

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
Кафедра теоретичної кібернетики



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ ПРОГРАМУВАННЯ

для студентів

галузь знань	12 «Інформаційні технології» <i>(шифр і назва)</i>
спеціальність	124 «Системний аналіз» <i>(шифр і назва спеціальності)</i>
освітній рівень	бакалавр <i>(молодший бакалавр, бакалавр, магістр)</i>
освітня програма	«Системний аналіз» <i>(назва освітньої програми)</i>
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2023/2024
Семестр	1, 2
Кількість кредитів ECTS	8
з них семестр 1	4
семестр 2	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: **Тетяна КАРНАУХ, к.ф.-м.н., доц.** (лекції, лабораторні заняття)
Ростислав ВЕКЛИЧ (лабораторні заняття)
Микола ЄФРЕМОВ (лабораторні заняття)
Андрій ЛЯШКО (лабораторні заняття)

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

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

КИЇВ – 2023

1. Мета дисципліни – знайомство з початками програмування, базовою термінологією, засобами керування порядком обчислень у мові програмування високого рівня, класичними задачами, оволодіння елементами технології створення програм.

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

1. *Знати* математику та інформатику в об'ємі шкільного курсу.
2. *Вміти* застосовувати знання з природничих дисциплін шкільного курсу до розв'язання задач.
3. *Володіти елементарними навичками* роботи з комп'ютером.

3. Анотація навчальної дисципліни:

Навчальна дисципліна розглядає такі розділи, як зображення числових та інших даних, організація керування порядком обчислень, винятки, підпрограми, рекурсія, ітерація, організація даних, контейнери, ітерування, обробка чисельних даних, обробка файлів, класи та інкапсуляція даних, успадкування, зв'язані структури даних, обробка графів.

Викладається у 1 та 2 семестрах 1 курсу в **обсязі – 240 год. (8 кредитів ECTS)** зокрема: *лекції – 56 год., лабораторні – 58 год., консультації – 4 год., самостійна робота – 122 год.* У курсі передбачено 3 *контрольні роботи* (дві в першому семестрі, одна в другому семестрі). Перший семестр вивчення завершується **заліком**, другий семестр вивчення – **іспитом**.

4. Завдання (навчальні цілі):

- розвивати практичні навички з розв'язування навчальних і практичних задач програмування
- розвивати здатність оцінювати та забезпечувати якість виконуваних робіт

А також розвивати програмні компетентності (подальший перелік наведено згідно освітньої програми):

K02. Здатність застосовувати знання у практичних ситуаціях

K03. Здатність планувати і управляти часом

K07. Здатність до пошуку, оброблення та аналізу інформації з різних джерел

K17. Здатність використовувати системний аналіз як сучасну міждисциплінарну методологію, що базується на прикладних математичних методах та сучасних інформаційних технологіях і орієнтована на вирішення задач аналізу і синтезу технічних, економічних, соціальних, екологічних та інших складних систем.

K21. Здатність формулювати задачі оптимізації при проектуванні систем управління та прийняття рішень, а саме: математичні моделі, критерії оптимальності, обмеження, цілі управління; обирати раціональні методи та алгоритми розв'язання задач оптимізації та оптимального керування.

K22. Здатність до комп'ютерної реалізації математичних моделей реальних систем і процесів; проектувати, застосовувати і супроводжувати програмні засоби моделювання, прийняття рішень, оптимізації, обробки інформації, інтелектуального аналізу даних.

K23. Здатність використовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей та прогнозування поведінки конкретних систем а саме: об'єктно-орієнтований підхід при проектуванні складних систем різної природи, прикладні математичні пакети, застосування баз даних і знань.

K25. Здатність представляти математичні аргументи і висновки з них з ясністю і точністю і в таких формах, які підходять для аудиторії як усно так і в письмовій формі.

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 та принципи розробки програм	Лекція, лабораторне заняття	Контрольна робота (КР), захист програмних проєктів (ПП), іспит	29%
RH1.2	Знати базові класичні алгоритми та різновиди структур даних	Лекція, лабораторне заняття	КР, захист ПП, іспит	18%
RH2.1	Вміти проєктувати, розробляти та тестувати програми	Лабораторне заняття, самостійна робота	КР, захист ПП, іспит	35%
RH2.2	Вміти використовувати прикладні бібліотеки	Лекція, лабораторне заняття, самостійна робота	КР, захист ПП, іспит	6%
RH3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проєктування та розробки програм, складати звіти	Лабораторне заняття	захист ПП	6%
RH4.1	Організувати свою самостійну роботу для досягнення результату	Самостійна робота	захист ПП	3%
RH4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Лабораторне заняття, самостійна робота	захист ПП	3%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	RH 1.1	RH 1.2	RH 2.1	RH 2.2	RH 3.1	RH 4.1	RH 4.2
Програмні результати навчання							
<i>(з опису освітньої програми)</i>							
ПРО8. Володіти сучасними методами розробки програм і програмних комплексів та прийняття оптимальних рішень щодо складу програмного забезпечення, алгоритмів процедур і операцій.	+	+	+	+		+	+
ПР15. Розуміти українську та іноземну мови на рівні, достатньому для обробки фахових інформаційно-літературних джерел, професійного усного і письмового спілкування, написання текстів за фаховою тематикою.					+		

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

Перший семестр

1. Контрольна робота 1: RH1.1, RH2.1 – 20 балів/10 балів.
2. Контрольна робота 2: RH1.1, RH2.1 – 30 балів/15 балів.
3. Програмний проєкт 1: RH1.1, RH2.1, RH3.1, RH4.1, RH4.2 – 10 балів/5 балів.
4. Програмний проєкт 2: RH1.1, RH1.2, RH2.1, RH3.1, RH4.1, RH4.2 – 20 балів/10 балів.
5. Програмний проєкт 3: RH1.1, RH1.2, RH2.1, RH3.1, RH4.1, RH4.2 – 20 балів/10 балів.

Другий семестр

1. Контрольна робота 3: RH1.1, RH1.2, RH2.1, RH2.2 – 15 бали/9 балів.
2. Програмний проєкт 4: RH1.1, RH1.2, RH2.1, RH3.1, RH4.1, RH4.2 – 15 балів/9 балів.
3. Програмний проєкт 5: RH1.1, RH1.2, RH2.1, RH2.2, RH3.1, RH4.1, RH4.2 – 15 балів/9 балів.
4. Програмний проєкт 6: RH1.1, RH1.2, RH2.1, RH2.2, RH3.1, RH4.1, RH4.2 – 15 балів/9 балів.

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

Матеріал, що виноситься на контрольну роботу 1:

див. запитання 1-16 для підготовки до оцінювання.

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

Матеріал, що виноситься на контрольну роботу 2:

див. запитання 1-29 для підготовки до оцінювання.

Типова контрольна робота 3 складається з теоретичних та практичних завдань (з відкритими та закритими відповідями) за матеріалом частин 1-4.

Матеріал, що виноситься на контрольну роботу 3:

див. запитання 1-63 для підготовки до оцінювання.

Конкретні завдання контрольних робіт обираються із загального пулу тестових завдань та задач за відповідним матеріалом випадково та динамічно. Приблизні приклади завдань та матеріали та завдання для підготовки до контрольних розміщуються за посиланням:

https://drive.google.com/drive/folders/0B8q_p3eYFBgvNUhTd210eWdZX1U

Деталізовані умови програмних проєктів та вимоги до них розміщуються за посиланням:

https://drive.google.com/drive/folders/0B8q_p3eYFBgvNUhTd210eWdZX1U

Програмні проєкти мають відповідати умові та варіанту. Варіанти, вимоги та схема оцінювання по кожному проєкту визначається окремо з урахуванням специфіки задачі.

Код програмних проєктів програмно перевіряється на дотримання принципів академічної доброчесності.

Оцінювання програмних проєктів відбувається в три етапи:

1) за наявності програмних засобів, код, що був наданий на перевірку в електронному вигляді за визначеним в умові програмного проєкту протоколом, проходить автоматизоване тестування (якщо хоча б один тест не пройдено, оцінка позитивною бути не може);

2) відбувається перевірка програмного проєкту викладачем на правильність розв'язання та якість коду, за наявності недоліків оцінка може знижуватись;

3) відбувається захист програмного проєкту студентом.

У разі виникнення підозри щодо несамотійного виконання контрольних робіт чи програмних проєктів викладач має право виставити бали згідно проведеної із студентом/студенткою співбесіди (за самим завданням та/або за відповідним теоретичним матеріалом), а також має право запропонувати інше завдання для розв'язання під контролем викладача, у тому числі й онлайн. У разі відмови від спростування підозри в запропонований викладачем спосіб відповідна робота оцінюється в 0 (нуль) балів.

Програмний проєкт може бути позитивно оцінений тільки у випадку, коли студент/студентка дав/дала правильні відповіді на всі запитання щодо структури власного коду, змісту та призначення його елементів, використаних алгоритмів, а також вільно орієнтується у власному коді та здатен/здатна правильно внести невеликі модифікації.

- підсумкове оцінювання:

Перший семестр (у формі заліку)

Згідно «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.

Другий семестр (у формі іспиту)

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

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

- форма проведення: письмова;

- види завдань:

за очної форми проведення іспиту: теоретичне питання (50%) та дві задачі (кожна по 25%, разом 50%);

за дистанційної форми проведення іспиту: 10 тестових запитань із закритими та відкритими відповідями по 4 бали кожне.

Матеріал, що виноситься на іспит:

див. запитання 1-67 для підготовки до оцінювання.

Студент допускається до іспиту, якщо в семестрі набрав не менше 20 балів та отримав позитивну кількість балів хоча б за один програмний проєкт. Для отримання загальної позитивної оцінки з дисципліни оцінка за іспит має бути не менше 24 балів.

Запитання для підготовки до оцінювання

1. Компільовані та інтерпретовані мови.
2. Парадигми програмування.
3. Позиційна система числення, двійковий запис, шістнадцятковий запис.
4. Кодування цілих чисел у комп'ютері.
5. Формати зображення дійсних чисел.
6. Лексеми мови Python. Літерали.
7. Поняття типу даних. Вбудовані типи.
8. Об'єктна система Python. Змінні. Типізація. Присвоювання.
9. Арифметичні операції. Вирази. Пріоритети операторів. Обчислення виразів.
10. Бібліотечні математичні функції.
11. Оператори порівняння (у широкому розумінні мови Python) та булеві оператори.
12. Семантика функції та її виклику, параметри та аргументи.
13. Глобальні, локальні та вільні змінні.
14. Засоби керування порядком обчислень.
15. Винятки.
16. Обробка винятків.
17. Цикл, умова продовження циклу, вигляд і семантика інструкцій циклу.
18. Структурованість інструкцій, переривання циклу.
19. Рекурсивні означення, рекурсивні функції.
20. Глибина рекурсії та загальна кількість рекурсивних викликів, їх вплив на розміри програмного стеку та на час виконання програми.
21. Функціональне проєктування.
22. Ознаки поганого коду.
23. Модулі та пакети.
24. Кортежі, списки.
25. Використання вбудованих ітераторів.
26. Рядки. Основні засоби роботи з ними.
27. Поняття зрізу.
28. Зображення векторів та матриць мовою Python.
29. Багатовимірні масиви бібліотеки numpy.
30. Віртуальне оточення.
31. Часова складність обчислень.
32. Просторова складність обчислень.
33. Емпіричний підхід до визначення складності обчислень.
34. Математичний підхід до визначення складності обчислень.
35. Складність найгіршого випадку.
36. Складність у середньому.
37. Асимптотичні оцінки складності.
38. Алгоритми сортування та їх складність.

39. Механізм наповнювання.
40. Обходи послідовностей.
41. Анонімні функції.
42. Специфіка роботи зі змінюваними значеннями.
43. Множини, словники.
44. Класи та екземпляри (об'єкти). Принцип інкапсуляції.
45. Абстракція даних.
46. Статичні поля, статичні методи, методи класів.
47. Спеціальні методи класів.
48. Ітератори.
49. Обробка файлів (відкриття й закриття потоків, операції введення/виведення, виявлення та обробка помилок).
50. Формати csv, json.
51. Контекстні менеджери.
52. Регулярні вирази. Бібліотека регулярних виразів мови Python.
53. Успадкування класів.
54. MRO.
55. Заміщення методів
56. Поліморфізм, види поліморфізму.
57. Специфіка віртуальних методів.
58. Поняття абстрактного типу даних.
59. Лінійні зв'язані структури даних.
60. Основні операції зі стеками та чергами.
61. Дерева. Обходи дерев.
62. Зображення графів у пам'яті комп'ютера.
63. Алгоритми пошуку в глибину, в ширину.
64. Обробка та візуалізація графів засобами бібліотеки NetworkX.
65. Візуалізація даних засобами бібліотеки matplotlib.
66. Бібліотека pandas.
67. Бібліотека unittest.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

Семестр 1

1. *Контрольна робота 1: до 8-ї лекції семестру.*
2. *Контрольна робота 2: до 14-ї лекції семестру.*
3. *Програмний проєкт (ПП) 1 має бути зданий на перевірку до 7-го лабораторного заняття (ЛЗ) семестру.*
4. *ПП 2 має бути зданий на перевірку до 11-го ЛЗ семестру.*
5. *ПП 3 має бути зданий на перевірку до 13-го ЛЗ семестру.*
6. *Захист програмних проєктів: до 14-го ЛЗ семестру.*

Семестр 2

1. *Контрольна робота 3: до 13-ї лекції семестру.*
2. *ПП 4 має бути зданий на перевірку до 8-го ЛЗ семестру.*
3. *ПП 5 має бути зданий на перевірку до 13-го ЛЗ семестру.*
4. *ПП 6 має бути зданий на перевірку не пізніше ніж за повний робочий тиждень до кінця теоретичного навчання в семестрі.*
5. *Захист програмних проєктів: останній тиждень теоретичного навчання у семестрі.*

Перескладання контрольних робіт не передбачаються. Перескладання програмних проєктів, за які вже були виставлені остаточні бали, з метою збільшення кількості балів не передбачаються. По кожному програмному проєкту повний пакет файлів надсилається

електронною поштою на призначену електронну поштову скриньку. Оцінюється остання надіслана на момент перевірки версія. Датою здачі вважається дата надходження останньої версії.

У першому семестрі:

Оцінка за програмні проекти, що були здані на перевірку невчасно, знижується на 20%. Програмні проекти, що надійшли на перевірку пізніше ніж у робочий день (до 18:00) напередодні заліку, перевіряються на офіційному перескладанні дисципліни (якщо воно буде призначене студенту), при цьому оцінка не може складати більше 60% від максимально можливої.

У другому семестрі:

Оцінка за програмні проекти, що були здані на перевірку невчасно, але не пізніше ніж за повний робочий тиждень до кінця теоретичного навчання в семестрі, знижується на 20%. Програмні проекти, що надійшли на перевірку пізніше ніж за повний робочий тиждень до кінця теоретичного навчання в семестрі, перевіряються на офіційному перескладанні дисципліни (якщо воно буде призначене студенту), при цьому оцінка не може складати більше 60% від максимально можливої.

7.3 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 і лабораторних занять

1-й семестр

№ п/п	Номер і назва теми	Кількість годин		
		лекції	лабораторні	самостійна робота
Частина 1 Організація обчислень				
1	Тема 1. Вступ. Типи даних. Вирази	2		1
2	Тема 2. Встановлення IDE. Знайомство з інтерпретатором Python та інтерфейсом IDE.			1
3	Тема 3. Позиційні системи числення. Зображення цілих чисел у комп'ютері		2	2
4	Тема 4. Зображення дійсних чисел у комп'ютері. Обчислення виразів в інтерпретаторі		2	2
5	Тема 5. Лексичні елементи програми. Модель даних мови Python	2	2	3
6	Тема 6. Функції	2	2	3
7	Тема 7. Умови. Керування порядком обчислень	2	2	3
8	Тема 8. Обробка помилок. Винятки. Елементи технології розробки програмного забезпечення	2	2	3
9	Тема 9. Програмний проєкт 1: обчислення математичних функцій.			4
10	Тема 10. Організація виконуваного коду в інтерпретаторі	2	2	2
11	Тема 11. Робота у віртуальному оточенні. Встановлення сторонніх пакетів Python. Jupyter Notebook			4
12	Тема 12. Рекурсія та ітерація	2	2	4

13	Тема 13. Цикли мови Python	2	2	4
14	Тема 14. Програмний проєкт 2: програмування чисельних алгоритмів.			4
	Контрольна робота 1			1
	Всього по частині 1	16	18	41
Частина 2 Організація даних				
15	Тема 1. Послідовні контейнери	2	2	2
16	Тема 2. Рядки. Обробка послідовностей	2	2	2
17	Тема 3. Обходи послідовностей засобами мови Python	2	2	2
18	Тема 4. Програмний проєкт 3: обробка послідовностей			4
19	Тема 5. Зображення багатовимірних масивів	2		1
20	Тема 6. Елементарна робота з бібліотекою numpy		2	2
21	Тема 7. Можливості бібліотеки numpy			3
22	Тема 8. Поняття складності обчислень.	2	1	2
23	Тема 9. Алгоритми сортування	2	1	2
	Контрольна робота 2			1
	Всього по частині 2	12	10	21
	ВСЬОГО (1-й семестр)	28	28	62

Загальний обсяг (1-й семестр) 120 год., у тому числі:

лекції – **28** год.

лабораторні – **28** год.

консультації – **2** год.

самостійна робота – **62** год.

2-й семестр

№ п/п	Номер і назва теми	Кількість годин		
		лекції	лабораторні	самостійна робота
Частина 3 Класи та структури даних				
24	Тема 1. Асоціативні контейнери. Множини та словники	2		
25	Тема 2. Типи множин мови Python		2	2
26	Тема 3. Тип словників мови Python		2	2
27	Тема 4. Обчислення над моделями математичних об'єктів засобами numpy			2
28	Тема 5. ООП. Класи та об'єкти в Python	2	2	3
29	Тема 6. Проєктування власних класів	2	2	2
30	Тема 7. Спеціальні методи класів. Поліморфізм	2	2	2
31	Тема 8. Успадкування	2	2	2
32	Тема 9. Тестування за допомогою unittest			2
33	Тема 10. Програмний проєкт 4: організація даних			6
34	Тема 11. Лінійні зв'язані структури даних	2	2	3
35	Тема 12. Дерева	2	2	3

№ п/п	Номер і назва теми	Кількість годин		
		лекції	лабораторні	самостійна робота
	Всього по частині 3	14	16	29
Частина 4 Обробка інформації на зовнішніх носіях та прикладні бібліотеки Python				
36	Тема 1. Обробка файлів	2		1
37	Тема 2. Керування ресурсами	2		1
38	Тема 3. Обробка файлів засобами мови Python		2	2
39	Тема 4. Формати csv, json. Робота з файловою системою	2	2	3
40	Тема 5. Регулярні вирази в програмуванні	2	2	4
41	Тема 6. Обробка графів. Алгоритми на графах	2	2	3
42	Тема 7. Зображення та візуалізація графів засобами бібліотеки NetworkX		2	
43	Тема 8. Програмний проєкт 5: обробка графів засобами бібліотеки NetworkX			6
44	Тема 9. Засоби бібліотеки pandas	2	2	2
45	Тема 10. Засоби бібліотеки matplotlib	2	2	2
46	Тема 11. Програмний проєкт 6: обробка та візуалізація даних			6
	Контрольна робота 3			1
	Всього по частині 4	14	14	31
	ВСЬОГО (2-й семестр)	28	30	60
	ВСЬОГО	56	58	122

Загальний обсяг (2-й семестр) 120 год., у тому числі:

лекції – **28** год.

лабораторні – **30** год.

консультації – **2** год.

самостійна робота – **60** год.

Загальний обсяг 240 год., у тому числі:

лекції – **56** год.

лабораторні – **58** год.

консультації – **4** год.

самостійна робота – **122** год.

Організація самостійної роботи студентів

На самостійну роботу виноситься детальне опрацювання лекційного матеріалу та прикладів коду до нього, підготовка до контрольних робіт, розв'язання задач за темами лекцій, виконання поточних домашніх завдань, програмних проєктів та завдань за темами, винесеними на самостійне вивчення.

Детальніше див. додаткові ресурси.

9. Рекомендовані джерела:

Основні:

1. Python 3.12 documentation. – <https://docs.python.org/3/>
2. NumPy. — <https://numpy.org/>
3. Matplotlib. — <https://matplotlib.org/>
4. Pandas. — <https://pandas.pydata.org/>

5. NetworkX. — <https://networkx.org/>
6. Карнаух Т.О. Програмування. Електронний конспект лекцій.
https://drive.google.com/drive/folders/0B8q_p3eYFBgvNUhTd210eWdZX1U
7. Обвінцев, О.В. Інформатика та програмування. Курс на основі Python. Матеріали лекцій : навч. посіб. / О. В. Обвінцев ; Київ. нац. ун-т ім. Тараса Шевченка. - Київ : Основа, 2017.
8. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліфорд Стайн. Вступ до алгоритмів. – К.І.С., 2019. ISBN 9786176842392.

Додаткові:

9. Карнаух, Т.О. Вступ до програмування мовою C++. Організація даних / Т. О. Карнаух, Ю. В. Коваль, М. В. Потієнко, А. Б. Ставровський. – К.: ВПЦ "Київський університет", 2015.
10. Веклич, Р.А. Вступ до програмування мовою C++. Структури даних / Р. А. Веклич, Т. О. Карнаух, А. Б. Ставровський. – К.: ВПЦ "Київський університет", 2018.
11. Heisler F., Amos D., Bader D., Jablonski J. Python basics: A practical introduction to Python 3, 2020.
12. Gamma E., Helm R., Johnson R., Vlissides J. Design patterns: elements of reusable object-oriented software. – Addison-Wesley Professional, 1995.

10. Додаткові ресурси:

https://drive.google.com/drive/folders/0B8q_p3eYFBgvNUhTd210eWdZX1U