



Технології програмування на мові Python

Робоча програма кредитного модуля (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення 123 Комп'ютерна інженерія 126 Інформаційні системи та технології
Освітня програма	
Статус дисципліни	Вибіркова
Форма навчання	Очна (денна)
Рік підготовки, семестр	3 курс, осінній семестр
Обсяг дисципліни	4 кредити, 120 годин 54 години аудиторної роботи, 66 годин самостійної роботи
Семестровий контроль, контрольні заходи	Залік
Розклад занять	Згідно розкладу на осінній семестр поточного навчального року за посиланням schedule.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу, викладачів	Асистент кафедри обчислювальної техніки, доктор філософії Шульга Максим Володимирович, shulha.maksym@iit.kpi.ua
Розміщення курсу	Онлайн лекції: https://bbb.comsys.kpi.ua/rooms/yfg-txk-udo-fbw Методичні матеріали: https://classroom.google.com/c/Nzc0OTU1Mjg1ODk5?cjc=lxayc5yr

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Метою навчальної дисципліни є надання студентам всебічної підготовки з технологій програмування на Python та їх застосування у сучасній інженерії програмного забезпечення. Це включає розробку додатків з використанням фреймворків Python, розгортання рішень за допомогою контейнеризації та хмарних обчислень, проектування мікросервісів і безсерверних архітектур, а також забезпечення безпеки та моніторингу додатків.

Предметом навчальної дисципліни є технології програмування на мові Python, що надають студентам розуміння Python, що включає фреймворки, контейнеризацію, хмарні обчислення, безпеку та моніторинг.

Завданням вивчення навчальної дисципліни є:

- ефективно використовувати Python для розробки додатків, включаючи веб-проекти з Flask та Django;
- застосовувати технології контейнеризації, такі як Docker, Kubernetes та OpenShift, для оптимізації процесів розгортання;

- проєктувати та реалізовувати мікросервіси й безсерверні архітектури для створення масштабованих і гнучких додатків;
- розгортати додатки на Python у хмарних середовищах;
- інтегрувати можливості генеративного штучного інтелекту у Python-додатки;
- розуміти та впроваджувати заходи безпеки для захисту додатків протягом усього процесу розробки;
- застосовувати інструменти моніторингу, журналювання та спостереження для забезпечення надійності та продуктивності додатків.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

При вивченні цієї дисципліни використовуються знання студентів з дисциплін: “Програмування”, “Вступ до операційної системи Linux”, “Організація баз даних”, “Інженерія програмного забезпечення”.

3. Зміст навчальної дисципліни

Розділ 1. Технології програмування на Python

Тема 1.1. Знайомство з Python

Тема 1.2. Розробка додатків з Flask

Тема 1.3. Розробка додатків з Django і базами даних

Тема 1.4. Робота з контейнерами: Docker, Kubernetes і OpenShift

Тема 1.5. Розробка додатків на основі мікросервісів і безсерверних архітектур

Тема 1.6. Хмарні обчислення для додатків на Python

Тема 1.7. Розробка додатків на основі генеративного штучного інтелекту з використанням Python

Розділ 2. Безпека та моніторинг додатків

Тема 2.1. Основи безпеки додатків на Python

Тема 2.2. Ризики безпеки додатків та найкращі практики

Тема 2.3. Моніторинг додатків

Тема 2.4. Журналювання та спостереження

4. Навчальні матеріали та ресурси

Базові ресурси:

1. Pierre Bourque and Richard Fairley: Guide to the Software Engineering Body of Knowledge, Version 3.0 SW. – IEEE Computer Society, 2014. – 346 p.
2. David Beazley, Brian Jones: Python Cookbook, 3rd ed. – O'Reilly Media, 2013. – 704 p.
3. Ian Sommerville: Software Engineering, 10th ed. – Addison-Wesley, 2016. – 790 p.
4. Karl Wieggers, Joy Beatty: Software Requirements, 3rd ed. – Microsoft Press, 2013. – 672 p.
5. Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Paulo Merson, Robert Nord, Judith Stafford: Documenting Software Architectures: Views and Beyond, 2nd ed. – Pearson Education, 2010. – 592 p.
6. David Budgen: Software Design, 2nd ed. – Addison-Wesley, 2003. – 400 p.
7. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, 3rd ed. – Addison-Wesley Prof, 2012. – 624 p.

Додаткові ресурси:

1. Zed Shaw: Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code. – Zed Shaw's Hard Way, 2017. – 320 p.
2. John Dooley: Software Development and Professional Practice. – Apress, 2011. – 260 p.
3. Robert C. Martin: Clean Code: A Handbook of Agile Software Craftsmanship. – Pearson, 2008. – 464 p.

Обладнання, що необхідне для проведення занять

Лекційні заняття проводяться в аудиторії, яка обладнана проектором, лабораторні заняття – в комп'ютерній лабораторії. Для дистанційних занять потрібен доступ до Інтернету та програмне й апаратне забезпечення для проведення відеоконференцій.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Структура навчальної дисципліни

Назви розділів, тем	Розподіл навчального часу				
	Всього	в тому числі			
		Лекції	Практичні (семінарські) заняття	Лабораторні роботи (комп'ютерний практикум)	CPC
1	2	3	4	5	6
Розділ 1. Технології програмування на Python					
Тема 1.1. Знайомство з Python	8	2			6
Тема 1.2. Розробка додатків з Flask	12	2		4	6
Тема 1.3. Розробка додатків з Django і базами даних	14	4		4	6
Тема 1.4. Робота з контейнерами: Docker, Kubernetes і OpenShift	14	4		4	6
Тема 1.5. Розробка додатків на основі мікросервісів і безсерверних архітектур	14	4		4	6
Тема 1.6. Хмарні обчислення для додатків на Python	12	4		2	6
Тема 1.7. Розробка додатків на основі генеративного штучного інтелекту з використанням Python	8	2			6
Разом за розділом 1	82	22	0	18	42
Розділ 2. Безпека та моніторинг додатків					
Тема 2.1. Основи безпеки додатків на Python	4	2			2

1	2	3	4	5	6
Тема 2.2. Ризики безпеки додатків та найкращі практики	4	2			2
Тема 2.3. Моніторинг додатків	4	2			2
Тема 2.4. Журналювання та спостереження	4	2			2
Разом за розділом 2	16	8	0	0	8
Модульна контрольна робота	10				10
Залік	12	6			6
Разом за семестр	120	36	0	18	66

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на літературу та завдання на СРС)
1	2
1	Тема 1.1. Знайомство з Python Основи Python, включаючи типи даних, вирази, змінні та структури даних, застосування логіки програмування Python за допомогою розгалуження, циклів, функцій, об'єктів і класів, використання бібліотек Python, доступ до веб-даних за допомогою API та веб-скрапінгу з Python.
2	Тема 1.2. Розробка додатків з Flask Кроки та процеси, пов'язані зі створенням додатка Python, створення модулів Python, запуск модульних тестів та пакування додатків, функції Flask і розгортання додатків в Інтернеті за допомогою фреймворку Flask.
3 4	Тема 1.3. Розробка додатків з Django і базами даних Бази даних, створення моделі даних зв'язків між сутностями для реляційної бази даних, створення запитів SQL для вставки, вибору, оновлення та видалення даних у базі даних, використання Django ORM для створення об'єктно-орієнтованих баз даних.
5 6	Тема 1.4. Робота з контейнерами: Docker, Kubernetes і OpenShift Використання контейнерів, переміщення додатків в будь-якому середовищі, створення власних хмарних додатків за допомогою Docker, Kubernetes, OpenShift і Istio, опис та використання архітектури Kubernetes, щоб налаштувати та використовувати систему управління контейнерами на основі всього життєвого циклу.
7 8	Тема 1.5. Розробка додатків на основі мікросервісів і безсерверних архітектур Основи мікросервісів, їхні переваги та контраст з монолітними архітектурами, створення кінцевих точок REST API та їх виклик за допомогою cURL і Postman, використання SwaggerUI для документування та тестування API, створення та розгортання мікросервісів за допомогою контейнерів Docker та безсерверних технологій, як-от IBM Code Engine.
9 10	Тема 1.6. Хмарні обчислення для додатків на Python Основи хмарних обчислень, включаючи їх визначення, моделі обслуговування (IaaS, PaaS, SaaS), моделі розгортання та ключові компоненти інфраструктури, нові тенденції, такі як HybridMulticloud, Microservices, Serverless, Cloud Native, DevOps і Application Modernization, сервіси від основних хмарних платформ, такі як AWS, Azure, Google Cloud, IBM Cloud і Alibaba Cloud.

1	2
11	Тема 1.7. Розробка додатків на основі генеративного штучного інтелекту з використанням Python Створення генеративних додатків на базі штучного інтелекту для різноманітних випадків використання, робота над практичними проєктами зі створення чат-ботів та додатків з використанням різних генеративних моделей та технологій штучного інтелекту.
12	Тема 2.1. Основи безпеки додатків на Python Як безпека вписується в робочий процес, практичні знання про концепції та термінологію безпеки, ключові стратегії пом'якшення ризиків для захисту додатка протягом розробки та впровадження.
13	Тема 2.2. Ризики безпеки додатків та найкращі практики Відкритий проєкт забезпечення безпеки веб-додатків (OWASP) та його 10 найголовніших проблем безпеки, найкращі практики написання коду та залежності програмного забезпечення.
14	Тема 2.3. Моніторинг додатків Моніторинг додатків, поширені терміни, що використовуються в моніторингу, та чому моніторинг важливий для розробників, синтетичний моніторинг та його важливість.
15	Тема 2.4. Журналювання та спостереження Концепція ведення журналу додатків та її важливість, спостереження, його переваги та три основи спостереження.

Лабораторні заняття

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1	2	3
1	Розробка додатків з Flask (Розділ 1, Тема 1.2)	4
2	Розробка додатків з Django і базами даних (Розділ 1, Тема 1.3)	4
3	Робота з контейнерами: Docker, Kubernetes і OpenShift (Розділ 1, Тема 1.4)	4
4	Розробка додатків на основі мікросервісів і безсерверних архітектур (Розділ 1, Тема 1.5)	4
5	Хмарні обчислення для додатків на Python (Розділ 1, Тема 1.6)	2
	Разом:	18

6. Самостійна робота студента

У процесі виконання індивідуальних завдань студенти повинні закріпити знання, отримані під час лекційних занять та лабораторних робіт, самостійно вивчати визначені теми, поглиблювати свої знання для подальшого навчання. Самостійна робота студентів полягає в наступному:

- підготовці до лекційного заняття шляхом вивчення матеріалу, що розглядався на попередньому лекційному занятті;

- підготовці до лабораторних занять шляхом виконання лабораторних робіт з вивченням теоретичних питань, які розглядаються на лекційних заняттях;
- оформленні протоколу про виконання лабораторних робіт;
- підготовці до залікової роботи.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Всі студенти повинні відвідувати лекційні та лабораторні заняття – як при очному навчанні (фізичне відвідування занять в аудиторіях), так і при дистанційному навчанні (віртуальне відвідування онлайн-занять).

Під час занять студенти повинні дотримуватись певних дисциплінарних правил:

- забороняється запізнюватись на заняття;
- не допускаються сторонні розмови або інший шум, що заважає проведенню занять;
- виходити з аудиторії під час заняття допускається лише з дозволу викладача.
- не допускається користування мобільними телефонами та іншими технічними засобами без дозволу викладача;
- не допускається займатися діяльністю, яка прямо не стосується навчальної дисципліни.

Результати виконання лабораторних робіт повинні бути оформлені у електронному форматі у вигляді файлів звітів. Такі файли повинні містити результати виконання відповідних завдань та відповідати вимогам та методичним вказівкам для кожної роботи.

Роботи, які здаються із порушенням термінів, оцінюються на нижчу оцінку.

Лабораторні роботи та модульна контрольна робота перевіряються на наявність плагіату. Наявність плагіату призводить до незадовільної оцінки роботи без можливості перездачі роботи.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Підсумкова рейтингова оцінка (R) студента складається з балів, які він отримує:

- за лабораторні роботи ($R_{\text{ЛАБ}}$);
- за тести (R_T);
- за модульну контрольну роботу ($R_{\text{МКР}}$);
- за залікову роботу (R_3).

Відповідно до ["Положення про систему оцінювання результатів навчання в КПІ ім. Ігоря Сікорського"](#), затвердженого Наказом №1/273 від 14.09.2020 р., застосована система оцінювання типу PCO-1, для якої рейтингова оцінка складається з:

- стартової оцінки (R_C) – оцінювання заходів впродовж семестру: $R_C = R_{\text{ЛАБ}} + R_T + R_{\text{МКР}}$;
- оцінки за залікову роботу (R_3).

Таким чином, $R = R_C + R_3$.

Студенти, які мають стартову оцінку (R_C) не меншу за деяке значення мінімально можливої загальної позитивної оцінки ($R_{\text{МИН}}$) та згодні з тим, що їх стартова оцінка (R_C) дорівнюватиме підсумковій рейтинговій оцінці (R), звільняються від написання залікової роботи.

Студенти, які мають стартову оцінку (R_C) меншу за деяке значення мінімально можливої загальної позитивної оцінки ($R_{\text{МИН}}$) або бажають виконати залікову роботу, отримують

стартову оцінку (R_C), що дорівнює оцінці за лабораторні роботи та модульну контрольну роботу ($R_{\text{ЛАБ}} + R_{\text{МКР}}$).

Протягом курсу студенти виконують 5 (п'ять) лабораторних робіт, 8 (вісім) тестів та модульну контрольну роботу. Тест вважається контрольним заходом.

Розрахунок шкал оцінювання

Шкала оцінювання (максимально можлива оцінка) лабораторної роботи: 10 балів.

Шкала оцінювання всіх лабораторних робіт: $R_{\text{ЛАБ}} = 10 \text{ балів} \times 5 = 50 \text{ балів}$.

Шкала оцінювання (максимально можлива оцінка) тесту: 5 балів.

Шкала оцінювання всіх тестів: $R_T = 5 \text{ балів} \times 8 = 40 \text{ балів}$.

Шкала оцінювання модульної контрольної роботи: $R_{\text{МКР}} = 10 \text{ балів}$.

Шкала стартової оцінки: $R_C = R_{\text{ЛАБ}} + R_T + R_{\text{МКР}} = 50 \text{ балів} + 40 \text{ бали} + 10 \text{ балів} = 100 \text{ балів}$.

Шкала оцінювання залікової роботи: $R_3 = 40 \text{ балів}$.

Підсумкова рейтингова шкала: $R = R_C = 100 \text{ балів}$ або $R = R_{\text{ЛАБ}} + R_{\text{МКР}} + R_3 = 60 \text{ балів} + 40 \text{ балів} = 100 \text{ балів}$.

Мінімально можлива загальна позитивна оцінка: $R_{\text{МИН}} = 60 \text{ балів}$.

Нижня межа позитивного оцінювання кожного контрольного заходу (тест) є не менше 60% від балів, визначених для цього контрольного заходу ($5 \text{ балів} \times 0,6 = 3 \text{ бали}$), а негативний результат оцінюється у 0 балів. Якщо студент не проходив або не з'явився на контрольний захід, його результат оцінюється у 0 балів.

Терміни здачі робіт

Термін здачі лабораторних робіт визначається наступним чином:

- лабораторна робота 1 – з 3 (третього) до 6 (шостого) тижня навчання включно;
- лабораторна робота 2 – з 5 (п'ятого) до 8 (восьмого) тижня навчання включно;
- лабораторна робота 3 – з 7 (сьомого) до 10 (десятого) тижня навчання включно;
- лабораторна робота 4 – з 7 (сьомого) до 12 (дванадцятого) тижня навчання включно;
- лабораторна робота 5 – з 7 (сьомого) до 14 (чотирнадцятого) тижня навчання включно.

Термін здачі тестів визначається:

- часом кінця заняття, на якому проводиться тест, у разі очного навчання;
- 2 (двома) днями після дня заняття, на якому проводиться тест, у разі дистанційного навчання.

Тести проводяться на наступних тижнях навчання: 2 (другий), 4 (четвертий), 6 (шостий), 8 (восьмий), 10 (десятий), 11 (одинадцятий), 13 (тринадцятий), 15 (п'ятнадцятий).

Заохочувальні та штрафні бали

Лабораторні роботи, які здаються до початку терміну здачі, оцінюються у заохочувальні +2 (плюс два) бали додатково до максимально можливої оцінки.

Лабораторні роботи, які здаються після терміну здачі без поважних причин, оцінюються у штрафні -2 (мінус два) бали додатково до максимально можливої оцінки.

Визначення умов отримання позитивного результату календарного контролю

Календарний контроль, як правило, проводиться на 7 та 13 тижні навчання. Необхідною умовою отримання позитивного результату календарного контролю є деяке значення поточної оцінки студента (R_{C1} , R_{C2}), яке не менше деякого мінімального значення ($R_{\text{МИНК1}}$, $R_{\text{МИНК2}}$) відповідно для першого та другого календарного контролю. Мінімальне значення для отримання позитивного результату календарного контролю визначається як 60% від максимально можливої оцінки R_C на час проведення календарного контролю. Тобто, при

здачі робіт згідно визначених термінів на час першого календарного контролю повинні бути здані 2 (дві) лабораторні роботи та 3 (три) тести, а на час другого календарного контролю – 4 (чотири) лабораторні роботи та 7 (сім) тестів. Таким чином:

- $R_{C1} = 10 \text{ балів} \times 2 + 5 \text{ балів} \times 3 = 20 \text{ балів} + 15 \text{ балів} = 35 \text{ балів};$
- $R_{C2} = 10 \text{ балів} \times 4 + 5 \text{ балів} \times 7 = 40 \text{ балів} + 35 \text{ балів} = 75 \text{ балів};$
- $R_{\text{МИНК1}} = 35 \text{ балів} \times 0,6 = 21 \text{ бал};$
- $R_{\text{МИНК2}} = 75 \text{ балів} \times 0,6 = 45 \text{ балів}.$

Тобто, для отримання позитивного результату першого календарного контролю необхідно заробити принаймні 21 бал, для другого календарного контролю – 45 балів.

Визначення умов допуску до заліку

Умовою допуску до заліку є виконання з певною успішністю та необхідної кількості завдань впродовж семестру, а також дотримання вимог академічної доброчесності, що описані в [Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського»](#), ухваленого Рішенням Вченої ради від 5 квітня 2021 р. (Протокол №4). Тобто, студент повинен впродовж семестру заробити деякий ненульовий стартовий рейтинг ($R_c > 0$), виконати модульну контрольну роботу, принаймні 3 (три) лабораторні роботи та принаймні 4 (чотири) тести.

Мінімальний стартовий рейтинг для допуску до заліку ($R_{\text{МИНДОП}}$) визначається як 60% від мінімально можливої загальної позитивної оцінки ($R_{\text{МИН}}$). Звідси значення $R_{\text{МИНДОП}} = R_{\text{МИН}} \times 0,6 = 60 \text{ балів} \times 0,6 = 36 \text{ балів}$. Таким чином, умовою допуску до заліку буде $R_c \geq R_{\text{МИНДОП}}$, тобто, значення стартового рейтингу R_c має бути не менше 36 балів.

Підсумкова рейтингова оцінка переводиться в оцінку за університетською шкалою.

Таблиця переведення підсумкової рейтингової оцінки в університетську шкалу оцінок

Підсумкова рейтингова оцінка R	Університетська шкала оцінок
95...100	Відмінно
85...94	Дуже добре
75...84	Добре
65...74	Задовільно
60...64	Достатньо
36...59	Незадовільно
0...35	Не допущено

Робочу програму навчальної дисципліни (силабус):

Склав асистент кафедри обчислювальної техніки, доктор філософії Шульга М.В.

Ухвалено кафедрою обчислювальної техніки (протокол № 12 від 23 червня 2025 р.)

Погоджено Методичною комісією факультету інформатики та обчислювальної техніки (протокол № 11 від 27 червня 2025 р.)