

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет

Затверджую
Декан факультету комп'ютерних
інформаційних технологій
Ігор ЯКИМЕНКО
2024 р.



Затверджую
Проректор з
науково-педагогічної роботи
Віктор ОСТРОВЕРХОВ
2024 р.



Затверджую
Директор навчально-наукового інституту
новітніх освітніх технологій
Святослав ПИТЕЛЬ
2024 р.



РОБОЧА ПРОГРАМА
з дисципліни
„Прикладне програмування”

Ступінь вищої освіти бакалавр
Галузь знань 12 Інформаційні технології
Спеціальність 124 Системний аналіз
Освітньо-професійна програма «Системний аналіз»

Кафедра економічної кібернетики та інформатики

Форма навчання/ факультет	Курс	Семестр	Лекції (год)	Практичні заняття (год)	ІРС (год)	Тренінг (год)	СРС	Разом	Залік (сем)
Денна	2	3	30	30	4	8	78	150	3
Заочна	2	3	8	4			138	150	4

30.01.2024

Тернопіль – ЗУНУ, 2024

Робочу програму склала к.е.н, доцент Оксана БАШУЦЬКА

Робоча програма затверджена на засіданні кафедри економічної кібернетики та інформатики, протокол № 1 від 28.08.2024 р.

Завідувач кафедри



Леся БУЯК

Розглянуто та схвалено групою забезпечення спеціальності „Системний аналіз”, протокол № 2 від 30.08.2024 2024 р.

Голова групи забезпечення спеціальності



Роман ПАСІЧНИК

Гарант ОП



Роман ПАСІЧНИК

**СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
"Прикладне програмування"**

1. Опис дисципліни "Прикладне програмування"

Дисципліна «Прикладне програмування»	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни
Кількість кредитів ECTS – 5	Галузь знань – 12 Інформаційні технології	Статус дисципліни: блок вибіркових дисциплін Мова навчання: українська
Кількість залікових модулів – 4	Спеціальність – 124 Системний аналіз.	Рік підготовки: <i>Денна – 2</i> <i>Заочна – 2</i> Семестр: <i>Денна – 3</i> <i>Заочна – 3, 4</i>
Кількість змістових модулів – 2	Ступінь вищої освіти – бакалавр	Лекції: <i>Денна – 30 год</i> <i>Заочна – 8 год</i> Практичні заняття: <i>Денна – 30 год</i> <i>Заочна – 4 год</i>
Загальна кількість годин – 150		Самостійна робота: <i>Денна – 78 год</i> <i>Заочна – 138 год</i> Тренінг: <i>Денна – 8 год</i> Індивідуальна робота: <i>Денна – 4 год</i>
Тижневих годин Денна форма навчання: 3 семестр – 10 год., з них аудиторних – 4 год.		Вид підсумкового контролю – залік

2. Мета й завдання вивчення дисципліни «Прикладне програмування»

2.1. Мета вивчення дисципліни

Метою викладання дисципліни "Прикладне програмування" полягає в формуванні у студентів професійних компетенцій, знань, умінь і навичок володіння мовою програмування та ефективного її використання для розв'язку прикладних завдань.

2.2. Завдання вивчення дисципліни

Завдання вивчення дисципліни полягає в:

- одержанні студентами знань умінь та навичок самостійної розробки програм на Python для різних розрахунків, обробки даних та візуалізації результатів у вигляді графіків;
- робота з інтегрованим середовищем розробки;
- ознайомлення з типовим циклом розробки програмного забезпечення.

2.3. В результаті вивчення дисципліни студенти повинні знати: типи змінних та основні оператори, умовні конструкції та основні типи циклів, методи планування програм, особливості формування зрізів та кортежів, списків та словників, користувацьких функцій, роботи із файлами та об'єктами.

2.4. В результаті вивчення дисципліни студенти повинні вміти: планувати та реалізовувати програмні модулі, будувати користувацькі функції, обробляти файли та виключення, створювати та використовувати програмні об'єкти, працювати із колекціями. Самостійно створювати скрипти та програмні модулі на мові програмування для розв'язання задач різного типу.

3. Програма вивчення дисципліни «Прикладне програмування»

Змістовий модуль 1 – Основи модульного програмування

Тема 1. Встановлення середовища та найпростіші конструкції мови.

Завантаження та інсталяція середовища розробки. Інтерактивний та сценарний режими роботи. Оператор виводу повідомлень. Різні формати повідомлень. Результати математичних операцій.

Тема 2. Змінні та оператори

Змінні та їх ідентифікатори. Правила побудови ідентифікаторів. Ввід та подання інформації від користувача. Стрічкові методи. Операція поєднання стрічок. Складені оператори присвоєння. Генерування випадкових значень.

Тема 3. Умовні конструкції та найпростіші цикли

Умовні конструкції. Конструкція `elif`. Цикл із логічним оператором. Вічний цикл та вихід із нього. Логічне значення арифметичного виразу. Складені логічні вирази. Перевірка реєстраційного імені та пароля.

Тема 4. Планування програм

Важливість планування програм. Приклад алгоритму із повсякденної практики. Покрокове доопрацювання алгоритму. Програма підтримки гри на відгадування.

Тема 5. Зрізи та кортежі

Застосування циклу `for`. Зрізи стрічок. Кортежі. Індксація кортежів. Зрізи кортежів. Гра анаграми.

Тема 6. Списки і словники

Операції із списками. Модифікація гри арсенал. Спискові методи. Умови застосування кортежів та списків. Модифікована програма рекордів. Гра на відгадування слів.

Тема 7. Користувацькі функції

Оголошення функцій. Інкапсуляція функцій. Повторне використання коду. Області видимості. Глобальні змінні. Модифікація глобальних змінних. Умови використання глобальних змінних Реалізація стратегічної гри.

Змістовий модуль 2 – Робота з файлами та об'єктами

Тема 8. Робота із файлами

Способи читання інформації з текстового файлу. Порядкове читання. Запис в текстовий файл. Режими доступу до бінарних файлів. Зберігання та читання об'єктів із файлів.

Тема 9. Виключення

Типи виключень. Аргумент виключень. Блок `else`, `except` в конструкції з

оператором try. Програма підтримки вікторини.

Тема 10. Програмні об'єкти

Класи. Конструктори. Параметр self. Параметри конструктора. Статичні методи. Декоратори. Інкапсуляція об'єктів. Непрямий доступ до атрибутів. Закриті атрибути та методи. Комунікація між об'єктами.

Тема 11. Складні об'єкти

Успадкування об'єктів. Приклади ієрархії об'єктів. Створення модулів. Імпорт модулів. Система модулів.

Тема 12. Робота із колекціями

Множини. Мінливі та незмінні множини. Методи та оператори множин. Генератори множин. Відображення. Словники. Методи словників. Генератори словників. Словники із замовчуваннями. Обхід та копіювання колекцій.

Тема 13. Візуалізація даних в Python.

Побудова гістограм. Побудова кругових діаграм та графіків розсіювання. Побудова полярних діаграм у Python.

Тема 14. Кореляційний аналіз в Python.

Бібліотеки pandas, numpy, matplotlib, seaborn. Обчислення кореляції в Python. Побудова графіка кореляційної карти.

4. Структура залікового кредиту дисципліни

«Прикладне програмування»

(денна форма навчання)

Теми	Кількість годин					
	Лекції	Практичні заняття	ІРС	Тренінг	СРС	Контроль заходів
<i>Змістовий модуль 1 – Основи модульного програмування</i>						
Тема 1. Встановлення середовища та найпростіші конструкції мови	2	2	2	4	6	поточне опит.
Тема 2. Змінні та оператори	2	2			6	поточне опит.
Тема 3. Умовні конструкції та найпростіші цикли	4	4			6	поточне опит.
Тема 4. Планування програм	2	2			6	поточне опит.
Тема 5. Зрізи та кортежі	2	2			6	поточне опит.
Тема 6. Списки і словники	2	2			6	поточне опит.
Тема 7. Користувацькі функції	2	2			6	поточне опит.
<i>Змістовий модуль 2 – Робота з файлами та об'єктами</i>						
Тема 8. Робота із файлами	2	2	2	4	6	поточне опит.
Тема 9. Виключення	2	2			5	поточне опит.
Тема 10. Програмні об'єкти	2	2			5	поточне опит.
Тема 11. Складні об'єкти	2	2			5	поточне опит.
Тема 12. Робота із колекціями	2	2			5	поточне опит.
Тема 13. Візуалізація даних в Python.	2	2			5	поточне опит.
Тема 14. Кореляційний аналіз в Python.	2	2			5	поточне опит.
Всього	30	30	4	8	78	150

(заочна форма навчання)

Теми	Кількість годин			
	Лекції	Практичні заняття	СРС	Контрол заходи
Тема1. Встановлення середовища танайпростіші конструкції мови	0.5	2	8	
Тема 2. Змінні та оператори	0.5		10	
Тема3. Умовні конструкції та найпростіші цикли	1		10	
Тема 4. Планування програм	0.5		10	
Тема 5. Зрізи та кортежі	0.5		10	
Тема 6. Списки і словники	1		10	
Тема 7. Користувацькі функції	0.5		10	
Тема 8. Робота із файлами	0.5	2	10	
Тема 9. Виключення	1		10	
Тема 10. Програмні об'єкти	0.5		10	
Тема 11. Складні об'єкти	0.5		10	
Тема 12. Робота із колекціями	1		10	
Тема 13. Візуалізація даних в Python.			10	
Тема 14. Кореляційний аналіз в Python.			10	
Всього	8	4	138	120

5. Тематика практичних занять

Практичне заняття 1. Змінні та вбудовані функції Python

Ознайомлення з мовою програмування Python. Робота зі змінними та вбудованими функціями Python.

Практичне заняття 2. Розв'язування простих математичних задач на мові Python

Ознайомлення з математичним модулем «math» та його стандартним набором функцій. Використання модуля для вирішення простих математичних задач в Python.

Практичне заняття 3. Оператори порівняння та умовні вирази в Python

Оператори порівняння. Структура умовного оператора if в Python. Оператор try/except.

За допомогою операторів порівняння та логічних операцій реалізувати розгалужені процеси на мові Python.

Практичне заняття 4. Оператори циклу в Python

Основні типи циклів та їх призначення. Оператори циклу while та for в Python. Функція range().

Практичне заняття 5. Функціональне програмування в Python

Поняття функції в мові Python. Анонімні функції в Python. Способи підключення та використання модулів. Інструкція from та формати її використання.

Практичне заняття 6. Робота з файлами та рядками в Python

Робота з файлами: запис даних у файл, читання даних з файлу, виведення змісту, копіювання файлу, перейменування, порівняння. Режими роботи з файлами. Основні операції та методи роботи з рядками.

Практичне заняття 7. Робота з списками в Python

Поняття списку в Python. Створення списку. Основні операції над списками. Основні методи роботи з списками.

Практичне заняття 8. Робота з словниками в Python

Поняття словників та їх призначення в Python. Способи створення словників. Основні операції та методи роботи з словниками. Генератори словників.

Практичне заняття 9. Об'єктно-орієнтоване програмування в Python

Поняття класу в Python. Поля та методи класу. Об'єкт класу. Створення нового класу в Python. Створення об'єкта класу в Python. Відмінність методу від функції. Інкапсуляція, наслідування та поліморфізм.

Практичне заняття 10. Складні об'єкти

Успадкування об'єктів. Приклади ієрархії об'єктів. Створення модулів. Імпорт модулів. Система модулів.

Практичне заняття 11. Робота із колекціями

Множини. Мінливі та незмінні множини. Методи та оператори множин. Генератори множин. Відображення. Словники. Методи словників. Генератори словників. Словники із замовчуваннями. Обхід та копіювання колекцій.

Практичне заняття 12. Візуалізація даних в Python. Matplotlib, основні поняття

Засоби мови Python для роботи з 2D графікою. Зображення графіка математичної функції. Налаштування параметрів графіків функцій (колір графіка,

тип лінії, маркери, легенда). Збереження зображення за допомогою бібліотеки Matplotlib в Python.

Практичне заняття 13. Візуалізація даних в Python.

Побудова гістограм. Побудова кругових діаграм та графіків розсіювання. Побудова полярних діаграм у Python.

Практичне заняття 14. Кореляційний аналіз в Python.

Бібліотеки pandas, numpy, matplotlib, seaborn. Обчислення кореляції в Python. Побудова графіка кореляційної карти.

Практичне заняття 15.

Підсумкове заняття. Модульний контроль.

6. Самостійної роботи студентів.

На самостійну роботу з дисципліни „Прикладне програмування” виносяться індивідуальні практичні завдання, які виконуються самостійно кожним студентом згідно з варіантом, отриманим у викладача. Кожен варіант складається з шести завдань: 1. Обчислення; 2. Масиви; 3. Матриці; 4. Стрічки; 5. Класи; 6. Списки.

7. Організація і проведення тренінгу

Тематика тренінгу: "Розробка фінального проекту на Python: інтеграція алгоритмів, роботи з даними та взаємодії з API"

Мета тренінгу: Закріпити знання з Python шляхом розробки повноцінного проекту, який поєднує алгоритмічні навички, роботу з даними та взаємодію з API. Тренінг спрямований на практичне застосування всіх вивчених тем та створення проекту, який демонструє вміння працювати з реальними даними і задачами.

Методика проведення тренінгу:

Вступна частина.

Огляд ключових тем, які будуть потрібні для виконання завдань: робота з API, маніпуляція даними (pandas), побудова візуалізацій (matplotlib, seaborn), використання функцій та класів. Презентація структури фінального проекту і вимог до нього.

Практична частина.

Індивідуальна робота: кожен студент або невелика група студентів (по 2-3 особи) працюють над своїм проектом. Студенти самі обирають тип задачі (наприклад, обробка даних, візуалізація, створення веб-додатку або взаємодія з API).

Презентація та обговорення результатів.

Кожна група або студент презентують свої рішення. Студенти пояснюють, як працює їх проект, і які технології та бібліотеки вони використали.

Обговорення результатів, аналіз ефективності коду, обмін ідеями та пропозиціями.

Підсумкова частина.

Підведення підсумків тренінгу та курсу в цілому. Рекомендації для подальшого самостійного розвитку в Python.

Завдання для студентів:

Завдання 1: Створення програми для обробки та аналізу даних з API.

Студенти повинні підключитися до API (наприклад, OpenWeatherMap для

отримання прогнозу погоди або Alpha Vantage для фінансових даних), завантажити реальні дані, обробити їх за допомогою бібліотек pandas та побудувати графік з використанням matplotlib або seaborn. Програма повинна мати функції для фільтрації та аналізу даних.

Завдання 2: Розробка програми для автоматизації роботи з файлами.

Написати програму, яка шукає, обробляє та аналізує дані з кількох текстових файлів або CSV-файлів, а потім створює зведений звіт. Завдання включає використання функцій для автоматизованого читання, очищення і запису даних.

Завдання 3: Візуалізація великих наборів даних.

Використовуючи бібліотеки pandas, matplotlib і seaborn, студенти аналізують великий набір даних (наприклад, дані про продажі або соціальні медіа), створюють інтерактивні графіки та представляють ключові інсайти через візуалізацію.

Завдання 4: Створення об'єктно-орієнтованого додатку.

Студенти розробляють об'єктно-орієнтовану програму, наприклад, симулятор простої системи керування замовленнями для онлайн-магазину. Програма повинна включати класи для товарів, замовлень та користувачів, а також мати можливість додавання, видалення і редагування замовлень.

Завдання 5: Створення фінансового аналізатора.

Програма, яка завантажує фінансові дані через API (наприклад, курси валют чи акцій), аналізує їх за допомогою математичних методів та візуалізує динаміку ринку. Можливі функції: прогнозування на основі середньодобових показників або аналіз волатильності.

8. Методи навчання

У навчальному процесі застосовуються: лекції, практичні та індивідуальні заняття, консультації, самостійна робота, робота у групах, методи опитування, моделювання практичних ситуацій, реферування, виконання індивідуальних завдань, підготовка і презентація проектів.

9. Засоби оцінювання та методи демонстрування результатів навчання

У процесі вивчення дисципліни «Прикладне програмування» використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточне опитування;
- модульне оцінювання;
- презентації результатів виконаних завдань на тренінгу;
- оцінювання самостійної роботи;
- контрольна робота.

10. Критерії, форми поточного та підсумкового контролю.

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Прикладне програмування» визначається як середньозважена величина, в залежності від питомої ваги кожної складової залікового кредиту:

Модуль 1		Модуль 2		Модуль 3	Модуль 4
20%	20%	20%	20%	5%	15%
Поточне оцінювання	Модульний контроль	Поточне оцінювання	Модульний контроль	Тренінги	Самостійна робота
Оцінка за поточне оцінювання визначається як середнє арифметичне з оцінок, отриманих під час занять (теми 1- 7). Модульний контроль по змістовому модулю 1		Оцінка за поточне оцінювання визначається як середнє арифметичне з оцінок, отриманих під час занять (теми 7 - 14). Модульний контроль по змістовому модулю 2.		Оцінка, отримана під час тренінгу	Оцінюється як середнє арифметичне з оцінок, отриманих при виконанні завдань що виносяться на самостійну роботу.

Шкала оцінювання:

За шкалою університету	За національною шкалою	За шкалою ECTS
90-100	Відмінно	A (відмінно)
85-89	Добре	B (дуже добре)
75-84		C (добре)
65-74		D (задовільно)
60-64	Задовільно	E (достатньо)
35-59	Незадовільно	FX (незадовільно з можливістю повторного складання)
1-34		F (незадовільно з обов'язковим повторним курсом)

12. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1.	Програмне середовище <i>PYTHON</i>	1-14

Рекомендовані джерела інформації

Основна література

1. Костюченко А. О. Основи програмування мовою Python : навч. посіб. Чернігів: ФОП Баликіна С. М., 2020. – 180 с.
2. Марк Лутц. Python. Довідник програміста. Київ: Науковий світ, 2023. – 294 с.
3. Васильєв О.М. Програмування мовою Python. Тернопіль: Навчальна книга – Богдан, 2022. – 504 с.
4. Ерік Маттес. Пришвидшений курс Python. Практичний проектно-орієнтований вступ до програмування. Львів: Видавництво Старого Лева, 2022. – 600 с.
5. Васильєв О.М. Книга Програмування в PYTHON. Теорія і практика. К: Ліра-К, 2023. – 475 с.
6. Практикум з програмування
<https://pythonexercises.rozh2sch.org.ua/#appendixB>.

Додаткова література

1. Крєневич А. П. Python у прикладах і задачах. Частина 1. Структурне програмування: навч. посіб. Київ: ВПЦ "Київський Університет", 2017. – 206 с.
2. Програмування числових методів мовою Python : підруч. / Анісімов А. В., Дорошенко А. Ю., Погорілий С. Д., Дорогий Я. Ю. Київ: Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с.
3. Інформатика: Комп'ютерна техніка. Комп'ютерні технології: підручник. 4-те вид. / Баженов В. А. та ін. Київ : Каравела, 2012. – 496 с.
4. Войтюшенко Н. М., Остапець А. І. Інформатика і комп'ютерна техніка: навч. посіб. Київ: Центр учбової літератури, 2009. – 564 с.
5. Мельник А. О. Архітектура комп'ютера. Луцьк: Волинська обласна друкарня, 2008. – 470 с.
6. Шеховцов В. А. Операційні системи. Київ: Видавнича група ВНУ, 2008. – 576 с.
7. <https://docs.python.org/uk/3.13/tutorial/index.html>
8. Уроки Python <https://itproger.com/ua/course/python>