

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра радіоелектронних і комп'ютерних систем

Затверджено

На засіданні кафедри РКС
факультету електроніки та комп'ютерних
технологій
Львівського національного університету
імені Івана Франка
(протокол № ____ від 31 серпня 2020 р.)

В.о. завідувача кафедри

Силабус з навчальної дисципліни
«Професійний Python»,
що викладається в межах ОПП «Інформаційні системи та
технології» першого (бакалаврського) рівня вищої освіти для
здобувачів з спеціальності
126 «Інформаційні системи та технології»

Назва дисципліни	Професійний Python
Адреса викладання дисципліни	м. Львів, вул. Драгоманова, 50
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій, кафедра радіоелектронних і комп'ютерних систем
Галузь знань, шифр та назва спеціальності	12 Інформаційні технології, 126 Інформаційні системи та технології
Викладачі дисципліни	Сінькевич Олег Олександрович, асистент
Контактна інформація викладачів	oleh.sinkevych@lnu.edu.ua, https://electronics.lnu.edu.ua/employee/o_sinkevych
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекційних занять (за попередньою домовленістю). Також можливі он-лайн консультації через MS Teams та Telegram (у текстовому режимі). Для погодження часу онлайн консультацій слід писати на електронну пошту викладача, або в особистий Telegram.
Сторінка дисципліни	https://electronics.lnu.edu.ua/course/mova-prohramuvannia-python-ist-2
Інформація про дисципліну	Дисципліна «Професійний Python» є вибірковою дисципліною з спеціальності 126 Інформаційні системи та технології для освітньої програми «Інформаційні системи та технології», яка викладається в 6 семестрі в обсязі 6.0 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Навчальну дисципліну розроблено таким чином, щоб надати студентам необхідні знання на практичні навички, обов'язкові для того, щоб вміти реалізовувати та застосовувати типові алгоритми та відповідні їм структури даних для вирішення алгоритмічних завдань та проходження технічних співбесід. Тому у дисципліні представлено як теоретичні основи властивих для мови Python структур даних та ООП підходів, так і конкретні реалізації та бібліотеки, які необхідні для створення скриптів та прикладних програмних інтерфейсів (API). Зокрема, детально розглянуто кроки для реалізації складних структур даних, алгоритмів для роботи з ними, а також типові мікрофреймворки Flask/Fastapi для швидкого створення веб-орієнтованих API.
Мета та цілі дисципліни	Метою вивчення вибіркової дисципліни «Професійний Python» є ознайомлення студентів з основами алгоритмів та структур даних Python, оволодіння алгоритмічними навичками з урахуванням особливостей мови Python, сучасними підходами створення API та їх застосування в межах типових мікрофреймворків.
Література для вивчення дисципліни	Основна література: 1. Род Стивенс. Алгоритмы. Теория и практическое применение. Wiley.-2013. -543 с. 2. Марк Лутц. Изучаем Python, 5-е издание. Символ-Плюс, O'Reilly.-2013. -1700 с. 3. Luciano Ramalho. Fluent Python: Clear, Concise, and Effective Programming. O'Reilly Media; 1 edition. - 2015. -792 с. 4. Бхаргава А. Грокаем Алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. Питер. - 2017. - 290 с. Додаткова література: 5. Phillips D. Python 3 Object-Oriented Programming: Build robust and maintainable software with object-oriented design patterns in Python 3.8, 3rd Edition // Pacts Publishing. - 2018. - 466 с.

	<p>6. Кормен Т., Лейзерсон Ч., Рівест Р., Стайн К. Вступ до алгоритмів. К. : К. І. С., 2019. — 1288 с.</p> <p>7. Real Python Tutorials [Електронний ресурс] – Режим доступу до ресурсу: https://realpython.com/.</p> <p>8. М. Гринберг. Разработка веб-приложений с использованием Flask на языке Python. ДМК Пресс. - 272 с.</p> <p>9. Kunal Relan. Building REST APIs with Flask // Apress Media, LLC. - 2019. - 199 с.</p> <p>10. Narasimha Karumanchi. Data Structures And Algorithms Made Easy // CareerMonk Publications. - 2017. - 828 с.</p> <p>11. Gabriele Lanaro, Quan Nguyen, Sakis Kasampalis. Advanced Python Programming: Build high performance, concurrent, and multi-threaded apps with Python using proven design patterns // Packer Publishing. - 2019. - 672 с.</p>
Обсяг курсу	64 години аудиторних занять. З них 32 години лекцій, 32 години лабораторних робіт та 116 годин самостійної роботи.
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <ul style="list-style-type: none"> - Знати процес написання скриптів; створення добре структурованих програм на мові Python з використанням ООП; основи модульного тестування розроблених класів; типові структури даних та алгоритми з їх використанням; теоретичний та практичний матеріал згідно програми курсу: типи даних, оператори, умови, цикли, функції, об'єкти, методи і т. п. - Вміти застосовувати засвоєний матеріал для програмної реалізації алгоритмічних задач; створювати та використовувати об'єктно-орієнтований код для вирішення типових задач, а також реалізувати складні структури даних та пов'язані алгоритми.
Ключові слова	Python, Flask, Fastapi, PyMongo, classes, linked lists, decision trees.
Формат курсу	Очний
	Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем.
Теми	Див. СХЕМА КУРСУ
Підсумковий контроль, форма	Залік в кінці семестру
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін «Алгоритми та структури даних», «Основи програмування».
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентація, лекції, лабораторні роботи, обговорення, дискусія.
Необхідне обладнання	Мультимедіа, платформа Moodle, комп'ютерне програмне забезпечення.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	<p>Оцінювання проводиться упродовж семестру за 100-бальною шкалою. Бали нараховуються за такими видами робіт з наступним співвідношенням:</p> <ul style="list-style-type: none"> • лабораторні роботи: 60% семестрової оцінки; максимальна кількість балів 60. • контрольні заміри (2 модулі): 30% семестрової оцінки; максимальна кількість балів 30. • відвідування занять: 10% семестрової оцінки; максимальна кількість балів 10. <p>Загалом упродовж семестру 100 балів.</p>
	Контрольні заміри проводяться у формі тестових завдань.

	<p>Академічна доброчесність: Очікується, що лабораторні та контрольні роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикавання джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p> <p>Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Студенти мають інформувати викладача про неможливість відвідати заняття. Студенти зобов'язані дотримуватися усіх термінів визначених для виконання усіх видів робіт, передбачених курсом.</p> <p>Література. Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.</p> <p>Політика виставлення балів. Враховуються бали набрані на поточному тестуванні та самостійній роботі. При цьому обов'язково враховуються присутність на заняттях та активність студента під час лабораторного заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання.</p> <p>Жодні форми порушення академічної доброчесності не толеруються.</p>
Питання до контрольних робіт	Перелік питань та завдань для проведення підсумкової оцінки знань певних тем до контрольних робіт розміщені на веб-сторінці.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

СХЕМА КУРСУ

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в Інтернеті	Завдання (лабораторна робота), год	Термін виконання
1	<p>Мислити як computer scientist. Динамічна типізація. Класи та ООП. Особливості ООП у Python 3: інкапсуляція, наслідування, поліморфізм, композиція, абстракція. Методи класів, їх типи.</p> <p>Важливість алгоритмічного мислення в контексті Python-розробки. Динамічна типізація. Об'єкти та класи. Взаємозв'язки між класами: асоціація, агрегація, композиція. Парадигми ООП. Специфіка інкапсуляції у Python. Методи класів. Статичний і метод класу.</p>	Лекція	1, 2, 4, 7	Реалізація класів шахових фігур, 3 год.	2 тиж. семестру
2	<p>Ієрархія абстракцій. Абстрактний клас. Структура Python-проектів. ООП реалізації зв'язних списків (ЗС) (алгоритми). ЗС для об'єктів. Сортування зв'язних списків.</p>	Лекція	1, 2, 3, 6, 7	Реалізації одно- та двозв'язних списків; алгоритми роботи, 4 год.	3 тиж. семестру

	Реалізація абстрактного класу. ABC та @abstractmethod. Приклад оформлення структури проекту. Однозв'язний список, його реалізація та алгоритми роботи з ним. Сортування однозв'язного списку. Двозв'язний список, реалізація, алгоритми роботи та сортування. Переваги та недоліки, а також важливість застосування.				
3	Масиви (списки). ООП реалізації. Контейнери об'єктів. Нерекурсивні алгоритми впорядкування об'єктів у масивах. Основи виключень. Об'єктно-орієнтована реалізація масиву (списку) для об'єктів. Контейнеризація об'єктів у структурі масиву. Сортування об'єктів за атрибутами вставкою та вибором. Робота з виключеннями, перехоплення помилок та власна реалізація класу для перехоплення помилок.	Лекція	1, 2, 4, 7, 10	Реалізація контейнеру для шахових фігур, його впорядкування, 2 год.	4 тиж. семестру
4	Словники та хеші. Робота з JSON. Серіалізація об'єктів. Основи NoSQL. Інтеграція з MongoDB. Singleton патерн. Використання логування. Хеш-функція та хеш-таблиця. Вбудовані структури даних: словники. Робота з json-форматом та основи серіалізації об'єктів класів. Основи NoSQL в контексті Python. Збереження стану об'єктів у базі даних MongoDB . Реалізація патерну Одинак для створення під'єднання до бази даних. Бібліотека logging та різновиди логування.	Лекція	1, 2, 3, 7, 9	Серіалізація об'єктів, робота з json та базою даних MongoDB, 3 год.	5 тиж. семестру
5	Пошукові алгоритми об'єктів. Бінарний, лінійний та інтерполяційний пошуки. Основи unit-тестування. Пошук об'єктів у масивах. Реалізація бінарного, лінійного та інтерполяційного пошуку, їх особливості, переваги та недоліки. Основи модульного тестування, бібліотеки unittest та pytest. Тестування власних класів та правила написання хорошого коду.	Лекція	1, 2, 7		
6	Стеки та черги об'єктів. ООП реалізація. Циклічна черга. Стек і ЗС. Об'єктно-орієнтована реалізація стеку та черги для інших об'єктів. Реалізація циклічної черги об'єктів. Побудова складених алгоритмів на основі зв'язних списків та стеку/черги.	Лекція	1, 2, 3, 5, 6, 7	ООП реалізація стеку та черги об'єктів, циклічна черга, 4 год.	7 тиж. семестру
7	Об'єкти і рекурсія. Стек	Лекція	1, 2, 4, 6		

	<p>викликів. Рекурсивні алгоритми сортування об'єктів. Рекурсивний пошук. Передача аргументів між рекурсивними викликами.</p> <p>Стек викликів та рекурсивний підхід. Реалізація простих рекурсивних алгоритмів. Передача аргументів функцій у рекурсивні виклики. Рекурсивні алгоритми сортування об'єктів: сортування злиттям, швидке сортування. Клас-сортувальник. Бінарний рекурсивний пошук.</p>				
8	<p>Об'єкти та графи. ООП реалізації. Пошуки по графу: вглиб та вшир. Приклад API з інтеграцією Neo4j.</p> <p>Важливість алгоритмів на графах з точки зору Python-розробника. Об'єктно-орієнтована реалізація класу графа. Типові алгоритми роботи з графами: пошук вшир та вглиб. Їх реалізація. Графові бази даних: OrientDB, Neo4j. Інтеграція Python-коду з базою даних Neo4j.</p>	Лекція	1, 2, 3, 5, 7	ООП реалізація графа, алгоритми пошуку, 4 год.	9 тиж. семестру
9	<p>Дерева та об'єкти. ООП реалізації простого дерева. Бінарне дерево та пошук об'єктів. Основи В-дерева.</p> <p>Об'єктно-орієнтована реалізація класу дерева. Особливості бінарного дерева та реалізація пошуку об'єктів у структурі даних на основі бінарних дерев. В-дерева та їх особливості для Python-розробника.</p>	Лекція	1, 2, 3, 5, 7		
10	<p>Дерево прийняття рішень. ООП реалізація. Основи CART алгоритму. Застосування у машинному навчанні.</p> <p>Об'єктно-орієнтована реалізація дерева прийняття рішень. Алгоритм CART. Простий алгоритм класифікації на базі розробленого дерева рішень. Приклад з машинного навчання. Переваги та недоліки таких алгоритмів. Алгоритм випадкового лісу.</p>	Лекція	1, 2, 3, 7, 10	ООП реалізація простого дерева рішень, підготовка даних, 4 год.	12 тиж. семестру
11	<p>Потоки та процеси у Python. Асинхронні потоки. Бібліотеки threading та multiprocessing. GIL.</p> <p>Особливості роботи з потоками у Python. Модуль threading та його застосування з власними</p>	Лекція	3, 11		

	<p>класами. GIL та проблема багатопоточності. Модуль multiprocessing. Основи асинхронних потоків.</p>				
12	<p>MVT архітектура. Створення простого Restful API. Мікрофреймворки Flask, Fastapi.</p> <p>Структурування проекту. Концепція model-view-template архітектури. Основи http-методів та REST парадигма. Restful api з використанням Flask та Fastapi. Створення простої Rest API для огортання розроблених класів.</p>	Лекція	7, 8, 9	Створення простого Restful api з використанням Flask/Fastapi на основі розроблених шахових класів, 4 год.	15 тиж. семестру
13	<p>Основи патернів проєктування. Типи патернів. Декорування класів та методів.</p> <p>Патерни проєктування. Структурні, породжувальні та поведінкові. Нативні патерни Python. Патерн Декоратор: декорування функцій та класів. Приклади декорування на базі мікрофреймворку Flask.</p>	Лекція	3, 7, 8		
14, 15	<p>Інтеграції коду і модулів. Система контролю версій GIT. Основи рефакторингу коду. Python - мова AI.</p> <p>Python модулі та правила створення власних пакетів. Налаштування системи контролю версій Git. Структурування проекту. Деякі засадничі правила рефакторингу коду Python. Python та AI фреймворки.</p>	Лекція	3, 7, 8, 9	Реалізація патерна Декоратор для шахових класів. Інтеграція з базою даних MongoDB та системою Gitlab, 4 год.	16 тиж. семестру
16	<p>Підсумкове заняття. Підведення підсумків. Настановчі питання.</p>	Лекція	2, 3		