



## Силабус освітнього компонента

Програма навчальної дисципліни



# Мультипарадигмальні мови програмування

Шифр та назва спеціальності

122 – Комп'ютерні науки

Інститут

ННІ Комп'ютерного моделювання, прикладної фізики та математики

Освітня програма

Комп'ютерні науки. Моделювання, проектування та комп'ютерна графіка

Кафедра

Комп'ютерне моделювання процесів та систем (162)

Рівень освіти

Бакалавр

Тип дисципліни

Профільні пакети 1 та 4, Вибіркова

Семестр

4

Мова викладання

Українська

## Викладачі, розробники



**Татарінова Оксана Андріївна**

**(відповідальний лектор)**

[oksana.tatarinova@khpi.edu.ua](mailto:oksana.tatarinova@khpi.edu.ua)

кандидат технічних наук, доцент

Спеціаліст з математичного та комп'ютерного моделювання нелінійних процесів. Автор понад 60 наукових статей і доповідей на конференціях, співавтор авторських свідоцтв, монографій, навчальних посібників

[Детальніше про викладача на сайті кафедри](#)



**Овсяніков Владислав Валерійович**

**(асистент з лабораторних робіт)**

[vladyslav.ovsianikov@khpi.edu.ua](mailto:vladyslav.ovsianikov@khpi.edu.ua)

Аспірант, Senior data engineer в Eram systems

[Детальніше про викладача на сайті кафедри](#)

## Загальна інформація

### Анотація

Дисципліна "Мультипарадигмальні мови програмування" зосереджена на вивченні Python - мови, яка демонструє гнучкість мультипарадигмального програмування. Програма дисципліни охоплює вивчення основних парадигм програмування, включаючи об'єктно-орієнтоване, процедурне та функціональне програмування. Особливу увагу приділяється розвитку навичок розробки чистого та ефективного коду, а також використання Python для розв'язання складних алгоритмічних задач.

## Мета та цілі дисципліни

Ознайомлення та оволодіння сучасними методами та теоретичними положеннями, притаманними мультипарадигмальним мовам програмування. Основна увага зосереджується на застосуванні цих знань при побудові алгоритмів та розробці програм, що дозволяє глибше зрозуміти та ефективно використовувати можливості, які надає мова Python у сфері програмування.

## Формат занять

Лекції, лабораторні роботи, самостійна робота, консультації. Підсумковий контроль – екзамен.

## Компетентності

Здатність ефективно використовувати мультипарадигмальні можливості мови Python для аналізу, проектування та реалізації складних програмних систем та компонентів.

ЗК2: Здатність застосовувати знання у практичних ситуаціях.

ЗК3: Знання та розуміння предметної області та розуміння професійної діяльності.

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

## Результати навчання

ПР9: 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

## Обсяг дисципліни

Загальний обсяг дисципліни 150 год. (5 кредитів ECTS): лекції – 16 год., лабораторні роботи – 48 год., самостійна робота – 86 год.

## Передумови вивчення дисципліни (пререквізити)

Для успішного освоєння курсу необхідні знання набути в дисциплінах "Лінійна алгебра", "Об'єктно-орієнтоване програмування та проектування", "Технології програмування", "Організація баз даних", "Математична логіка, теорія алгоритмів та структури даних в частині алгоритмів та структур даних

## Особливості дисципліни, методи та технології навчання

В рамках курсу використовується як основна мова програмування Python та середовище розробки PyCharm Community Edition, що є безкоштовним на базі відкритого вихідного коду.

Лекції проводяться інтерактивно з використанням мультимедійних технологій, де використовуються демонстраційно-ілюстративний підхід, аналіз конкретних прикладів, систематизація та узагальнення теоретичних концепцій, а також ведення дискусій на основі критичного мислення. На лабораторних роботах використовуються методи практичних завдань та вправ, демонстрації робочих проектів, активних дискусій та групової роботи, самостійної роботи студентів та оцінки результатів для навчання мультипарадигмальному програмуванню з фокусом на Python.

Навчальні матеріали доступні студентам на Microsoft OneDrive.

## Програма навчальної дисципліни

### Теми лекційних занять

#### Тема 1. Вступ до програмування мовою Python.

Історія. Область застосування. Місце у сучасному світі. Динаміка та перспективи розвитку.

Парадигми програмування. Встановлення Python. Робота в REPL. Створення та запуск скриптів.

Компіляція, інтерпретація та виконання Python коду. Дізаемблінг коду. Передача параметрів до скрипта. Лексичні категорії. Базовий синтаксис. Деякі вбудовані функції. Стиль кодування.

### Тема 2. Дані та типи.

Змінні та константи. Змінювані та незмінювані сутності. Еквівалентність та ідентичність об'єктів. Базові типи даних. Перевірка та порівняння типів. Конверсія типів. Інтроспекція

### Тема 3. Функції.

Параметри по замовчуванню. Довільна кількість аргументів. Функція як об'єкт. Функція як аргумент іншої функції. Вкладені функції. Замикання. Області видимості. Рекурсія.

### Тема 4. Структури даних. Колекції

Створення рядків. Порядок. Індексція. Slicing. Ітерованість. Незмінність рядків. Операції додавання та множення. Методи рядків. Членство. Метод format. Порівняння рядків. Цикл for. Байти. Масиви байтів

Створення списків. Comprehension. Порядок. Індексція. Slicing. Ітерованість. Змінюваність списків. Операції додавання та множення. Функції над списками. Методи списків. Інкрементальні операції над списками. Порівняння списків. Список як параметр функції. Кортежі.

Створення словників. Змінюваність словників.

Невпорядкованість. Ітерованість. Функції над словниками. Методи словників. Членство.

Порівняння словників. Множини.

### Тема 5. Робота з файлами. Опрацювання виняткових ситуацій. Декоратори. Генератори

Відкриття та читання файлів. Додавання у файл. Запис файлу. Бінарний і текстовий режими.

Ітератори. Генератори-функції. Генераторні вирази. Декоратори. Обробка виняткових ситуацій.

### Тема 6. Модулі і пакети. Тестування та профілювання коду. Регулярні вирази

Бібліотеки numpy, scipy, pandas, matplotlib, seaborn, tkinter.

Модулі. Пакети. Модульні тести. Профілювання коду.

Використання регулярних виразів.

Огляд та використання модульних тестів для валідації коду.

### Тема 7. Огляд систем управління базами даних: SQL та NoSQL.

Використання SQLite у Python.

Робота з MySQL та PostgreSQL з Python

## Теми практичних занять

Не передбачено навчальним планом

## Теми лабораторних робіт

### Тема 1. Вступ до програмування мовою Python

Створення та запуск першого Python-скрипту. Робота з введенням та виведенням даних.

### Тема 2. Дані та типи

Робота з числами та математичними операціями.

Робота зі рядками та рядковими операціями.

Створення та робота із списками.

Використання кортежів у Python.

Основи роботи зі словниками.

### Тема 3. Функції

Створення та використання функцій у Python. Функції з дефолтними параметрами.

Рекурсія та рекурсивні функції.

Лямбда-функції та функції вищого порядку.

### Тема 4. Структури даних. Колекції

Створення та операції зі списками.

Робота із кортежами та їх особливості.

Основи роботи із словниками.

Інтерполяція рядків та робота із текстом.

Використання спискового включення.

### Тема 5. Робота з файлами. Опрацювання виняткових ситуацій.

Введення до роботи з файлами.

Читання та запис даних у текстові файли.

Опрацювання виняткових ситуацій при роботі з файлами.

Робота із бінарними файлами.

**Тема 6. Модулі і пакети. Тестування та профілювання коду.**

Робота з бібліотеками numpy, scipy, pandas.

Візуалізація даних з matplotlib та seaborn.

Розробка простої гри з графічним інтерфейсом.

Модульне тестування програм.

**Тема 7. Огляд систем управління базами даних: SQL та NoSQL.**

Використання SQLite у Python.

Робота з MySQL та PostgreSQL з Python

## Самостійна робота

*Індивідуальне завдання* передбачає розробку програми для аналізу текстових даних. Ключовим елементом цього проєкту є створення скрипта, який може читати, обробляти та аналізувати великі обсяги тексту, наприклад, літературні твори або наукові статті. Студент повинен продемонструвати здатність до ефективної роботи з рядками, використання регулярних виразів для пошуку та фільтрації інформації, а також застосування колекцій для зберігання та обробки даних. Частиною завдання є розробка алгоритму для визначення частотності слів, виявлення ключових слів та фраз у тексті, або аналіз. Важливим аспектом є також створення зрозумілого, добре структурованого коду з використанням функцій та модулів для оптимізації та уникнення повторюваності коду.

*Самостійне вивчення тем та питань, які не викладаються на лекційних заняттях:*

1. Принципи SOLID у контексті Python.
2. Процеси та потоки у Python.
3. Асинхронне програмування з asyncio.
4. Concurrency і паралелізм.
5. Огляд фреймворків Flask, Django.

## Література та навчальні матеріали

1. Васильєв О. Програмування мовою Python. - Тернопіль: Навчальна книга - Богдан, 2019. – С. 504.
2. Довідник з мови Python [Електронний ресурс]. - Режим доступу: <https://docs.python.org/uk/3/reference/index.html#reference-index>
3. Бен С. Python. Збірник вправ. Введення в мову Python із завданнями та рішеннями. – Харків, 2021. – 238 с.
4. Беррі Пол. Head First.- К. “Фабула”, 2021. – 624 с.
5. Висоцька В., Оборська О. Python. Алгоритмізація та програмування.- Н.Світ 2000. – 2021.-514 с.
6. Маттес Е. Пришвидшений курс Python. – Львів.: Видавництво Старого Лева, 2021. – 600 с.

## Система оцінювання

### Критерії оцінювання успішності студента та розподіл балів

Семестровий контроль проводиться в усній формі за екзаменаційними білетами. Результати поточного контролю враховуються як допоміжна інформація для виставлення оцінки з даної дисципліни. Студент вважається допущеним до семестрового екзамену за умови захисту усіх лабораторних робіт та індивідуального завдання. Якщо лабораторна робота пропущена з поважної причини, то вона можуть бути захищена без зменшення кількості балів за неї. Екзамен є обов'язковим. На екзамен виносяться також теми з самостійного опрацювання.

Бали нараховуються наступним чином:

- екзамен – 100% семестрової оцінки

### Шкала оцінювання

Сума балів	Національна оцінка	ECTS
90–100	Відмінно	A
82–89	Добре	B
75–81	Добре	C
64–74	Задовільно	D
60–63	Задовільно	E
35–59	Незадовільно (потрібне додаткове вивчення)	FX
1–34	Незадовільно (потрібне повторне вивчення)	F

## Норми академічної етики і політика курсу

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при неможливості вирішення конфлікту – доводитися до відома співробітників дирекції інституту. Нормативно-правове забезпечення впровадження принципів академічної доброчесності НТУ «ХПІ» розміщено на сайті: <http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/>

## Погодження

Силабус погоджено

28.08.2023

Завідувач кафедри  
Дмитро БРЕСЛАВСЬКИЙ

28.08.2023

Гарант ОП  
Оксана ТАТАРІНОВА