



Python для інженерних розрахунків в електроенергетиці

Силабус освітнього компоненту

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>перший (бакалавр)</i>
Галузь знань	<i>14 «Електрична інженерія»</i>
Спеціальність	<i>141 «Електроенергетика, електротехніка та електромеханіка»</i>
Освітня програма	<i>Управління, захист та автоматизація енергосистем, Електричні системи і мережі, Нетрадиційні та відновлювані джерела енергії, Електричні станції, Електротехнічні пристрої та електротехнологічні комплекси, Електричні машини й апарати, Електромеханічні системи автоматизації, електропривод та електромобільність</i>
Статус дисципліни	<i>Вибіркові компоненти, цикл професійної підготовки</i>
Форма навчання	<i>Очна (денна), змішана, дистанційна</i>
Рік підготовки, семестр	<i>2-й курс, осінній семестр</i>
Обсяг дисципліни	<i>120 годин / 4 кредитів ECTS (36 год лек., 18 год. ЛР, 18 год.прак., 48 СРС)</i>
Семестровий контроль/ контрольні заходи	<i>Залік / МКР</i>
Розклад занять	http://rozklad.kpi.ua/ <i>1 лекція 1 раз на тиждень, 1 ЛР/прак. заняття 1 раз на тиждень;</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: <i>Заколюдажний Володимир Васильович,</i> <i>zakolodyazhny-fea@ill.kpi.ua</i>
Розміщення курсу	<i>Google Classroom</i> https://classroom.google.com/c/NzY5NTc5Mjg5MTAx?cjc=jlj7js5z

Програма навчальної дисципліни

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

Метою навчальної дисципліни є оволодіння основою однієї з сучасних мов та технологій програмування, набуття умінь та досвіду зі створення прикладних програм, програмних комплексів з врахуванням особливостей сучасних мов та технологій програмування при вирішенні задач у науковій, інженерно-технічній та економічній сферах діяльності та сприяння розвитку логічного та аналітичного мислення студентів. А також – є закріплення у студентів відповідних загальних та фахових спеціальних компетентностей згідно ОПП.

Предмет навчальної дисципліни – основні принципи програмування та створення алгоритмів для розв'язання прикладних задач з використанням мови програмування Python. Знайомство з мовою програмування Python та її базовими бібліотеками. Робота з операторами розгалуження та ітераційними циклами.

Програмні результати навчання, на покращення яких спрямована дисципліна: ПРО6. Застосовувати прикладне програмне забезпечення, мікроконтролери та мікропроцесорну

техніку для вирішення практичних проблем у професійній діяльності; ПР18. Вміти самостійно вчитися, опановувати нові знання і вдосконалювати навички роботи з сучасним обладнанням, вимірювальною технікою та прикладним програмним забезпеченням; ПР24 Вміти розробляти алгоритми вирішення задач в галузі управління, захисту та автоматизації енергосистем з використанням математичного апарату та сучасного програмного забезпечення.

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

Для успішного засвоєння дисципліни студент повинен володіти англійською мовою на базовому рівні та математикою в рамках шкільної програми (особливо – розділ математична логіка), «Обчислювальна техніка та програмування. Частина 1», та частково «Вища математика 1».

Отримані знання можуть бути корисні для успішного засвоєння таких дисциплін як: «Обчислювальна техніка та програмування-2», «Обчислювальні методи та алгоритмізація», «Математичні задачі енергетики», «Пакети прикладних програм для ПЕОМ», "Релейний захист та автоматизація енергосистем", «Теорія автоматичного керування», тощо та подальшого якісного виконання досліджень за темою атестаційної роботи. Також, матеріал даної навчальної дисципліни може бути застосований при вивченні інших навчальних дисциплін.

3. Зміст навчальної дисципліни

Дисципліну структурно розподілено на кілька тем, а саме:

Тема 1. Середовища програмування для Python (IDE).

Кількість годин: лекції – 4 годин, лабораторні/практичні – 4 год.

Тема 2. Базовий синтаксис мови Python.

Кількість годин: лекції – 6 годин, лабораторні/практичні – 6 год.

Тема 3. Основні конструкції мови Python.

Кількість годин: лекції – 20 годин, лабораторні/практичні – 20 год.

4. Навчальні матеріали та ресурси

Основні інформаційні ресурси:

1. Основи програмування алгоритмічною мовою Python [Електронний ресурс]: навч. посіб. для студ. освітньої програми «Комп'ютерні системи та мережі» спеціальності 123 «Комп'ютерна інженерія» / М.А. Новотарський – Електронні текстові дані (1 файл: 18 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 701с
2. Основи програмування. Python. Частина 1 [Електронний ресурс]: підручник для студ. спеціальності 122 "Комп'ютерні науки", спеціалізації "Інформаційні технології в біології та медицині" / А. В. Яковенко ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,59 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 195 с. (посилання <https://ela.kpi.ua/handle/123456789/25111> або <https://ela.kpi.ua/server/api/core/bitstreams/dbbe8ff5-11d7-4a92-918c-d1445c3d20a7/content>)
3. Івановський О.А, Парненко В.С Інформатика. Програмування на PYTHON [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Конструювання та дизайн машин» спец. 131 «Прикладна механіка» / О.А. Івановський, В.С. Парненко, ; КПІ ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл). – Київ : КПІ ім. Ігоря Сікорського, 2023. – 232 с. (посилання <https://ela.kpi.ua/handle/123456789/62922> або <https://ela.kpi.ua/bitstreams/c26afa66-3722-4411-a4e0-f8b7988dbc2d/download>)
4. Основи інформатики та програмування. Конспект лекцій [Електронний ресурс]: навчальний посібник для здобувачів першого (бакалаврського) рівня вищої освіти за освітньою програмою «Комп'ютерні технології в біології та медицині» спеціальності

122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського ; уклад.: І. В. Федорін. – Електронні текстові дані (1 файл: 1,09 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2023. – 179 с. (посилання <https://ela.kpi.ua/handle/123456789/63428> або <https://ela.kpi.ua/bitstreams/2a094e6b-2dbb-4004-a92a-7793e2e6abfe/download>)

5. Мова Python для інженерних та наукових задач [Електронний ресурс] : підручник для здобувачів ступеня бакалавра за спеціальністю 162 «Біотехнології та біоінженерія», магістра за спеціальністю за спеціальністю 162 «Біотехнології та біоінженерія», бакалавра за спеціальністю 104 «Фізика та астрономія» та магістра за спеціальністю 104 «Фізика та астрономія» / О. Ю. Горобець, С. В. Горобець, К. Ю. Хахно ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 8,95). – Київ : КПІ ім. Ігоря Сікорського, 2024. – 277 с. (посилання <https://ela.kpi.ua/handle/123456789/67188>)
6. Алгоритми та структури даних. Конспект лекцій дистанційного навчання [Електронний ресурс] : навчальний посібник для студентів, які навчаються за спеціальністю 124 «Системний аналіз» / КПІ ім. Ігоря Сікорського ; уклад.: Ю. М. Селін – Електронні текстові дані (1 файл: 41,27 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 156 с. (посилання <https://ela.kpi.ua/handle/123456789/57195>)

Додаткові інформаційні ресурси:

7. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліффорд Стайн. Вступ до алгоритмів — К. : К. І. С., 2019. — 1288 с. ISBN 978-617-684-239-2
8. Кнут Д. Искусство программирования, том 1. Основные алгоритмы – The Art of Computer Programming, vol.1. Fundamental Algorithms / Д. Кнут. – М. : "Вильямс", 2006. – 720 с.
9. The Art of Computer Programming,. (Reading, Massachusetts: Addison-Wesley, 2023), 3904pp. ISBN 978-0-13-793510-9, 0-13-793510-2. Volume 1: Fundamental Algorithms. Third Edition (Reading, Massachusetts: Addison-Wesley, 1997), 650pp. ISBN 978-0-201-89683-1, 0-201-89683-4 (2011).
10. The Python Software Foundation (посилання <https://www.python.org/>)
11. W3Schools. Python Tutorial (посилання <https://www.w3schools.com/python/default.asp>)
12. W3SchoolsUA. Python Підручник (<https://w3schoolsua.github.io/python/index.html#gsc.tab=0>)
13. LinkedIn Learning. Python Essential Training (посилання <https://www.linkedin.com/learning/python-essential-training-18764650>)
14. Відеокурс на YouTube: Learn Python – Full Course for Beginners [Tutorial] (посилання https://youtu.be/rfscVS0vtbw?si=_DwJ1zNk4XdgfSrB)
15. Freecodecamp.org. Legacy Python For Everybody (посилання <https://www.freecodecamp.org/learn/python-for-everybody/>)
16. PEP 8 – the Style Guide for Python Code (посилання <https://pep8.org/#pep-8-%E2%80%94-the-style-guide-for-python-code>)

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на інформаційні джерела)
1	Вступне заняття (розглядаються організаційні питання)
2	Тема 1. Середовища програмування для Python (IDE). Загальні поняття про мову Python. Вибір та встановлення середовища програмування Літературні джерела: [1], [2..5], [10, 11]
3	Тема 2. Базовий синтаксис мови Python Структура програми Python. Ідеологія мови ("Дзен Пайтона"). Типи даних у мові Python. Правила найменування змінних. Прості типи: bool, int, float, complex, None, str, bytes, bytearray. Спискові типи: list, tuple, set, dict, range. Змінювані й незмінювані типи. Послідовності. Словники. Літературні джерела: [1], [2..5], [10, 11]
4	Тема 2. Базовий синтаксис мови Python Функції, модулі, об'єкти. Присвоювання значення. Групове призначення. Подвійне посилання. Позиційне призначення. Перевірка типу даних. Перетворення типів даних. Видалення змінної. Літературні джерела: [1], [2..5], [10, 11]
5	Тема 2. Базовий синтаксис мови Python Оператори у мові Python. Арифметичні оператори. Точність чисел. Побітові оператори. Спеціалізовані оператори (для послідовностей, для конкатенації, для повторення). Оператори входження. Оператори присвоювання. Оператори порівняння. Логічні оператори. Оператори тотожності. Пріоритет виконання операторів. Літературні джерела: [1], [2..5], [10, 11]
6	Тема 3. Основні конструкції мови Python Оператори розгалуження й цикли. Синтаксис операторів. Оператор if-elif-else. Вкладені оператори. Вибір варіантів: switch-case, вибір по ключу словника. Літературні джерела: [1], [2..5], [10, 11]
7-8	Тема 3. Основні конструкції мови Python Оператор циклу for-in-else. Функція range. Функція enumerate(), next(). Оператор циклу while. Оператори continue та break. Літературні джерела: [1], [2..5], [10, 11]
9	Тема 3. Основні конструкції мови Python Операції і функції для числових типів. Арифметичні операції. Модуль math, random. Літературні джерела: [1], [2..5], [10, 11]
10-11	Тема 3. Основні конструкції мови Python Створення рядків і робота з рядками. Спеціальні символи. Документування. Операції з рядками. Форматування рядків. Функції та методи роботи з рядками. Літературні джерела: [1], [2..5], [10, 11]
12	Тема 3. Основні конструкції мови Python Функції користувача. Параметри. Глобальні та локальні змінні. Функція як об'єкт. Вкладені функції. Область видимості. Літературні джерела: [1], [2..5], [10, 11]
13	Тема 3. Основні конструкції мови Python Анонімні функції. Генератори (yield). Анотації функцій. Літературні джерела: [1], [2..5], [10, 11]

14	Тема 3. Основні конструкції мови Python Модулі та пакети. Імпортування модулів. Стандартні бібліотеки та сервіси. Літературні джерела: [1], [2..5], [10, 11]
15-16	Тема 3. Основні конструкції мови Python Робота з файлами. Обробка виключень. Літературні джерела: [1], [2..5], [10, 11]
17	Тема 3. Основні конструкції мови Python Робота з графікою. Літературні джерела: [1], [2..5], [10, 11]
18	МКР

Лабораторні заняття (комп'ютерний практикум)

№ з/п	Назва роботи	Кількість ауд. годин
1	Вступне заняття	2
1	Заняття №1. Встановлення компілятора та середовища розробки (IDE) Літературні джерела: [1-5] [10, 11]	4
2	Заняття №2. Типи даних у мові Python Літературні джерела: [1-5] [10, 11]	4
3	Заняття №3. Функції, модулі, об'єкти Літературні джерела: [1-5] [10, 11]	4
4	Заняття №4. Оператори розгалуження й цикли Літературні джерела: [1-5] [10, 11]	4
	ЗАГАЛОМ	18

Практичні заняття (комп'ютерний практикум)

№ з/п	Назва роботи	Кількість ауд. годин
1	Заняття №1. Операції і функції для числових типів Літературні джерела: [1-5] [10, 11]	4
2	Заняття №2. Створення рядків і робота з рядками Літературні джерела: [1-5] [10, 11]	6
3	Заняття №3. Функції користувача. Анонімні функції Літературні джерела: [1-5] [10, 11]	4
4	Заняття №4. Робота з графікою Літературні джерела: [1-5] [10, 11]	4
	ЗАГАЛОМ	18

Контрольна робота

Метою контрольної роботи є закріплення та перевірка теоретичних знань із освітнього компоненту, набуття студентами практичних навичок.

Модульна контрольна робота (МКР) виконується після вивчення предмету. Кожний студент отримує індивідуальне завдання, відповідно до якого необхідно відповісти на тестові запитання.

Самостійна робота студентів

№ з/п	Вид самостійної роботи	Кількість годин
1	Підготовка до аудиторних занять (лекцій) Літературні джерела: [1], [2..5], [10, 11]	8
2	Підготовка до практичних занять	30

	Літературні джерела: [1], [2..5], [10, 11]	
3	Підготовка до МКР Літературні джерела: [1], [2..5], [10, 11]	4
4	Підготовка до заліку Літературні джерела: [1], [2..5], [10, 11]	6
	ЗАГАЛОМ	48

Політика та контроль

6. Політика навчальної дисципліни (освітнього компонента)

Система вимог, які викладач ставить перед студентом:

- правила відвідування занять: відповідно до Наказу 1-273 від 14.09.2020 р. заборонено оцінювати присутність або відсутність здобувача на аудиторному занятті, в тому числі нараховувати заохочувальні або штрафні бали.
- правила поведінки на заняттях: студент має можливість отримувати бали за відповідні види навчальної активності на лекційних та лабораторних заняттях, передбачені РСО дисципліни. Використання засобів зв'язку для пошуку інформації в інтернеті, в дистанційному курсі на платформі Сікорський здійснюється за умови вказівки викладача;
- правила призначення заохочувальних та штрафних балів: заохочувальні та штрафні бали не входять до основної шкали РСО, а їх сума не перевищує 10% стартової шкали. Заохочувальні бали нараховують за участь у факультетських та інститутських олімпіадах та наукових конференціях. Штрафні бали нараховують за несвоєчасну подачу студентом лабораторних робіт.
- для отримання хороших оцінок – важливо вчасно виконувати завдання. Також, допускається випереджати графік здавання лабораторних чи практичних робіт.
- політика дедлайнів: несвоєчасне лабораторних робіт передбачає нарахування штрафних балів (зниження оцінки). Якщо студент не проходив або не з'явився на МКР, його результат оцінюється у 0 балів;
- політика щодо академічної доброчесності: Кодекс честі Національного технічного університету України «Київський політехнічний інститут» <https://kpi.ua/files/honorcode.pdf> встановлює загальні моральні принципи, правила етичної поведінки осіб та передбачає політику академічної доброчесності для осіб, що працюють і навчаються в університеті, якими вони мають керуватись у своїй діяльності, в тому числі при вивченні та складанні контрольних заходів з дисципліни;
- при використанні цифрових засобів зв'язку з викладачем (мобільний зв'язок, електронна пошта, переписка на форумах та у соц.мережах тощо) необхідно дотримуватись загальноприйнятих етичних норм, зокрема бути ввічливим та, по можливості – обмежувати спілкування робочим часом викладача. Допускається задавати питання і в інший час, по мірі їх виникнення.
- дисципліна ведеться з допомогою використання Google-classroom (та Google Meet), але для оперативного спілкування з викладачем рекомендується використовувати один з месенджерів (контакт надається до чи під час вступного заняття). Якщо питання загальне – то запитувати можна через створену спеціальну загальну групу по цьому предмету. Якщо питання специфічне – то можна задавати його приватно у тому ж месенджері.

7. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Поточний контроль: виконання практичних/лабораторних завдань, МКР.

Календарний контроль: провадиться двічі в семестр як моніторинг поточного стану виконання вимог силабусу, і базується на кількості виконаних завдань на дату КК. На перший календарний контроль необхідно виконати не менше 25 % усіх завдань на цю дату (тобто, не

менше половини від заданого, по графіку, на дату календарного контролю). На другий календарний контроль необхідно виконати не менше 50 % усіх завдань (тобто, не менше половини від заданого, по графіку, на дату календарного контролю).

Семестровий контроль: залік

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно

Загальна рейтингова оцінка студента після завершення семестру складається з балів, отриманих за:

- виконання та захист семестрових завдань (лабораторних/практичних робіт);
- виконання модульної контрольної роботи (МКР).

ЛР	ПР	МКР	R
40	40	20	100

Якщо наприкінці семестру після проходження всіх контрольних заходів з дисципліни студент отримав не менше ніж 60 рейтингових балів, він отримує позитивну оцінку відповідно набраних протягом семестру рейтингових балів.

Умови допуску: виконати 85% від кількості усіх семестрових завдань (допускається не виконати 15% за наявності особливих об'єктивних причин), та набрати не менше 30 балів.

Якщо наприкінці семестру після проходження всіх контрольних заходів з дисципліни студент отримав менше ніж 30 рейтингових балів, або виконав менше 85% усіх завдань –, він отримує оцінку "не допущено".

1. Лабораторні роботи, за кожну роботу:

Кожна лабораторна робота оцінюється за 100 бальною шкалою.

Протокол має містити всі необхідні пункти, відповідно до Навчального посібника з лабораторного практикуму.

За невчасну подачу звіту нараховуються штрафні бали за кожен тиждень затримки (1..7 днів після терміну вказаного викладачем – 3 бали, 8..14 днів – 6 балів, і т.д.).

За кожну роботу студент може отримати:

- «відмінно»– 95-100 балів, повне виконання завдання та усна відповідь на контрольні запитання (не менше 90% потрібної інформації);
- «добре» - 75-84 бали та «дуже добре» 85-94 бали, дано відповіді на переважну більшість тестових питань, та робота містить несуттєві помилки при виконання та оформленні звіту;
- «достатньо» - 60-64 бали та «задовільно» - 65-74 бали, багато помилок у відповідях на тестові запитання, значні помилки при розв'язанні завдання (програма), та при оформленні протоколу та побудові блок-схем алгоритмів програми;
- «незадовільно» - 0 балів, студент не набрав необхідну кількість балів для позитивної оцінки або не здав роботу. Це означає, що роботу треба доздати у відведені навчальним графіком терміни.

В кінці семестру для зданих лабораторних робіт знаходиться середнє арифметичне значення $(Л1+Л2+...+Лn)/N$, де $Л1, Л2, ..., Лn$ – оцінки за відповідні роботи), отримане значення помножується на коефіцієнт 0.4, тобто переводиться в бали РСО до 40 балів.

2. Виконання практичних робіт

Кожне завдання оцінюється за 100 бальною шкалою.

Критерії оцінювання:

- практичне завдання не виконано або протокол не представлений – повертається на відпрацювання або доопрацювання - 0 балів.
- виконання практичного завдання, самостійне виконання обчислень, оформлення протоколу досліджень, відповідь на питання до захисту надана повністю помилковою – 60 балів;
- виконання практичного завдання, самостійне виконання обчислень, оформлення протоколу досліджень, відповідь на питання до захисту надана з суттєвими помилками – 60-69 балів;
- виконання практичного завдання, самостійне виконання обчислень, оформлення протоколу досліджень, відповідь на питання до захисту надана з несуттєвими помилками – 70-79 балів;
- виконання практичного завдання, самостійне виконання обчислень, оформлення протоколу досліджень, відповідь на питання до захисту має ряд неточностей – 80-89 балів;
- виконання практичного завдання, самостійне виконання обчислень, оформлення протоколу досліджень, відповідь на питання до захисту має одну неточність – 90-99 балів;
- виконання практичного завдання, самостійне виконання обчислень, оформлення протоколу досліджень, повна відповідь на питання за темою практичної роботи – 100 балів.

В кінці семестру для зданих лабораторних робіт знаходиться середнє арифметичне значення $(P1+P2+...+Pn)/N$, де $P1, P2, ..., Pn$ – оцінки за відповідні роботи), отримане значення помножується на коефіцієнт 0.4, тобто переводиться в бали РСО до 40 балів.

3. Модульна контрольна робота.

Оцінюється за 100 бальною шкалою. Складається з відповідей на запитання і відображає відсоток правильних відповідей на питання тестів. Набрані бали в кінці семестру перераховуються з коефіцієнтом 0.2, що дає від 0 до 20 балів РСО.

Форма семестрового контролю – залік

Залік проводиться згідно Положення про поточний, календарний та семестровий контроль результатів навчання в КПІ ім. Ігоря Сікорського.

1. Залік проводиться в період останніх двох тижнів теоретичного навчання у семестрі, як правило, на останньому за розкладом занятті з навчальної дисципліни (освітнього компонента).

2. Здобувач отримує позитивну залікову оцінку за результатами роботи в семестрі, якщо має підсумковий рейтинг за семестр не менше 60 балів та виконав умови допуску до семестрового контролю, які визначені РСО.

2. Якщо здобувач виконав умови РСО щодо допуску до семестрового контролю, але має підсумковий рейтинг за семестр менше 60 балів, або хоче підвищити поточну оцінку з відповідної навчальної дисципліни (освітнього компонента), він виконує залікову контрольну роботу (проходить залікову співбесіду), як правило, на останньому за розкладом аудиторному занятті з відповідної навчальної дисципліни (освітнього компонента).

Критерії оцінювання залікової роботи

Ваговий бал кожної задачі – 30.

Ваговий бал кожного теоретичного питання – 20.

Максимальний бал за залікову роботу – 100.

Критерії оцінювання задачі

- правильне розв'язання задачі – 100% від кількості балів за задачу;
- часткове розв'язання задачі, наявність незначних помилок – 60-95% від кількості балів за задачу;
- часткове розв'язання задачі, наявність значних помилок – 10-55% від кількості балів за задачу;
- відсутність відповіді – 0 балів.

Критерії оцінювання теоретичного питання

- студент дав вичерпну відповідь на питання, дає чіткі визначення всіх понять і величин, відповіді логічні і послідовні – 18-20 балів;
- майже вичерпна відповідь, наявність незначних неточностей – 15-17 балів;

- часткова відповідь, показує знання, але недостатньо розуміє суть процесів, наявність незначних неточностей – 12-14 балів;
- часткова відповідь, недостатнє розуміння суті процесів, наявність значних помилок – 1-11 балів;
- відсутність відповіді – 0 балів.

Робочу програму навчальної дисципліни (силабус) складено
асистент кафедри автоматизації енергосистем Заколюдажний В.В.

Ухвалено кафедрою автоматизації енергосистем ФЕА (протокол № 14 від 10.06.2025 р.)
Погоджено навчально-методичною комісією факультету (протокол №6 від 27.06.2025 р.)

Індивідуальні завдання на модульну контрольну роботу

Для заданих викладачем питань, зі списку нижче, написати відповіді. Результат завантажити у відповідне завдання класруму, у форматі pdf.

1. Що таке масиви та списки в Python?
2. Як створити список в Python?
3. Як отримати доступ до елементів списку?
4. Що таке індекс у списку? Як працює індексація в Python?
5. Як змінити значення елемента списку?
6. Які основні методи для роботи зі списками в Python? Наведіть приклади.
7. Як додавати та видаляти елементи зі списків?
8. Які види сортування списків існують в Python?
9. Як використовувати списки у циклах?
10. Як копіювати список в Python? Яка різниця між поверхневим та глибоким копіюванням списків?
11. Що таке рядки в Python? Як створити рядок?
12. Що таке екранування символів в рядках? Для чого воно використовується?
13. Як виконується конкатенація (з'єднання) рядків?
14. Що таке зрізи рядків? Як їх використовувати?
15. Які основні вбудовані методи рядків в Python? Наведіть приклади їх використання.
16. Що таке регулярні вирази? Для чого вони використовуються при роботі з рядками?
17. Як зробити рядок незмінним? Чи можна змінити символ в рядку?
18. Як перевести число в рядок і навпаки?
19. Як рядки використовуються в конструкціях управління потоком (цикли, умови)?
20. Що таке форматування рядків та як воно використовується в Python?
21. Що таке кортежі в Python? Які їх основні властивості?
22. Як створити кортеж? Як отримати доступ до елементів кортежу?
23. Чи можна змінити елемент кортежу? Чому?
24. Що таке словники в Python? Як створити словник?
25. Як отримати доступ до елементів словника? Як додати, змінити або видалити елемент словника?
26. Що таке множини в Python? Які властивості мають множини?
27. Як створити множину? Які операції можна виконувати з множинами?
28. Які вбудовані методи доступні для кортежів, словників та множин? Наведіть приклади.
29. Які ситуації найбільш підходять для використання кортежів, словників та множин?
30. Як конвертувати один тип даних (кортеж, словник, множину) в інший?
31. Що таке ітерація по кортежам, словникам і множинам?
32. Що таке помилка в програмуванні? Які основні типи помилок ви знаєте?
33. Що таке винятки в Python? Як вони пов'язані з помилками?
34. Які основні вбудовані винятки існують в Python? Для чого вони використовуються?
35. Що таке обробка винятків? Яким чином вона реалізована в Python?
36. Як використовувати інструкції try/except для обробки винятків? Дайте приклади.
37. Яка різниця між використанням блоків except, else та finally? В яких ситуаціях вони використовуються?
38. Що таке функція в Python і як її створити?
39. Що таке параметри та аргументи функції? Яка різниця між ними?
40. Що таке значення за замовчуванням у функціях? Як вони використовуються?
41. Як Python розуміє область видимості змінних у функціях?
42. Що таке модуль в Python? Як імпортувати модуль у свій скрипт?

43. Які основні операції можна виконувати з файлами в Python? Як відкрити, прочитати, записати та закрити файл?
44. Що таке режими відкриття файлів та які з них ви знаєте?
45. Які функції використовуються для читання та запису даних у файл?
46. Що таке виключення та як їх обробляти при роботі з файлами?
47. Що таке анонімна функція або лямбда-функція? Для чого вона використовується?
48. Що таке об'єктно-орієнтоване програмування (ООП)?
49. Що таке клас і об'єкт в контексті ООП? Яка їх різниця?
50. Як створити клас в Python?
51. Як створити об'єкт (екземпляр класу) в Python?
52. Що таке метод класу? Як його створити та викликати?
53. Що таке атрибут класу? Як його створити та використовувати?
54. Що таке інкапсуляція? Як вона реалізується в Python?
55. Що таке наслідування? Як створити підклас в Python?
56. Що таке поліморфізм? Надайте приклад з Python.
57. Чому об'єктно-орієнтоване програмування важливе для розробки програмного забезпечення?
58. Що таке бітові операції та чому вони важливі в програмуванні?
59. Які основні бітові операції ви знаєте і який вплив вони мають на біти?
60. Як працює бітова операція AND в Python? Дайте приклад.
61. Як працює бітова операція OR в Python? Дайте приклад.
62. Що робить бітова операція XOR в Python? Дайте приклад.
63. Як працює бітова операція NOT в Python? Дайте приклад.
64. Як працюють операції зсуву вліво (<<) та зсуву вправо (>>) в Python? Дайте приклад.
65. У яких випадках ви б можете використовувати бітові операції в програмуванні?