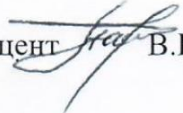


Міністерство освіти і науки України
Львівський національний університет природокористування
Факультет механіки, енергетики та інформаційних технологій
Кафедра інформаційних технологій



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 природокористування, 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, Інформаційна безпека, Мобільні додатки.

ЛЬВІВ 2024

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 122 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

Рівень вищої освіти – перший (бакалаврський)

Кількість кредитів – 9

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 популярною 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 «Інформаційні системи та технології» 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 «Інформаційні системи та технології» 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 «Інформаційні системи та технології» 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.

Структура курсу

Тема	Результати навчання	Завдання
Тема 1. Вступ	Мета, структура і предмет курсу. Парадигми і мови програмування. Передумови виникнення ООП.	Питання, практична 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, практична 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, практична робота
Тема 4. Структурне і процедурне програмування	Складні алгоритми та програми. Історичні витоки появи структурного та процедурного програмування.	Питання, практична 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, практична робота
Тема 6. Наслідування в ООП	Взаємозв'язки між класами. Оголошення наслідування.	Питання, практична робота
Тема 7. Поліморфізм	Перевантаження методів. Способи організації поліморфізму.	Питання, практична 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, практична робота
Тема 9. Шаблони (патерни) проектування програмного забезпечення	Поняття шаблону проектування. Передумови та історія появи. Основні групи шаблонів (твірні, структурні, поведінкові).	Питання, практична робота

Тема 10. Твірні шаблони проектування	Основні твірні шаблони. Розгляд базового прикладу.	Питання, практична робота
Тема 11. Структурні шаблони проектування	Основні структурні шаблони. Розгляд базового прикладу.	Питання, практична робота
Тема 12. Поведінкові шаблони проектування.	Основні поведінкові шаблони. Розгляд базового прикладу.	Питання, практична робота
Тема 13. Принципи проектування програмного забезпечення	Практичні та теоретичні принципи проектування. DRY, АНА, KISS тощо. Комплекс принципів проектування SOLID..	Питання, практична робота
Тема 14. Якість коду. Рефакторинг	Поняття якості коду. Складові якості коду (читабельність, розширюваність, підтримуваність, testability тощо). “Чистий” та “брудний” код.	Питання, практична 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, практична 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, практична 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.
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.
СК16	Здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації.
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.

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 ЛНУП, державних органів науково-технічної інформації, наукових, науково-технічних бібліотек та інших наукових бібліотек України.
2. Віртуальне навчальне середовище ЛНУП. 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.

Політика оцінювання

Політика щодо дедлайнів та перескладання: Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (75% від можливої максимальної кількості балів за вид діяльності балів). Перескладання модулів відбувається за наявності поважних причин (наприклад, лікарняний).

Політика щодо академічної доброчесності: Списування під час контрольних робіт заборонені (в т.ч. із використанням мобільних девайсів). Мобільні пристрої дозволяється використовувати лише під час он-лайн тестування та підготовки практичних завдань під час заняття.

Політика щодо відвідування: Відвідування занять є обов'язковим компонентом оцінювання. За об'єктивних причин (наприклад, хвороба, працевлаштування, міжнародне стажування) навчання може відбуватись в он-лайн формі за погодженням із ведучим викладачем курсу.

Оцінювання

Остаточна оцінка за курс розраховується наступним чином: поточний контроль оцінюється в 50 балів, та складається із двох модулів по 25 балів кожен. В суму балів кожного модуля входять бали за підготовку, виконання та захисту практичних.

П1, П2 ... П16 – практичні роботи; СР – самостійна робота.

До Силабусу також готуються матеріали навчально-методичного комплексу:

- 1) Навчальний контент (розширений план лекцій);
- 2) Тематика та зміст практичних робіт;
- 3) Тематика та методичні рекомендації до виконання курсової роботи;
- 4) Завдання для підсумкової роботи, питання на іспит;
- 5) Електронне навчання у віртуальному навчальному середовищі ЛНУП (<https://moodle.lnup.edu.ua/>).