

Міністерство освіти і науки України
Львівський національний університет ветеринарної медицини та біотехнологій
ім.С.З.Гжицького
Факультет механіки, енергетики та інформаційних технологій
Кафедра інформаційних технологій



ЗАТВЕРДЖЕНО

Гарант освітньо-професійної
програми «Комп'ютерні науки»
першого (бакалаврського) рівня
вищої освіти

к.т.н., доцент  В.В. Пташник

СИЛАБУС
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«Об'єктно-орієнтоване програмування»

освітньо-професійна програма «Комп'ютерні науки»
спеціальність 122 «Комп'ютерні науки»
перший (бакалаврський) рівень вищої освіти

ВИКЛАДАЧ

Татомир Андрій Володимирович



Електронна пошта:

andrew.tatomyr@gmail.com

Телефон

+380931128177

Доцент кафедри інформаційних технологій Львівського національного університету ветеринарної медицини та біотехнологій імені С.З. Гжицького, кандидат технічних наук, доцент. Викладач з 19-річним досвідом, має 44 публікації, з них 32 наукових та 12 навчально-методичного характеру, у тому числі наукові праці, опубліковані у вітчизняних і міжнародних рецензованих фахових виданнях.

Читає курси: Об'єктно-орієнтоване програмування, Програмування, Технологія розробки програмного забезпечення.

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 122 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

Рівень вищої освіти – перший (бакалаврський)

Кількість кредитів – 8

Рік підготовки, семестр – 2 рік, 4 семестр; 3 рік, 5 семестр

Компонент освітньої програми: обов'язкова

Мова викладання: українська

Опис дисципліни

Дисципліна “Об’єктно-орієнтоване програмування” вивчає способи організації програмного коду для полегшення його написання, відлагодження, підтримки та розширення. Під час вивчення дисципліни студенти отримують знання та вміння, спрямовані на якість написання коду відповідно до найкращих існуючих практик. В процесі навчання вони знайомляться з популярної об’єктно-орієнтованою мовою програмування Python, отримують знання про принципи об’єктно-орієнтованого програмування та вчать організовувати програмний код відповідно до існуючих виробничих стандартів.

Програма дисципліни «Об’єктно-орієнтоване програмування» відноситься до дисциплін професійної підготовки та складена відповідно до освітньо-професійної програми «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти.

Міждисциплінарні зв'язки: освітня компонента «Об’єктно-орієнтоване програмування» є складовою частиною циклу професійної підготовки для здобувачів освітньо-професійної програми «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти. Вивчення дисципліни передбачає наявність систематичних та ґрунтовних знань із суміжних курсів – «Вища математика», «Теорія ймовірності та математична статистика», «Алгоритми та структури даних» «Програмування».

Вимоги до знань та умінь визначаються галузевими стандартами вищої освіти України.

Предметом вивчення освітньої компоненти «Об’єктно-орієнтоване програмування» є процес навчання і підготовки фахівця за освітньо-професійною програмою «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти, який дозволить використовувати методи та засоби об’єктно-орієнтоване програмування, а також використання практичних інструментів для розробки програмного забезпечення.

Метою вивчення освітньої компоненти «Об’єктно-орієнтоване програмування» є набуття теоретичних та практичних знань про специфіку об’єктно-орієнтованої розробки додатків, а також формування у фахівців з інформаційних технологій знань та умінь професійного використання принципів об’єктно орієнтовано програмування. Додатково передбачається вивчення основних принципів і особливостей інших сучасних парадигм програмування

Основними завданнями освітньої компоненти «Об’єктно-орієнтоване програмування» є: підготовка фахівців, які здатні забезпечити самостійне розв’язування інженерних задач, пов’язаних із аналізом та проектуванням складних додатків, розвинення у студентів бачення переваг та недоліків застосування різних парадигм програмування, в першу чергу - об’єктно-орієнтованого програмування.

3
Структура курсу

Тема	Результати навчання	Завдання
Тема 1. Вступ	Мета, структура і предмет курсу. Парадигми і мови програмування. Передумови виникнення ООП.	Питання, лабораторна робота
Тема 2. Системи контролю версій програмного забезпечення	Концепція версійності програмного коду. Відслідковування змін в коді. Пошук несправностей програмного забезпечення. Система контролю версій Git. Основні команди та функціонал.	Питання, лабораторна робота
Тема 3. Загальний огляд об'єктно-орієнтованої мови програмування Python. Запуск і робота з Python	Налаштування робочого середовища Python 3 для систем на базі Unix. Особливості роботи у Windows. Редактори коду та інтегровані середовища розробки.	Питання, лабораторна робота
Тема 4. Структурне і процедурне програмування	Складні алгоритми та програми. Історичні витoki появи структурного та процедурного програмування.	Питання, лабораторна робота
Тема 5. Принцип інкапсуляції даних	Класифікація об'єктів. Стан об'єкта. Створення літералів об'єктів. Взаємозв'язки між класами і об'єктами. Класи як типи даних. Атрибути (властивості, поля) об'єктів. Змінні класу.	Питання, лабораторна робота
Тема 6. Наслідування в ООП	Взаємозв'язки між класами. Оголошення наслідування.	Питання, лабораторна робота
Тема 7. Поліморфізм	Перевантаження методів. Способи організації поліморфізму.	Питання, лабораторна робота
Тема 8. Модулі та пакети.	Базове ядро Python 3. Розширення функціональності за рахунок перевикористання стороннього коду. Імпорт пакетів. Роботи з PIP. Open source software.	Питання, лабораторна робота
Тема 9. Шаблони (патерни) проєктування програмного забезпечення	Поняття шаблону проєктування. Передумови та історія появи. Основні групи шаблонів (твірні, структурні, поведінкові).	Питання, лабораторна робота
Тема 10. Твірні шаблони проєктування	Основні творні шаблони. Розгляд базового прикладу.	Питання, лабораторна робота
Тема 11. Структурні шаблони проєктування	Основні структурні шаблони. Розгляд базового прикладу.	Питання, лабораторна робота
Тема 12. Поведінкові шаблони проєктування.	Основні поведінкові шаблони. Розгляд базового прикладу.	Питання, лабораторна робота
Тема 13. Принципи проєктування програмного забезпечення	Практичні та теоретичні принципи проєктування. DRY, АНА, KISS тощо. Комплекс принципів проєктування SOLID..	Питання, лабораторна робота

Тема 14. Якість коду. Рефакторинг	Поняття якості коду. Складові якості коду (читабельність, розширюваність, підтримуваність, testability тощо). “Чистий” та “брудний” код.	Питання, лабораторна робота
Тема 15. Переваги декларативного стилю програмування. Загальні відомості про функціональне програмування	Відмінності між імперативним та декларативним стилем написання програм. Когнітивна складність коду. Загальні відомості про лямбда-числення.	Питання, лабораторна робота
Тема 16. Застосування ООП та інших парадигм програмування в реальному виробництві. Підведення підсумків	Сучасні підходи до розробки програмного забезпечення. Мультипарадигменна розробка. Якість коду. Статична перевірка коду. Тестування програмного забезпечення. Види тестування. Автоматичне тестування. Unit, integration та E2E тестування	Питання, лабораторна робота

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.
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.
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 14	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.

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 забезпечення, 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

Літературні джерела

Базові

1. Крєневич А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Навчальний посібник – К.: ВПЦ "Київський Університет", 2020. – 152 с. ©
2. Крєневич А.П., 2020 рік 2. Крєневич А.П. Методичні вказівки до лабораторних занять із дисципліни "Об'єктно-орієнтоване програмування" для студентів механіко-математичного факультету – К.: ВПЦ "Київський Університет", 2019. – ____ с. 3.
3. The Python Tutorial [Електронний ресурс] – Режим доступу до ресурсу: [https:// docs. python. org /3/ tutorial/ index .html](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

Інформаційні ресурси в Інтернеті

1. Бібліотечно-інформаційні ресурси – книжковий фонд, періодика та фонди на електронних носіях бібліотеки ЛНУВМБ, державних органів науково-технічної інформації, наукових, науково-технічних бібліотек та інших наукових бібліотек України.

2. Віртуальне навчальне середовище ЛНУВМБ. URL: <https://moodle.lnup.edu.ua/>
3. Електронні інформаційні ресурси мережі інтернет:

- Прискорена гібридна хмарна платформа даних. URL: <https://www.weka.io/accelerated-cloud-data-platform/>
- JupyterLab. URL: <https://jupyter.org/>
- Evolutionary Complexity Research Group (EPlex): <http://eplex.cs.ucf.edu/>
- Матеріали відкритого курсу OpenDataScience [Електронний ресурс]. Електрон. дан. Режим доступу: World Wide Web. URL: <https://habr.com/ru/company/ods/blog/344044>.
- The latest in machine learning. Papers With Code [Електронний ресурс]. Електрон. дан. Режим доступу: World Wide Web. URL: <https://paperswithcode.com/>.
- Платформа для змагань з аналітики та передбачувального моделювання. [Електронний ресурс]. Режим доступу: <https://www.kaggle.com/>.
- Портал відкритих даних України. URL: <https://data.gov.ua/>
- Weka Machine learning software to solve data mining problems [Електронний ресурс]. URL: https://sourceforge.net/projects/weka/?source=typ_redirect.

КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ

У 4 семестрі успішність студента оцінюється шляхом проведення поточного контролю.

Максимальна кількість балів з дисципліни «Об'єктно-орієнтоване програмування», яку може отримати студент протягом семестру за всі види роботи за результатами поточного оцінювання становить 100. Результати **поточного контролю** оцінюються за чотирибальною («2», «3», «4», «5») шкалою. В кінці семестру обчислюється середнє арифметичне значення (САЗ) усіх отриманих студентом оцінок з наступним переведенням його у сто бальну шкалу за формулою: $ПК = 20 \cdot САЗ$.

Поточний контроль здобувачів освіти заочної форми навчання під час настановчої та лабораторно-екзаменаційної сесії оцінюється максимум у 30 балів. Ще 70 балів відведено на виконання тематичної самостійної роботи (ТСР) у міжсесійний період за програмою курсу. Підсумкова оцінка розраховується за формулою: $ПК + ТСР$.

В залікову відомість студентів у графі «за національною шкалою» виставляється оцінка «зараховано/незараховано» відповідно до таблиці 1. Присутність студента при виставленні підсумкової оцінки обов'язкова, якщо ним виконані усі передбачені види робіт.

У 5 семестрі успішність студента оцінюється шляхом проведення поточного та підсумкового контролю.

Максимальна кількість балів з дисципліни «Об'єктно-орієнтоване програмування», яку може отримати студент протягом семестру за всі види роботи, становить 100, при цьому 50 балів за результатами поточного оцінювання, та 50 – за результатами екзаменаційного контролю.

Результати **поточного контролю** оцінюються за чотирибальною («2», «3», «4», «5») шкалою. В кінці семестру обчислюється середнє арифметичне значення (САЗ) усіх отриманих студентом оцінок з наступним переведенням його у 50-ти бальну шкалу за формулою:

$$ПК = 10 \cdot САЗ$$

Критерії поточного оцінювання знань студентів

Оцінка	Критерії оцінювання
5 («відмінно»)	У повному обсязі володіє навчальним матеріалом, вільно, самостійно та аргументовано його викладає, глибоко і всебічно розкриває зміст, використовуючи обов'язкову та додаткову літературу. Правильно вирішив 90% тестових завдань.
4 («добре»)	Достатньо повно володіє навчальним матеріалом, обґрунтовано його викладає, в основному розкриває зміст завдань, використовуючи обов'язкову літературу. При викладанні окремих питань не вистачає достатньої глибини та аргументації, допускаються несуттєві неточності й незначні помилки. Правильно вирішив більшість тестових завдань.
3 («задовільно»)	У цілому володіє навчальним матеріалом, викладає його основний зміст, але без глибокого всебічного аналізу, обґрунтування та аргументації, допускаючи окремі суттєві неточності та помилки. Правильно вирішив близько половини тестових завдань.
2 («незадовільно»)	Не в повному обсязі володіє навчальним матеріалом. Викладає матеріал фрагментарно та поверхово, без аргументації й обґрунтування, недостатньо розкриває зміст теоретичних і практичних завдань, допускає суттєві неточності. Правильно вирішив меншість тестових завдань.

Переведення підсумкових рейтингових оцінок з дисципліни, виражених у балах за 100-бальною шкалою, у оцінки за національною шкалою та шкалою ECTS

Таблиця 1 – Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, диференційованого заліку, курсового проєкту (роботи), практики	для заліку
90–100	A	відмінно	зараховано
82–89	B	добре	
74–81	C		
64–73	D		
60–63	E	задовільно	
35–59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0–34	F	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

Максимальна кількість балів за курсову роботу (проєкт) становить 100. Критерії оцінювання курсових робіт (проєктів) представлено у таблиці 1.

Захист курсових робіт (проєктів) здійснюється перед комісією у складі 2-3 викладачів кафедри, у тому числі - керівника курсової роботи (проєкту).

До Силабусу також готуються матеріали навчально-методичного комплексу:

- 1) Навчальний контент (розширений план лекцій)
- 2) Тематика та зміст лабораторних робіт
- 3) Питання на іспит

4) Електронне навчання у системі MOODLE.