	2	2	4
6	Тема 6. Уточнення основних понять програмування	2	2	4
7	Тема 7. Програмні системи різних рівнів абстракції. Мови специфікацій та програмування	2	2	4
	<i>Контрольна робота 1</i>			
	Всього по частині 1	14	12	30
	Частина 2. Синтаксис та семантика програм. Рекурсивні програми.			
8	Тема 8. Основні методи подання синтаксису мов програмування: БНФ та їх модифікації. Формальні мови та граматики. Ієрархія граматик Хомського. Операції над мовами. Автоматні формалізми сприйняття мов.	2	2	4
9	Тема 9. Контекстно-вільні граматики і мови та їх властивості. Нормальні форми. Рівняння в алгебрах формальних мов.	2	2	6
10	Тема 10. Розв'язні та нерозв'язні проблеми теорії формальних мов	2	2	4
11	Тема 11. Функціональна семантика програм, композиційна семантика, денотаційна семантика, аксіоматична семантика. Абстрактні типи даних.	2	2	6
12	Тема 12. Рекурсія в мовах програмування. Повні впорядковані множини. Теореми про найменшу нерухому точку. Застосування теорії найменшої нерухомої точки.	2	2	4
13	Тема 13. Методи семантичного аналізу програм.	2	2	4
14	Тема 14. . Сучасні методи розробки програм та їх формалізація.	2	2	6
	<i>Контрольна робота 2</i>			
	Всього по частині 2	14	14	34
	ВСЬОГО	28	26	64
	Консультація	2		

Загальний обсяг 120 год., в тому числі:

Лекцій – 28 год.

Практичних занять – 26 год

Консультації – 2 год.

Самостійна робота - 64 год.

9. Теми, винесені на самостійне вивчення:

1. Аналіз визначень мов програмування. Математичні поняття, що утворюють базис формалізації програм: множини та функції [9–12].
2. Властивості алгебр даних та функцій. Композиції програм. Семантичні терми. Обчислення значень семантичних термів. [1, 2,9]
3. Доведення властивостей композицій програм. Доведення часткової та повної коректності програм [1, 4, 13].
4. Побудова формальних моделей композиційного типу для фрагментів мов специфікацій та програмування.[1]
5. Побудова граматик та синтаксичних діаграм для фрагментів мов програмування. [1, 6]. Властивості автоматів різних класів. Побудова автоматів для фрагментів мов програмування. [1,3, 6].
6. Доведення властивостей замкненості контекстно-вільних мов. [1,2,6].
7. Побудова розв'язків рівнянь в алгебрах мов. [1,3,6]
8. Дослідження розв'язності проблем різних типів.[1,3,6]
9. Методи збагачення мов програмування та їх формалізація в композиційній семантиці. [1,4].
10. Побудова рекурсивних програм. Дослідження властивостей повних впорядковані множин. [1,9].
11. Неперервні оператори, побудова нерухомих точок. [1].
12. Доведення властивостей рекурсивних програм мови SIPL [1].
13. Основні методи семантичного аналізу програм [1,7,13].
14. Розробка програмних систем в RAISE, B, Z та доведення їх властивостей [1, 13].

10. Рекомендовані джерела:

Основна

1. М.С. Нікітченко. Теорія програмування. Частина 1. Навчальний посібник. – ніжин. Видавництво НДУ імені М.В. Гоголя, 2010.– 119 с.
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.

Додаткова

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