

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теоретичної кібернетики



Олена КАШПУР

2021 року

## РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### Парадигми програмування

#### для студентів

галузь знань	<b>12 «Інформаційні технології»</b> <i>(шифр і назва)</i>
спеціальність	<b>122 «Комп'ютерні науки»</b> <i>(шифр і назва спеціальності)</i>
освітній рівень	<b>бакалавр</b> <i>(молодший бакалавр, бакалавр, магістр)</i>
освітня програма	<b>«Інформатика»</b> <i>(назва освітньої програми)</i>
вибірковий блок	<b>«Інформаційні технології та системи»</b> <i>(назва спеціалізації)</i>
вид дисципліни	<b>вибіркова</b>

Форма навчання	<b>денна</b>
Навчальний рік	<b>2021/2022</b>
Семестр	<b>5</b>
Кількість кредитів ECTS	<b>3</b>
Мова викладання, навчання та оцінювання	<b>українська</b>
Форма заключного контролю	<b>залік</b>

Викладачі: **Анатолій ПАШКО, д.ф.-м.н., проф.** (лекції)

Пролонговано: на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» 20\_\_ р.  
(підпис, ПІБ, дата)


на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» 20\_\_ р.  
(підпис, ПІБ, дата)

КИЇВ – 2021

Розробник: Анатолій ПАШКО, д.ф.-м.н., професор, професор кафедри теоретичної кібернетики

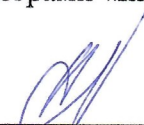
ЗАТВЕРДЖЕНО

Зав. кафедри теоретичної кібернетики

  
Юрій КРАК  
(підпис)

Протокол № 7 від « 8 » 02 2021 р.

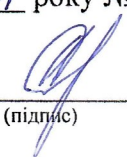
Схвалено Гарантом освітньо-професійної програми «Інформатика»

  
Людмила ОМЕЛЬЧУК  
(підпис)

« 11 » 02 2021 р.

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

Протокол від « 11 » 02 2021 року № 7

Голова науково-методичної комісії   
Людмила ОМЕЛЬЧУК  
(підпис)

« 11 » 02 2021 року

**1. Мета дисципліни** – вивчення основних алгоритмів та методів програмування, оволодіння технікою розробки програмного забезпечення для реалізації сучасних алгоритмів, знайомство з елементами технології створення складних програмних систем в задачах штучного інтелекту.

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

1. *Знати* дискретну математику, математичну логіку, теорію алгоритмів та основи програмування в об'ємі стандартних університетських курсів.

2. *Вміти* застосовувати знання з вказаних вище дисциплін до розв'язання задач.

3. *Володіти елементарними навичками* використання логіко-математичної символіки та основами програмування .

**3. Анотація навчальної дисципліни:**

Навчальна дисципліна розглядає різноманітність стилів і мов програмування, відображає еволюцію мов програмування при розв'язуванні прикладних задач та при розробці програмного забезпечення.

Протягом вивчення курсу студенти мають опанувати основні парадигми програмування. Отримані знання дозволять ефективно застосовувати на практиці методи та алгоритми для розв'язування навчальних та практичних задач, обґрунтовувати власний погляд на розв'язання задач, спілкуватися з колегами з питань програмування, складати звіти з розв'язання задач.

Викладається у 5 семестрі 3 курсу в обсязі – 90 год. (3 кредити ECTS), зокрема: лекції – 38 год., консультації – 2 год., самостійна робота – 50 год.

**4. Завдання (навчальні цілі):**

Набуття поглиблених фахових компетентностей. Зокрема, розвивати:

СК8. 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.

СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

СК17.3. Здатність реалізовувати фази та ітерації життєвого циклу створення програмних систем на основі моделей та методів розробки програмного забезпечення.

СК19.3. Здатність застосовувати математичний апарат та принципи програмування в процесі розробки програмних систем.

**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			
РН1.1	Знати основні методи та алгоритми, їх реалізацію сучасними мовами програмування	Лекція, самостійна робота (СР)	контрольна робота (КР), практичні завдання (ПЗ),	20
РН1.2	Знати принципи застосування основних сучасних алгоритмів при розробці елементів програмного забезпечення	Лекція	КР, ПЗ,	20

РН2.1	Вміти застосовувати сучасні алгоритми при розробці елементів програмного забезпечення	Лекція, самостійна робота	КР, ПЗ	20
РН3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти	Лекція, СР	ПЗ	10
РН4.1	Організувати свою самостійну роботу для досягнення результату	Самостійна робота	ПЗ	10
РН4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Лекція, СР	КР, ПЗ	20

## 6 Співвідношення результатів навчання дисципліни з програмними результатами навчання

Програмні результати навчання <i>(з опису освітньої програми)</i>	Результати навчання дисципліни					
	РН1.1	РН1.2	РН2.1	РН3.1	РН4.1	РН4.2
ПРН4. 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.	+	+	+	+	+	+
ПРН8. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.	+		+		+	
ПРН11. Володіти навичками використання методології управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти готувати проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, креативний бриф, угоду, договір, контракт та ін.).		+	+	+		+
ПРН17.3. Знати математичний апарат та принципи програмування та вміти застосовувати їх у створенні програмних систем.	+		+	+	+	+
ПРН18.3. Знати математичний апарат та принципи програмування та вміти застосовувати їх у створенні програмних систем.	+	+	+	+		+

## 7. Схема формування оцінки.

### 7.1 Форми оцінювання студентів:

#### - семестрове оцінювання:

1. Контрольна робота 1: РН1.1, РН1.2 – 20 балів/10 балів.
2. Контрольна робота 2: РН1.1, РН1.2, РН2.1, РН4.2 – 20 балів/10 балів.
3. Практичне завдання: РН1.1, РН1.2, РН 2.1, РН3.1, РН4.1, РН4.2 – 60 балів/36 балів.

**Типова контрольна робота 1** складається з теоретичних та практичних завдань (з відкритими та закритими відповідями) за матеріалом розділу 1.

*Матеріал, що виноситься на контрольну роботу 1:*  
див. запитання 1-23 для підготовки до оцінювання.

**Типова контрольна робота 2** складається з теоретичних та практичних завдань (з відкритими та закритими відповідями) за матеріалом розділу 1 та 2.

*Матеріал, що виноситься на контрольну роботу 2:*  
див. запитання 1-50 для підготовки до оцінювання.

**Практичне завдання** складається з шести задач, кожна з яких оцінюється максимум в 10 балів, таких типів:

1). Калькулятор працює згідно п'яти сентенцій:

- якщо число оточене дужками, то дужки можна прибрати,
- підвираз виду "число \* число" можна замінити на результат множення,
- підвираз виду "число + число" можна замінити на результат додавання,
- підвираз виду "число - число" можна замінити на результат віднімання.

Запрограмувати такий калькулятор

2). Побудувати калькулятор, що обчислює вирази представлені в зворотній польській формі. Мова програмування C++ (подієве програмування).

3). Нехай на кухню надійшов сигнал - перемістити чайник на деяку конфорку. Кухня перенаправила цю вказівку відповідній плиті. Плита вказала переміститися чайнику. Але чайник не зміг це зробити з деякої причини. Продумати механізм обробки виняткових ситуацій, щоб інформація про неможливість переміщення конкретного чайника дійшла до кухні, дійшла в правильному вигляді, з повною інформацією про неполадку.

Реалізувати функціональності у вигляді набору глобальних функцій, перший аргумент яких є посиланням на базовий об'єкт і дозволяє уникнути дублювання однієї і тієї ж функціональності. Повну інформацію про те, що трапилися неполадки, зберігати всередині структури даних "кухня".

4). Привести реалізацію структури даних **стек** об'єктно-орієнтованою мовою C++.

5). Реалізувати програму «Телефон». Телефон має наступні макро-стани:

- очікування дзвінка
- хтось дзвонить
- йде розмова, встановлено з'єднання
- перегляд телефонної книжки
- навігація по меню.

Потік вхідних даних - це послідовність натиснутих клавіш і (паралельно!) сигнали, що приймаються телефоном.

Глобальними змінними (пам'яттю) є всі налаштування, адресна книга, списки викликів, SMS повідомлення, прапорці про пропущені виклики або отримані SMS повідомлення.

Множина станів телефону – це (множина режимів)\* (множина станів пам'яті).

6). Реалізувати мовою функціонального програмування обчислення факторіала заданого числа, операції додавання, множення, ділення для цілих чисел необмеженої довжини.

У разі неякісного виконання практичних завдань викладач має право не зарахувати його або знизити за нього бали. У разі виникнення підозри щодо несамотійного виконання практичних завдань та/чи контрольних робіт викладач має право виставити бали згідно проведеної із студентом співбесіди (за самим завданням та/або за відповідним теоретичним матеріалом), а також має право запропонувати інше завдання для розв'язання під контролем викладача. У разі відмови від спростування підозри в запропонований викладачем спосіб відповідна робота оцінюється в 0 (нуль) балів.

**- підсумкове оцінювання (у формі заліку):**

Згідно пп. 4.6.1 та 7.1.5 «Положення про організацію освітнього процесу у Київському національному університеті імені Тараса Шевченка» залік виставляється на підставі поточного контролю (див. семестрове оцінювання) як сума оцінок/балів за всіма успішно оціненими результатами навчання; оцінки, нижчі мінімального порогового рівня, до підсумкової оцінки не додаються.

**Запитання для підготовки до оцінювання**

1. Особливості мов програмування
2. Поняття парадигми програмування
3. Розвиток парадигм програмування
4. Розвиток підходів до програмування
5. Критерії якості програм
6. Визначення систем програмування
7. Рівні систем програмування
8. Віденський метод визначення мов програмування
9. Мовно орієнтована абстрактна машина
10. Множина символів формальної системи
11. Інтерпретація або універсальна функція
12. Система базових функцій
13. Низькорівневе програмування
14. Процеси асемблерування
15. Зв'язок асемблера з архітектурою машини
16. Машинно- орієнтоване програмування
17. Машино - незалежне ефективне програмування
18. Мови макрообробки текстів
19. Поняття макропроцесора
20. Поняття макросу
21. Поняття мови управління процесами
22. Типові задачі організації процесів
23. Багатоядерні архітектури
24. Функціональна модель представлення процесів
25. Паралельні процеси, їх взаємодія
26. Особливості функціонального програмування
27. Базові поняття функціонального програмування
28. Списки і атоми
29. Визначення функції вищих порядків у функціональному програмуванні
30. Особливості символічної обробки даних
31. Принципи функціонального програмування
32. Представлення даних у функціональному програмуванні
33. Підходи до системного програмування
34. Проблема спряженості програм, що написані різними мовами
35. Переваги структурного програмування
36. Декларативний стиль програмування
37. Організація недетермінованих обчислень
38. Зв'язок діагностичної інтерпретації і засобів логічного програмування
39. Основні принципи об'єктно –орієнтовного програмування
40. Методи обробки об'єктів заданого класу
41. Класи і екземпляри об'єктів
42. Поняття суперкласу

43. Основні поняття паралельного програмування
44. Переваги і недоліки мов паралельного програмування
45. Функції вищих порядків
46. Регулярна обробка формалізованої інформації
47. Підходи до узагальнення процесів інформаційної обробки
48. Методи динамічної оптимізації обчислень
49. Оптимізація компіляції
50. Лінійні обчислення
51. Компілятор як метод оптимізації процесів
52. Компіляція. Віденський метод.
53. Технології створення прототипів програм
54. Трудомісткість технологій розробки програм
55. Розробка розподілених інформаційних систем

## 7.2 Організація оцінювання:

### Терміни проведення форм оцінювання:

1. Контрольна робота 1: через тиждень після лекції №9.
2. Контрольна робота 2: до 13 тижня семестру включно.
3. Практичні завдання (ч. 1, 2, 3, 4, 5, 6): до 12 тижня семестру включно.

Перескладання контрольних робіт не передбачаються.

Студент має право здавати практичні завдання після закінчення визначеного для них терміну, але не пізніше 13-го навчального тижня включно та з втратою 10% балів за кожен повний чи не повний тиждень, що пройшов від закінчення терміну її здачі.

## 7.3 Шкала відповідності оцінок

Зараховано / Passed	60-100
Не зараховано / Fail	0-59

## 8. Структура навчальної дисципліни. Тематичний план лекцій

№ п/п	Назва теми	Кількість годин	
		Лекції	Самостійна робота
<b>Розділ 1. Основні поняття парадигми програмування.</b>			
1.	<b>Тема 1.</b> Поняття «парадигма програмування» і його історія. Взаємозв'язок мов програмування і парадигм програмування <i>Самостійна робота:</i> Поняття обчислювальної моделі, базових програмних одиниць. Методи розділення абстракцій.	2	2
2.	<b>Тема 2.</b> Визначення мов програмування. <i>Самостійна робота:</i> Парадигма низькорівневого програмування на асемблері	4	6
3.	<b>Тема 3.</b> Машинно орієнтоване програмування. <i>Самостійна робота:</i> Підходи до машинно-незалежного ефективного програмування	4	4
4.	<b>Тема 4.</b> Мови макрообробки текстів <i>Самостійна робота:</i> Структура макропроцесорів нестандартних мов програмування	4	4

5.	<b>Тема 5.</b> Мови управління процесами <i>Самостійна робота:</i> Мови для реалізації операційних систем.	2	6
	<i>Контрольна робота 1</i>		2
Всього по розділу 1		16	24
<b>Розділ 2. Основні види парадигм програмування.</b>			
6.	<b>Тема 1.</b> Функціональне програмування. <i>Самостійна робота:</i> Приклади програмного коду на мовах високого рівня.	6	6
7.	<b>Тема 2.</b> Стандартне програмування. <i>Самостійна робота:</i> Підходи до декларативного програмування.	4	4
8.	<b>Тема 3.</b> Об'єктно-орієнтоване програмування. <i>Самостійна робота:</i> Порівняння програмного коду на мовах об'єктно-орієнтованого програмування	4	4
9.	<b>Тема 4.</b> Мови паралельного програмування <i>Самостійна робота:</i> Функції вищих порядків для організації висококваліфікованих процесів інформаційної обробки.	6	6
10.	<b>Тема 5.</b> Розробка і оптимізація програм <i>Самостійна робота:</i> Тенденції сучасного програмування.	2	4
	<i>Контрольна робота 2</i>		2
Всього по розділу 2		22	26
<b>ВСЬОГО</b>		38	50

**Загальний обсяг 90 год.**, в тому числі:

Лекцій – **38 год.**

Консультації – **2 год.**

Самостійна робота - **50 год.**

*Організація самостійної роботи студентів*

Крім зазначених у таблиці тем на самостійну роботу виносяться детальне опрацювання лекційного матеріалу, а також підготовка до контрольних робіт, виконання оцінюваних практичних завдань.

## 9. Рекомендовані джерела:

### *Основні:*

1. Хопкрофт Дж. Введение в теорию автоматов, языков и вычислений. / Дж. Хопкрофт, Р. Мотвани, Дж. Ульман. – М.: Вильямс, 2002.
2. Ахо А. Компиляторы: принципы, технологии и инструментарий. / А. Ахо, М. Лам, Р. Сети, Дж. Ульман. – М.: Вильямс, 2008.
3. Фридл Дж. Регулярные выражения / Дж. Фридл. – СПб.: Питер, 2018.
4. Кормен Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – М.: Вильямс, 2013.
5. Городня Л.В. Парадигмы программирования: анализ и сравнение. / Л.В. Городня. - Новосибирск: Из-во СО РАН, 2017.
6. Атенсио Л. Функциональное программирование на JavaScript: как улучшить код JavaScript-программ / Л. Атенсио. –М.: Диалектика, 2019.

### *Додаткові:*

7. Йосифович П. Работа с ядром Windows. / П. Йосифович. — СПб.: Питер, 2021.
8. Себеста Р.У. Основные концепции языков программирования./ Р.У. Себеста. - М.: Диалектика-Вильямс, 2001.
9. Уоткинс Д., Программирование на платформе .Net / Д. Уоткинс, М. Хаммонд, Б.М. Эйбрамз. – М.: Вильямс, 2003.
10. Лейнингем И. Освой самостоятельно Python / И. Лейнингем –М.: Вильямс, 2001.
11. Пальмер С.З. Практическое руководство по функционально-ориентированной разработке ПО / С.З. Пальмер, Дж. М. Фелсинг - М.: Вильямс, 2002.
12. Гинзбург С. Математическая теория контекстно-свободных языков./ С. Гинзбург. – М.: Мир, 1970.
13. Филд А. Функциональное программирование / А. Филд, П. Харрисон. - М. Мир, 1993.
14. Пашко А.О. Комп'ютерний менеджмент мобільних систем / А.О. Пашко, О.В. Соломаха, Ю.В. Погорілий // Техніко-технологічні аспекти розвитку та випробування нової техніки і технологій для сільського господарства України. Збірник наукових праць УкрЦВТ. Вип. 5(19). 2002. С.106-111.
15. Пашко А.О. Дослідження методів тестування складних програмних систем / А.О. Пашко, С.В. Михайлюк // Інформаційні технології в економіці, менеджменті і бізнесі. Проблеми науки практики освіти. Матеріали XIV міжнародної науково-практичної конференції, 27 листопада 2008р. Київ. Т1. С. 306-308.

## 10. Додаткові джерела:

<https://drive.google.com/file/d/13nL54FGOrpSifTVNDKxb2i2X1Ycv-khR/view?usp=sharing>