



Факультет математики та інформатики
Кафедра диференціальних рівнянь і прикладної математики

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ НА PYTHON

Рівень вищої освіти: Перший (бакалаврський)

Освітня програма: Прикладна математика, Комп'ютерне моделювання та технології програмування

Спеціальність: F1 Прикладна математика

Галузь знань: F Інформаційні технології

Затверджено на засіданні кафедри
Протокол № 1 від 26 серпня 2025 р.

ЗМІСТ

1. [Загальна інформація](#)
2. [Анотація навчальної дисципліни](#)
3. [Мета та цілі навчальної дисципліни](#)
4. [Результати навчання](#)
5. [Структура навчальної дисципліни](#)
6. [Система оцінювання навчальної дисципліни](#)
7. [Політика навчальної дисципліни](#)
8. [Ресурсне забезпечення навчальної дисципліни](#)

1. ЗАГАЛЬНА ІНФОРМАЦІЯ

| | |
|---|--|
| Назва дисципліни | Об'єктно орієнтоване програмування Object-Oriented Programming |
| Освітні програми | Комп'ютерне моделювання та технології програмування |
| Спеціальність | F1 Прикладна математика |
| Галузь знань | F Інформаційні технології |
| Освітній рівень | перший (бакалаврський) |
| Статус дисципліни | вибіркова |
| Рік навчання/Семестр | 3 ^{ий} / 5 ^{ий} |
| Обсяг дисципліни | 3 кредити ECTS / 90 год. |
| Розподіл годин за видами занять | лекції – 12 год лабораторні – 18 год самостійна робота – 60 год |
| Форма контролю | залік |
| Мова викладання | українська |
| Посилання на сайт дистанційного навчання | https://d-learn.pnu.edu.ua/oop https://classroom.google.com/oop |
| Пререквізити | програмування на Python |
| Постреквізити | – |

| | |
|---------------------------|---|
| Кафедра | диференціальних рівнянь і прикладної математики ауд. 315 ЦК, https://kdrpm.cnu.edu.ua |
| Викладач(-і) | Віктор МАЗУРЕНКО |
| Контактний телефон | (0342)596027 |
| E-mail | viktor.mazurenko@cnu.edu.ua |
| Профайл | https://mazurenko.cnu.edu.ua |
| Консультації | згідно з розкладом консультацій на сайті кафедри |

2. АНОТАЦІЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

У цьому курсі здобувачі вищої освіти за спеціальністю «Прикладна математика» мають нагоду заглибитись у світ об'єктно-орієнтованого програмування (де будь-яку програму розглядають як сукупність «об'єктів», які взаємодіють між собою і кожен з яких є екземпляром якогось «класу»), освоїти його ключові концепції та практики. Така парадигма програмування є хоч і не надто простою, але вельми ефективною при розробці складних програмних проєктів. Завдяки таким ключовим концепціям як «абстрагування», «інкапсуляція», «успадкування» і «поліморфізм», програми, написані в об'єктно-орієнтованому стилі, володіють більшою модульністю і гнучкістю, читабельністю і оптимізованістю коду, захищеністю даних і адаптованістю до змін, ніж програми, написані в інших стилях. Розуміння концепцій об'єктно-орієнтованого програмування, його переваг і недоліків разом з практичними навиками написання програм мовою Python в рамках цієї парадигми дозволить студентові бути успішним в області сучасних технологій програмування.

3. МЕТА І ЦІЛІ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Сформувати у здобувачів вищої освіти фахових компетентностей з об'єктно-орієнтованого програмування для створення програмного забезпечення, яке є більш розширюваним, підтримуваним і зрозумілим як для розробників, так і користувачів.

Ознайомити студентів з основними поняттями і концепціями об'єктно-орієнтованого програмування. Сформувати вміння писати програми мовою Python в об'єктно-орієнтованому стилі для автоматизації задач різної складності. Сприяти розвитку навиків командної роботи та спільної програмної розробки.

4. РЕЗУЛЬТАТИ НАВЧАННЯ

В результаті вивчення навчальної дисципліни здобувач вищої освіти має:

- розуміти концепції об'єктно-орієнтованого програмування: абстрагування, інкапсуляцію, успадкування, поліморфізм;
- пояснювати принципи проєктування програмних систем із використанням спеціальних методів, перевантаження операторів, ітераторів, генераторів, абстрактних класів, інтерфейсів і UML-діаграм класів;
- розробляти модульні програмні рішення з використанням функцій (включно з рекурсивними, генераторними та декорованими) і класів, реалізуючи механізми інкапсуляції, успадкування та поліморфізму;
- проєктувати структуру програмної системи, створювати ієрархії класів, реалізовувати перевизначення методів, обробку виключень, перевантаження операторів і документувати структуру засобами UML;
- представляти результати розробки у формі програмного коду та UML-діаграм, дотримуючись професійної термінології та стандартів оформлення.

5. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

| ТЕМА, ПЛАН | КІЛЬКІСТЬ ГОДИН | | |
|---|-----------------|-------------|-------------------|
| | ЛЕКЦІЇ | ЛАБОРАТОРНІ | САМОСТІЙНА РОБОТА |
| ФУНКЦІЇ - поняття підпрограми і функції, визначення і виклик функцій, повернення кількох значень - аргументи (параметри) функції: формальні і фактичні, передача аргументів: позиційні та іменовані (ключові) аргументи; типові (за замовчуванням) значення аргументів - функції з довільною кількістю аргументів, області видимості: локальні і глобальні змінні - поняття про функційний тип даних, анонімні (lambda-функції) - рекурсія і рекурсивні функції, генератор-функції - декоратори для функцій | 2 | 2 | 8 |
| КЛАСИ І ОБ'ЄКТИ, АТРИБУТИ І МЕТОДИ - основні парадигми програмування та об'єктно-орієнтований стиль програмування - класи і об'єкти (екземпляри) класу, атрибути (поля і методи) класу/об'єкта, абстрагування і абстракція - створення класу, посилання self на об'єкт класу, конструктор __init__ (ініціалізація) і деструктор __del__ (знищення) об'єкта - створення і знищення об'єктів - звернення до атрибутів і методів об'єкта, атрибути об'єкта і атрибути класу, додавання і вилучення атрибутів і методів - методи і функції | 2 | 4 | 8 |
| КОНЦЕПЦІЯ ІНКАПСУЛЯЦІЇ - інкапсуляція - статичні атрибути і методи - вступ до UML: діаграма класів, видимість атрибутів і методів, відношення між класами | 2 | 2 | 8 |
| КОНЦЕПЦІЯ УСПАДКУВАННЯ - успадкування (наслідування) - перевизначення атрибутів і методів - особливості роботи з методом-конструктором, виклик методів базового класу, суперклас object - успадкування на діаграмах класів - множинне і багатократне успадкування - проблеми, пов'язані з множинним успадкуванням - виклик методів базових класів | 2 | 4 | 8 |

| | | | |
|--|-----------|-----------|-----------|
| КОНЦЕПЦІЯ ПОЛІМОРФІЗМУ - поліморфізм - віртуальні методи - спеціальні атрибути - спеціальні методи - спеціальні методи для колекцій - перевантаження операторів (арифметичні, присвоєння, порівняння, унарні) - ітератори і генератори | 2 | 2 | 8 |
| КЛАСИ ВИКЛЮЧЕНЬ І АБСТРАКТНІ КЛАСИ - класи виключень - абстрактні класи - інтерфейси та шаблони програмування - домішки | 2 | 2 | 8 |
| МОДУЛЬНИЙ КОНТРОЛЬ | – | 2 | 12 |
| ВСЬОГО | 12 | 18 | 60 |

6. СИСТЕМА ОЦІНЮВАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

| | |
|------------------------------------|--|
| Загальна система оцінювання | Підсумкова оцінка з навчальної дисципліни є сумою оцінок з таких видів навчальних робіт: лабораторні роботи (виконання і захист), самостійна робота (за окремими темами), модульний контроль (тест і контрольна робота). Підсумкова оцінка визначається відповідно до поданої нижче таблиці оцінювання за різними шкалами (100-бальна, ECTS, національна). |
| Лабораторні роботи | Максимальна оцінка за вчасно виконані і захищені лабораторні роботи становить 50 балів. |
| Модульний контроль | Тест з об'єктно-орієнтованого програмування містить 15 завдань закритого/відкритого типу на знання і розуміння базових концепцій (абстрагування, інкапсуляція, успадкування, поліморфізм) об'єктно-орієнтованого програмування, на вміння створювати класи і об'єкти, працювати з атрибутами, на тестування програмного коду і обробку помилок. Максимальна оцінка за тест становить 15 балів. Контрольна робота містить одне комплексне завдання, що стосується написання програми в об'єктно-орієнтованому стилі для розв'язання практичної задачі. Максимальна оцінка за контрольну роботу становить 20 балів. |
| Самостійна робота | Максимальна оцінка за активну і змістовну аудиторну роботу та самостійну роботу за окремими темами становить 5 балів. |
| Підсумковий контроль | Залік. Підсумкова залікова оцінка є сумою оцінок за всі види робіт впродовж семестру. Максимальна оцінка становить 100 балів. |

Накопичування балів впродовж вивчення дисципліни

| Види навчальної роботи | Максимальна кількість балів |
|--------------------------|-----------------------------|
| Лабораторні роботи | 50 |
| Тест | 20 |
| Контрольна робота | 20 |
| Самостійна робота | 10 |
| Підсумкова оцінка | 100 |

ШКАЛА ОЦІНЮВАННЯ: НАЦІОНАЛЬНА ТА ECTS

| Сума балів за всі види навчальної діяльності | Оцінка ECTS | для екзамену, курсового проєкту (роботи), практики | для заліку | |
|--|--|--|------------|---|
| | | відмінно | зараховано | |
| 80 – 89 | добре | | | |
| 70 – 79 | | задовільно | | |
| 60 – 69 | незадовільно з можливістю повторного складання | | | не зараховано з можливістю повторного складання |
| 50 – 59 | | | | |
| 26 – 49 | FX | | | |
| 0-25 | F | | | |

КРИТЕРІЇ ОЦІНЮВАННЯ НАВЧАЛЬНИХ ДОСЯГНЕНЬ

(відповідно до [Положення](#) про організацію освітнього процесу)

«відмінно» – здобувач освіти міцно засвоїв теоретичний матеріал, глибоко і всебічно знає зміст навчальної дисципліни, основні положення рекомендованої літератури, логічно мислить і буде відповідь, вільно використовує набуті теоретичні знання при аналізі практичного матеріалу, висловлює своє ставлення до тих чи інших проблем, демонструє високий рівень засвоєння практичних навичок;

«добре» – здобувач освіти добре засвоїв теоретичний матеріал, володіє основними аспектами рекомендованої літератури, аргументовано викладає його; має практичні навички, висловлює свої міркування з приводу тих чи інших проблем, але припускається певних неточностей і похибок у логіці викладу теоретичного матеріалу або при аналізі практичного матеріалу;

«задовільно» – здобувач освіти в основному опанував теоретичними знаннями навчальної дисципліни, орієнтується в першоджерелах та рекомендованій літературі, але непереконливо відповідає, плутає поняття, додаткові питання викликають невпевненість або відсутність стабільних знань; відповідаючи на запитання практичного характеру, виявляє неточності у знаннях, не вміє оцінювати факти та явища, пов'язувати їх із майбутньою діяльністю;

«незадовільно» – здобувач освіти не опанував навчальний матеріал дисципліни, не знає наукових фактів, визначень, майже не орієнтується в рекомендованій літературі, відсутнє наукове мислення, практичні навички не сформовані.

7. ПОЛІТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

| | |
|--|--|
| Академічна доброчесність | Важливим є самостійне виконання навчальних завдань та посилання на джерела інформації у разі використання ідей, розробок, тверджень, відомостей. Плагіат та інші види академічної недоброчесності не принесуть користі, тому є недоречними та контролюються відповідно до Положення |
| Пропуски занять (відпрацювання) | Наслідком періодичних пропусків є самостійне опрацювання навчального матеріалу з можливим консультуванням у викладача відповідно до графіка консультацій. Наслідком систематичних пропусків є додаткові види самостійної роботи, які контролюються на передбачених у графіку навчального процесу тижнях контролю самостійної роботи |
| Виконання завдань пізніше встановленого терміну | Приводить до втрати частини балів, запланованих у системі оцінюванні навчальної дисципліни (усі види навчальної роботи важливо виконувати належним чином і вчасно, щоб зберігати загальний темп курсу, котрий сприяє ефективному засвоєнню матеріалу без шкоди здоров'ю |
| Додаткові бали | До 5 балів за активність, комунікативність, креативність, наполегливість, самостійність при вивченні дисципліни |
| Неформальна освіта | Можливість повного або часткового зарахування результатів неформальної освіти відповідно до Положення . Рекомендовані платформи: coursera.org , prometheus.org.ua , udemy.com , netacad.com , edx.org |

8. РЕСУРСНЕ ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

| | |
|--|---|
| Матеріально-технічне забезпечення | Лекційна аудиторія, мультимедіа, комп'ютерна лабораторія з програмним забезпеченням |
| Навчально-методичне забезпечення | Навчальний контент на освітніх платформах https://d-learn.pnu.edu.ua/PwP https://classroom.google.com/PwP |
| Інформаційне забезпечення | Наукова бібліотека КНУВС , Рекомендована література |

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Будаї А. Дизайн-патерни - просто як двері. – Ел. вид., 2012. – Режим доступу: [url](#)
2. Васильєв О.М. Програмування мовою Python. – Тернопіль: Навчальна книга – Богдан, 2019. – Режим доступу: [url](#)
3. Висоцька В.А., Оборська О.В. Python: алгоритмізація та програмування: навчальний посібник. – Львів: Новий Світ – 2000, 2021. – Режим доступу: [url](#)
4. Кренивч А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування: Навчальний посібник. – К.: ВПЦ "Київський університет", 2020. – Режим доступу: [url](#)

5. Мізюк О. Путівник мовою програмування Python. – Ел. вид., 2020. – Режим доступу: [url](#)
6. Швець О. Занурення в патерни проектування. – Ел. вид., 2021. – Режим доступу: [url](#)
7. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns. Elements of Reusable Object-Oriented Software. - Addison-Wesley Professional, 1994.
8. PEP 8 – Style Guide for Python Code. – Access mode: [url](#)
9. The Python Tutorial. - Access mode: [url](#)

Викладач В.Мазуренко