Міністерство освіти і науки України
Львівський національний університет природокористування
Факультет механіки, енергетики та інформаційних технологій
Кафедра інформаційних технологій

"ЗАТВЕРДЖУЮ"
Проректор з НПР
проф. Віталій БОЯРЧУК
_____
"____" _____ 20____ року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

(назва навчальної дисципліни)

спеціальність 122 «Комп'ютерні науки»
перший (бакалаврський) рівень вищої освіти

Львів 2024

Робоча програма навчальної дисципліни «Об'єктно-орієнтоване програмування» для здобувачів першого (бакалаврського) рівня вищої освіти ОП «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки»

Розробник: Татомир Андрій Володимирович, доцент, к.т.н.

Робочу програму схвалено на засіданні кафедри інформаційних технологій Протокол № 1 від 12 серпня 2024 року

Завідувач кафедри **Інформаційних технологій**

(підпис) (Тригуба А.М.)
(прізвище та ініціали)

Робочу програму схвалено на засіданні методичної комісії факультету механіки, енергетики та інформаційних технологій Протокол № 1 від 29 серпня 2024 року.

Голова методичної комісії факультету механіки, енергетики та інформаційних технологій.

(підпис) (Ковалишин С.Й.)
(прізвище та ініціали)

1. Опис навчальної дисципліни

Галузь знань, спеціальність, освітній ступінь

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 122 «Комп'ютерні науки»

Характеристика навчальної дисципліни: обов'язкова (цикл професійної підготовки)

Кількість кредитів – 9

Загальна кількість годин – 270

Індивідуальне науково-дослідне завдання – курсова робота

Вид контролю: іспит, іспит

Тижневих аудиторних годин для денної форми навчання (4 семестр) – 4

Тижневих аудиторних годин для денної форми навчання (5 семестр) – 4

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить (%):

для денної форми навчання – 62,7%;

для заочної форми навчання – 15,4%.

2. Програма навчальної дисципліни

Розділ 1. Об'єктно-орієнтоване програмування (частина 1)

Тема 1. Вступ

1. Мета, структура і предмет курсу.
2. Парадигми і мови програмування.
3. Передумови виникнення ООП. Історичний процес розвитку підходів до програмування. Структуроване програмування. Процедурне програмування. Складність коду. Концепція об'єктно-орієнтованого програмування.
4. Переваги та недоліки ООП. Альтернативи ООП. Продовження і варіанти реалізації ООП у різних мовах.

Тема 2. Системи контролю версій програмного забезпечення

1. Концепція версійности програмного коду. Відслідковування змін в коді. Пошук несправностей програмного забезпечення.
2. Система контролю версій Git. Основні команди та функціонал.
3. Розширене використання систем контролю версій. Робота з віддаленими репозиторіями. Колаборація.

Тема 3. Загальний огляд об'єктно-орієнтованої мови програмування Python. Запуск і робота з Python

1. Налаштування робочого середовища Python 3 для систем на базі Unix. Особливості роботи у Windows. Редактори коду та інтегровані середовища розробки.
2. Поняття компіляції та интерпретації. Динамічна типізація.
3. Змінні та оператори.

4. Базові та комплексні типи даних. Вбудовані типи даних у Python 3. Списки (Lists) та словники (Dictionaries). Розширення типів. Формат даних JSON.
5. Базові оператори для роботи з простими типами даних (математичні та логічні операції, робота зі стрічками). Приведення типів.

Тема 4. Структурне і процедурне програмування

1. Складні алгоритми та програми. Історичні витоки появи структурного та процедурного програмування.
2. Конструкції для підтримки структурованого програмування. Лінійне виконання. Оператори розгалуження. Цикли в Python 3. Переваги та недоліки структурованого програмування.
3. Процедурне програмування. Функції та процедури. Оголошення та виконання функцій. Аргументи і параметри функцій. Рекурсія.

Тема 5. Принцип інкапсуляції даних

1. Класифікація об'єктів. Стан об'єкта. Створення літералів об'єктів.
2. Взаємозв'язки між класами і об'єктами. Класи як типи даних.
3. Атрибути (властивості, поля) об'єктів. Змінні класу.
4. Оголошення методів. Створення об'єкта (class instance) за допомогою класу (конструктори і деструктори). Звертання класу до об'єкта через параметр "self". Методи об'єкта. Методи класу (параметр "cls"). Статичні методи.
5. Безпечне звертання до атрибутів. Концепція публічних, приватних та захищених полів об'єктів.

Тема 6. Наслідування в ООП

1. Взаємозв'язки між класами. Оголошення наслідування.
2. Перевикористання властивостей та методів (resolution order). Глобальний (вбудований) object.
3. Пряме та непряме наслідування. Функція super.
4. Множинне успадкування в Python.
5. Розуміння наслідування (функції help, isinstance, issubclass).
6. Формальне представлення наслідування. UML-діаграми класів.
7. Особливості прототипно-орієнтованих мов програмування.

Тема 7. Поліморфізм

1. Перевантаження методів. Способи організації поліморфізму.
2. "Магічні методи" в Python 3 та їх використання. Основні вбудовані оператори, що допускають перевизначення.
3. Декоратори. Getter'и, setter'и та deleter'и.

Тема 8. Модулі та пакети

1. Базове ядро Python 3.
2. Розширення функціональності за рахунок перевикористання стороннього коду. Імпорт пакетів. Роботи з PIP. Open source software.
3. Написання та підключення власних модулів.

Розділ 2. Об'єктно-орієнтоване програмування (частина 2)

Тема 9. Шаблони (патерни) проєктування програмного забезпечення

1. Поняття шаблону проєктування. Передумови та історія появи.
2. Основні групи шаблонів (твірні, структурні, поведінкові).
3. Використання шаблонів у виробництві.

Тема 10. Твірні шаблони проєктування

1. Основні твірні шаблони.
2. Розгляд базового прикладу.

Тема 11. Структурні шаблони проєктування.

1. Основні структурні шаблони.
2. Розгляд базового прикладу.

Тема 12. Поведінкові шаблони проєктування

1. Основні поведінкові шаблони.
2. Розгляд базового прикладу.

Тема 13. Принципи проєктування програмного забезпечення.

1. Практичні та теоретичні принципи проєктування. DRY, AHA, KISS тощо.
2. Комплекс принципів проєктування SOLID. Принцип єдиної відповідальності. Принцип відкритості/закритості. Принцип підстановки Лісков. Принцип розділення інтерфейсу. Принцип інверсії залежностей.

Тема 14. Якість коду. Рефакторинг

1. Поняття якості коду. Складові якості коду (читабельність, розширюваність, підтримуваність, testability тощо). “Чистий” та “брудний” код. ”Code smells”. Досвід розробників програмного забезпечення (developer experience).
2. Статична перевірка коду.
3. Тестування програмного забезпечення. Види тестування. Автоматичне тестування. Unit, integration та E2E тестування.
4. Процес рефакторингу. Тестування в процесі рефакторингу. Використання систем контролю версій. Техніки рефакторингу (перейменування, переміщення, спрощення або переписування тощо).

Тема 15. Переваги декларативного стилю програмування. Загальні відомості про функціональне програмування

1. Відмінності між імперативним та декларативним стилем написання програм. Когнітивна складність коду.
2. Загальні відомості про лямбда-числення.
3. Основні положення функціонального програмування.
4. Відмінності між ООП та ФП. Імутабельність. Композиція замість наслідування.
5. Використання фукціонального програмування в Python 3. Лямбда-функції.

Тема 16. Застосування ООП та інших парадигм програмування в реальному виробництві. Підведення підсумків

1. Сучасні підходи до розробки програмного забезпечення.
2. Мультипарадигменна розробка.

3. Структура навчальної дисципліни

Назви тем	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб.	інд.	с. р.		л	п	лаб.	інд.	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
	Рік підготовки 2 Семестр 4						Рік підготовки 2 Семестр 4					
Розділ 1. Об'єктно-орієнтоване програмування (частина 1)												
Тема 1.	8	2	-	2	-	4	8	2	-	-	-	6
Тема 2.	10	2	-	2	-	6	10	1	-	-	-	9
Тема 3.	12	2	-	6	-	4	12	1	2	-	-	9
Тема 4.	12	2	-	4	-	6	12	1	2	-	-	9
Тема 5.	12	2	-	4	-	6	12	1	-	-	-	11
Тема 6.	12	2		4		6	12	1	2			10
Тема 7.	12	2	-	5	-	5	12	1	2	-	-	10
Тема 8.	12	2	-	5	-	5	12	-	-	-	-	10
Іспит	30	-	-	-	-	30	30	-	-	-	-	30
Разом за семестр I	120	16	-	32		72	120	8	8	-	-	104
	Рік підготовки 3 Семестр 5						Рік підготовки 3 Семестр 5					
Розділ 2. Об'єктно-орієнтоване програмування (частина 2)												
Тема 9.	18	4	-	8	-	8	18	2	2	-	-	16
Тема 10.	10	4	-	2	-	4	10	1	1	-	-	8
Тема 11.	10	4	-	2	-	4	10	2	2	-	-	6
Тема 12.	12	4	-	4	-	4	12	1	1	-	-	10
Тема 13.	12	3	-	4	-	5	12	1	1	-	-	10

Тема 14.	14	3	-	4	-	5	14	1	1	-	-	10
Тема 15.	7	3	-	2	-	2	7	1	1	-	-	5
Тема 16.	7	3		2		2	7	1	1			5
Іспит	30	-	-	-	-	30	30	-	-	-	-	30
Разом за семестр II	120	28	-	28	-	64	120	10	10	-	-	100
К.Р.	30	-	-	-	-	30	30	-	-	-	-	30
Усього годин	270	44	-	60	-	166	270	18	18	-	-	234

4. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
	Розділ 1. Об'єктно-орієнтоване програмування (частина 1)	
1	Вступне заняття. Налаштування середовища розробки Python3 і запуск програм	2
2	Вивчення простих типів даних і методів роботи з ними у Python 3	2
3	Основи структурного програмування в Python 3	4
4	Основи процедурного програмування в Python 3	4
5	Створення та використання класів	4
6	Змінні класу та об’єкта	4
7	Використання методів класу і статичних методів	3
8	Наслідування в об’єктно-орієнтованому програмуванні	3
9	Поліморфізм в Python 3	3
10	Використання декораторів методів	3
	Разом розділ 1	32
	Розділ 2. Об'єктно-орієнтоване програмування (частина 2)	
11	Твірні шаблони проєктування	5
12	Структурні шаблони проєктування	5
13	Поведінкові шаблони проєктування	5
14	Принципи проєктування програмного забезпечення	4
15	Рефакторинг програмного забезпечення	4
16	Застосування ООП та інших парадигм програмування в реальному виробництві.	5
	Разом розділ 2	28

5. Теми, питання та завдання винесені на самостійне вивчення

№ з/п	Назва теми
1	Особливості і прийоми роботи з IDE PyCharm
2	PEP-8
3	Регулярні вирази
4	Бібліотека matplotlib
5	Бібліотеки numpy і scipy

6	Бібліотека Pandas
7	Побудова графіків візуалізації даних в Python за допомогой бібліотеки Matplotlib
8	Виведення результатів роботи у файли в Python.
9	Створення власних бібліотек
10	Написання тестів

6. Курсова робота

Мета курсової роботи полягає у закріпленні та розширенні теоретичних знань, отриманих під час вивчення дисципліни «Об'єктно-орієнтоване програмування», розвитку навичок самостійного проектування та розробки програм; оволодінні методами та інструментами розробки програмного забеспечення, вмінні застосовувати отримані знання для вирішення конкретних практичних завдань та розвитку вміння оформлення науково-дослідницької роботи.

Завдання курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» включають:

- вибір та обґрунтування теми дослідження;
- огляд та аналіз літератури з обраної теми;
- формулювання мети та завдань дослідження;
- вибір та застосування інструментів розробки для вирішення поставлених завдань;
- отримання та інтерпретація результатів дослідження;
- формулювання висновків та рекомендацій;
- оформлення курсової роботи відповідно до встановлених вимог.

Розподіл балів за виконання курсової роботи

Пояснювальна записка	Вибір методів та інструментів розробки для поставлених завдань	Інтерпретація отриманих результатів та формулювання обґрунтованих висновків	Презентація результатів дослідження та відповіді на запитання	Сума
до 20 балів	до 25 балів	до 35 балів	до 20 балів	100

7. Методи навчання

1. Словесні методи (лекція, пояснення).

2. Наочні методи

– ілюстрація (малюнки, таблиці, моделі тощо),

– демонстрування засобу демонстрування: навчальний фільм,

3. Практичні методи: комплексні проекти, практичні роботи, вправи.

8. Методи контролю

1. Усне опитування (фронтальне, індивідуальне).

2. Письмова аудиторна та поза аудиторна перевірка (підготовка різних відповідей, рефератів, контрольні роботи (з конкретних питань тощо).

3. Практична перевірка (виконання лабораторної роботи, виконання комплексного тематичного завдання).

Види контролю: Поточний контроль, проміжна та семестрова атестація.

9. Очікувані результати навчання з дисципліни

Набуття теоретичних та практичних знань про специфіку об’єктно-орієнтованої розробки додатків, а також формування у фахівців з інформаційних технологій знань та вмінь професійного використання принципів об’єктно орієнтовано програмування. Додатково передбачається вивчення основних принципів і особливостей інших сучасних парадигм програмування.

Навчальний контент

Формування програмних компетентностей

Індекс в матриці ОПП	Програмні компоненти
ІНТ	Здатність розв’язувати складні спеціалізовані задачі та практичні проблеми у галузі комп’ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов
ЗК2	Здатність застосовувати знання у практичних ситуаціях.
ЗК7	Здатність до пошуку, оброблення та аналізу інформації з різних джерел
СК3	Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв’язності та нерозв’язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.
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.
СК17	Здатність проектувати та реалізовувати програмне забезпечення для вирішення різноманітних прикладних задач інтелектуальної комп’ютеризації у сфері природокористування із використанням різних мов програмування, сучасних бібліотек крос-платформного програмування, проводити тестування на різних апаратних платформах, впроваджувати і підтримувати роботу інформаційних систем на сучасних платформах.
ПРН5	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.
ПРН 9	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.
ПРН 15	Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно- орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо- технічних систем.
ПРН19	Застосовувати та удосконалювати підходи до моделювання та оптимізації станів біологічних об’єктів та процесів природокористування, створювати та удосконалювати математичні моделі і програмні системи, а також використовувати сучасні бібліотеки та фреймворки для проектування і розробки інтелектуальних систем у сфері природокористування.

10. Розподіл балів, які отримують студенти

Розділ 1. Об'єктно-орієнтоване програмування (частина 1)

(4-й семестр)

Поточне тестування та самостійна робота (разом 50 балів)										Підсумковий контроль іспит	Сума
Т1	Т2	Т3	Т4	Т5	Т6	Т7	Т8	Т9	Т10		
5	5	5	5	5	5	5	5	5	5	50	100

Розділ 2. Об'єктно-орієнтоване програмування (частина 2)

(5-й семестр)

Поточне тестування та самостійна робота (разом 50 балів)						Підсумковий контроль іспит	Сума
Т11	Т12	Т13	Т14	Т15	Т16		
10	10	10	10	5	5	50	100

Т1, Т2 ... Т16 – теми.

11. Методичне забезпечення

Підручники і навчальні посібники; інструктивно-методичні матеріали до лабораторних занять; контрольні роботи; текстові та електронні варіанти тестів для поточного і підсумкового контролю, методичні матеріали для організації самостійної роботи студентів.

12. Рекомендована література

Базова

1. Креневич А.П. Python у прикладах і задачах. Частина 2. Об’єктно-орієнтоване програмування. Навчальний посібник – К.: ВПЦ "Київський Університет", 2020. – 152 с.
2. Креневич А.П., 2020 рік 2. Креневич А.П. Методичні вказівки до лабораторних занять із дисципліни "Об’єктно-орієнтоване програмування" для студентів механіко-математичного факультету – К.: ВПЦ "Київський Університет", 2019. с. 33.
3. The Python Tutorial [Електронний ресурс] – Режим доступу до ресурсу: https:// docs. python. org /3/ tutorial/ index .html .
4. Chun, Wesley. Core python applications programming / Wesley J. Chun. — 3rd ed. Pearson Education, Inc., 2012. – 852p.

5. Jason R. Briggs. Python for kids. A Playful Introduction to Programming - No Starch Press, Inc., San Francisco, CA 2013. – 318p

Допоміжна

6. M. Lutz: Learning Python: Powerful Object-Oriented Programming, 5th ed. // O'Reilly Media, Inc., 2013.
7. D. Beazley, B.K. Jones: Python Cookbook: Recipes for Mastering Python 3, 3rd ed. // O'Reilly Media, Inc., 2013.
8. A. Martelli, A. Ravenscro, S. Holden: Python in a Nutshell: The Definitive Reference, 3rd ed. // O'Reilly Media, Inc., 2017.
9. B. Lubanovic: Introducing Python: Modern Computing in Simple Packages. // O'Reilly Media, Inc., 2015.
10. J. Hunt: A Beginners Guide to Python 3 Programming. // Springer, 2019.
11. J. Hunt: Advanced Guide to Python 3 Programming. // Springer, 2019

13. Інформаційні ресурси

1. Віртуальне навчальне середовище ЛНУП - https://moodle.lnup.edu.ua/?redirect=0.
2. Бібліотечно-інформаційні ресурси - книжковий фонд, періодика та фонди на електронних носіях бібліотеки ЛНУП, державних органів науково-технічної інформації, наукових, науково-технічних бібліотек та інших наукових бібліотек України.

- Бібліотека Національного університету "Львівська політехніка" - 79013, Львів, вул. С. Бандери, 74;
- Бібліотека Інституту аграрної економіки НАН України - 01127, м. Київ, вул. Героїв Оборони, 10;
- Бібліотека Інституту регіональних досліджень НАН України - 70026, Львів, вул. Козельницька, 4;
- Бібліотека Львівського інституту менеджменту - 79601, Львів, пр. Чорновола, 57;
- Бібліотека Львівського національного університету ветеринарної медицини та біотехнологій ім. С.З. Гжицького - 79010, Львів, вул. Пекарська, 50;
- Бібліотека Львівської комерційної академії - 79034, Львів, вул. Туган-Барановського, 10;
- Бібліотека Національного університету біоресурсів і природокористування України - 01127, м. Київ, вул. Героїв Оборони, 15;
- Львівська наукова бібліотека імені В. Стефаника НАН України – м. Львів, вул. В.Стефаника,
- Національна бібліотека України імені В.І.Вернадського – м.Київ, пр. 50-річчя Жовтня, 4.

3. Електронні інформаційні ресурси мережі інтернет з переліком сайтів:

- http://www.cprogramming.com
- https://www.w3schools.com/default.asp
- https://developer.mozilla.org/en-US/docs/Web/HTML