



ПРОГРАМУВАННЯ. Частина 1

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	F(12) Інформаційні технології
Спеціальність	F7(123) Комп'ютерна інженерія
Освітня програма	Комп'ютерні системи та мережі
Статус дисципліни	Нормативна
Форма навчання	Очна (денна)
Рік підготовки, семестр	1 курс, осінній семестр
Обсяг дисципліни	6 кредитів, 180 год. лекції 60 год., лабораторний практикум 30 год., СРС 90 годин.
Семестровий контроль/ контрольні заходи	Екзамен/Лабораторний практикум
Розклад занять	Згідно розкладу на весняний семестр поточного навчального року за адресою https://schedule.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор(): д.т.н, професор, Новотарський Михайло Анатолійович novotarskyi.mykhailo@iit.kpi.ua Лабораторні роботи: асистент, Грибенко Єгор Володимирович
Розміщення курсу	https://classroom.google.com/c/NDE3ODIzNzU2MDk1?cjc=nnenp23

Програма навчальної дисципліни

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

Метою навчальної дисципліни «Програмування. Частина 1» є оволодіння основами 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.

Основні завдання навчальної дисципліни полягають:

- **у вивченні** основних етапів процесу проектування програмного забезпечення з використанням агрегатора "Anaconda", що включає IDE "PyCharm" та "Jupyter notebook"; типових алгоритмічних конструкцій, які є характерними для скриптової мови програмування Python; принципів процедурного, структурного та об'єктно-орієнтованого програмування в Python; особливостей застосування сучасних програмних засобів на базі мови програмування Python, призначених для вирішення економічних задач; базових типів даних, операторів управління програмою, правил роботи з функціями методами типів мови програмування Python; системи вводу-виводу та основних принципів роботи з файлами в Python; правил роботи з регулярними виразами та шаблонами; основних підходів, що використовуються при створенні графічного інтерфейсу користувача для програм на Python;

- **у оволодінні** основами програмування, тобто умінні створювати програми мовою Python, забезпечуючи розв'язування задач математичного характеру; створення і обробку складних структур даних; найпростішу обробку файлів; використання стандартних функцій та функцій користувача; використання методів стандартних типів мови Python; використання раніше складених програм і здійснення супроводу програм, внесення змін в програму, виконання налаштування програм за допомогою вбудованих інструментальних засобів;

- **у набутті** досвіду з володіння методами та технологіями програмування алгоритмічною мовою Python, створення та використання структур даних, які дозволяють ефективно розв'язувати практичні задачі, створення програм з графічним інтерфейсом користувача (GUI).

Таке поєднання теоретичних та практичних знань та вмінь сприяє як достатньому оволодінню технологією роботи з конкретним середовищем розробки, так і полегшенню освоєння нових засобів програмування та переходу в подальшому на нові середовища та операційні системи.

На початку вивчення дисципліни кожен студент має бути ознайомлений з програмою дисципліни і формами організації навчання, а також з усіма видами контролю та методикою оцінювання знань. Тематичний план дисципліни включає змістовний модуль «Основи програмування», який складається з окремих самостійних блоків, що логічно поєднані в кілька навчальних елементів дисципліни за змістом та взаємозв'язками.

Основні компетентності

Здобувачі ступеня бакалавра після засвоєння нормативної дисципліни «Програмування. Частина 1» мають набути таких:

загальних компетентностей:

ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК07. Вміння виявляти, ставити та вирішувати проблеми.

ЗК08. Здатність працювати в команді.

фахових компетентностей:

ФК02. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення

ФК16. Здатність проектувати, розробляти, впроваджувати та обслуговувати програмноапаратне забезпечення для високопродуктивних паралельних та розподілених комп'ютерні систем та їх складових на сучасній елементній базі, зокрема, з використанням ПЛІС і систем автоматизованого проектування.

Основні програмні результати навчання

ПРН03. Знати новітні технології в галузі комп'ютерної інженерії

ПРН04. Знати та розуміти вплив технічних рішень в суспільному, економічному, соціальному і екологічному контексті.

ПРН05. Мати знання основ економіки та управління проектами

ПРН07. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

ПРН08. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей

ПРН12. Вміти ефективно працювати як індивідуально, так і у складі команди

ПРН16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення

ПРН18. Використовувати інформаційні технології та для ефективного спілкування на професійному та соціальному рівнях

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

Матеріал курсу пов'язаний з матеріалами, які вивчаються у навчальних дисциплінах «Комп'ютерна логіка», «Алгоритми та структури даних», «Вища математика».

Матеріал даної навчальної дисципліни буде застосований при вивченні наступних навчальних дисциплін: «Дискретна математика», «Алгоритми та методи обчислень», «Комп'ютерне моделювання», «Програмна інженерія», «Системне програмування», «Технологія розподілених обчислень», «Мережні та інформаційні технології», «Комп'ютерне моделювання», «Основи програмної інженерії», «Системне програмування», «Технологія розподілених обчислень», «Мережеві та інформаційні технології» та інші.

3. Зміст навчальної дисципліни

Розділ 1. Вступ в курс лекцій «Програмування».

Лекція 1

Тема 1.1. Методи та задачі курсу, зв'язок з іншими дисциплінами. Принципи створення архітектури сучасних комп'ютерів.

Тема 1.2. Загальне поняття про програмне забезпечення сучасних комп'ютерів: базовий рівень, системний рівень, службовий рівень, прикладний рівень. Поняття про програмне забезпечення системного рівня. Класифікація службових програмних засобів. Класифікація прикладного програмного забезпечення.

Тема 1.3. Алгоритмічні мови й системи програмування. Рівні алгоритмічних мов програмування. Опис структури середовищ розробки програмного забезпечення.

Розділ 2. Загальні відомості про скриптову мову програмування Python та інтегроване середовище розробки програм PyCharm, інтегратор Anaconda

Лекція 2

Тема 2.1. Основні характеристики мови Python: імперативне програмування, об'єктно-орієнтоване програмування, подійно-керовані програми (GUI), формати представлення даних. Технологія освоєння мови Python: семантика, синтаксис та прагматика. Історія створення мови Python. Дзен Пайтона.

Тема 2.2. Короткі відомості про інсталяцію Python. Середовище програмування на Python IDLE. Основні правила роботи в середовищі IDLE, меню.

Тема 2.3. Установка інтегратора Anaconda. Середовище розробки PyCharm, меню. Початкові відомості про Jupyter notebook.

Розділ 3. Змінні в мові Python

Лекція 3

Тема 3.1. Іменування змінних, список ключових слів та інструкція для їх виводу. Базові типи даних скриптової мови програмування Python.

Тема 3.2. Способи присвоювання значення змінним. Визначення та перевірка типу даних. Способи перетворення типів даних. Видалення змінної.

Лекція 4

Тема 3.3. Кеширування, позиційне присвоювання

Тема 3.4. Функції для перетворення типів даних

Розділ 4. Оператори в мові програмування Python

Лекція 5

Тема 4.1. Загальне поняття про оператори, арифметичні оператори

Тема 4.2. Точність представлення дійсного числа. Точні обчислення

Тема 4.3. Побітові оператори

- Тема 4.4. Оператори для послідовностей.
- Тема 4.5. Оператори присвоювання.
- Тема 4.6. Оператори порівняння.
- Тема 4.7. Логічні оператори.
- Тема 4.7. Спеціальні оператори. Пріоритет виконання операторів

Лекція 6

- Тема 4.7. Оператор розгалуження if-elif-else та його варіанти використання
- Тема 4.8. Оператор циклу for-in-else
- Тема 4.9. Використання функцій range() та enumerate() у операторі циклу for
- Тема 4.10. Оператор while та його застосування
- Тема 4.11. Оператори continue та break

Розділ 5. Представлення чисел в мові Python

Лекція 7

- Тема 5.1. Базові типи для представлення чисел. Представлення чисел в десятковій системі числення. Операції над числами.
- Тема 5.2. Числа, що задані з фіксованою точністю за підтримки модуля decimal. Виконання операцій над дробами за підтримки методів модуля fractions. Формат представлення та операції над комплексними числами. Вбудовані функції для роботи з числами.

Лекція 8

- Тема 5.3. Стандартні константи модуля math. Основні функції модуля math для роботи з числами.
- Тема 5.4. Модуль random. Генерація випадкових чисел
- Тема 5.5. Рядки у мові Python. Поняття про кодову сторінку. Базові строкові типи в Python та сфери їх застосування. Способи створення рядка. Вплив модифікатора r на правила форматування рядка. Операції з рядками.

Розділ 6. Форматування рядків. Функції та методи для роботи з рядками.

Лекція 9

- Тема 6.1. Оператор форматування рядків % та його параметри
- Тема 6.2. Методи форматування center(), ljust(), rjust() та zfill()
- Тема 6.2. Метод format(), його параметри та способи застосування. О
- Тема 6.3. Літеральна інтерполяція рядків (f-рядки)
- Тема 6.4. Шаблонізація (Templating)

Лекція 10.

- Тема 6.5. Функції для роботи з рядками.
- Тема 6.6. Методи рядкового типу
- Тема 6.7. Функції для роботи з символами

Розділ 7. Типи даних bytes та bytearray

Лекція 11

- Тема 7.1. Способи створення об'єктів типу bytes: за допомогою функції bytes з різними видами параметрів, за допомогою методу рядків encode(), за допомогою префікса b"xxx", за допомогою методу bytes.fromhex().
- Тема 7.2. Способи створення об'єктів типу bytearray: за допомогою функції bytearray з різними видами параметрів, за допомогою методу bytearray.fromhex(). Методи зміни об'єкта типу bytearray.
- Тема 7.3. Шифрування рядків за допомогою функцій модуля hashlib.

Розділ 8. Списки, кортежі, множини і діапазони в мові Python

Лекція 12

- Тема 8.1. Означення списків кортежів та множин
- Тема 8.2. Особливості створення списків та їх копіювання

Тема 8.3. Операції над списками та способи позиційного присвоювання

Лекція 13

Тема 8.4. Перебір елементів у списку

Тема 8.5. Застосування функцій `range()` та `enumerate()` при модифікації списків

Тема 8.6. Генератори списків та вирази-генератори

Тема 8.7. Функції `sum()`, `map()`, `zip()`, `filter()` та `reduce()`

Тема 8.8. Функції додавання та видалення елементів списку

Лекція 14

Тема 8.9. Функції вибору елементів `any()` та `all()`

Тема 8.10. Функції та методи сортування списків

Розділ 9. Означення, властивості та застосування кортежів множин та діапазонів. Ітератори.

Лекція 14 (продовження)

Тема 9.1. Незмінювані послідовності типу кортеж. Основна властивість типу «множина». Особливості змінюваних і незмінюваних множин. Основні властивості діапазонів.

Тема 9.2. Методи створення кортежів. Одержання елемента кортежу по індексу.

Тема 9.3. Створення множини за допомогою функції `set()`. Перебір елементів множини в циклі `for`. Означення та застосування операторів на множинах, які виконують логічні перетворення множин. Методи для роботи з множинами. Генератори множин та їх синтаксис.

Тема 9.4. Незмінювані множини типу `frozenset`. Оператори та методи, що підтримують перетворення даних типу `frozenset`. Розгляд даних типу «діапазон» як послідовностей. Параметри та формат функції `range` для задавання діапазону.

Тема 9.5. Функції, що підтримують перетворення даних типу «діапазон». Ітератори, що представлені функціями модуля `itertools`.

Розділ 10. Означення, властивості та застосування словників.

Лекція 15

Тема 10.1. Означення словників як відображень та порівняння їх властивостей з властивостями асоціативних масивів в інших мовах програмування. Способи створення словників.

Тема 10.2. Операції над словниками: доступ до елементів словника, перевірка існування ключа. Визначення кількості ключів.

Тема 10.3. Вилучення елементів словника за функцією `del`. Перебір елементів словника за допомогою циклу `for`. Сортування по ключах. Методи для роботи зі словниками. Генератори словників.

Розділ 11. Визначення та властивості функцій користувача в Python

Лекція 16

Тема 11.1. Визначення функції як фрагменту коду багаторазового використання. Схема створення функції за допомогою ключового слова `def`. Інструкція `return`.

Тема 11.2. Виклик функції. Параметри у виклику та у визначенні

Тема 11.3. Глобальні та локальні змінні.

Тема 11.4. Робота з об'єктами типу `function`.

Тема 11.5. Необов'язкові параметри функції та їх зіставлення по ключах. Змінне число параметрів функції. Комбінування параметрів.

Тема 11.6. Складність функцій користувача

Лекція 17

Тема 11.7. Фіксоване та змінне число параметрів функції

Тема 11.8. Поєднання обов'язкових параметрів з зірковими

Тема 11.9. Використання словника для збереження іменованих параметрів

Тема 11.10. Передача параметрів тільки по іменах.

Тема 11.11. Анонімні функції `lambda`. Способи задавання та сфери використання анонімних функцій. Функції-генератори. Застосування методу `__next__` до функцій-генераторів.

Тема 11.12. Виклик функції-генератора з функції-генератора за допомогою ключового слова `yield from`.

Лекція 18

Тема 11.13. Декоратори функцій та їх застосування.

Тема 11.14. Рекурсія, обчислення факторіала.

Тема 11.15. Глобальні та локальні змінні. Видимість глобальних та локальних змінних.

Тема 11.16. Одержання словників глобальних та локальних ідентифікаторів за допомогою функцій `globals()` та `locals()`. Вкладені функції.

Тема 11.17. Анотації функцій. Атрибут об'єкта функції `__annotations__`

Розділ 12. Модулі і пакети

Лекція 19

Тема 12.1. Визначення модуля в Python. Головний модуль `"__main__"`. Атрибут об'єкта модуля `__main__`: `__name__`.

Тема 12.2. Інструкція `import` та варіанти її застосування для імпортування модулів. Імпорт кількох модулів однією інструкцією.

Тема 12.3. Модуль `time` та його функції.

Тема 12.4. Шляхи пошуку модулів, модифікація змінної `PYTHONPATH`, файли з розширенням `*.pth`

Тема 12.5. Означення пакета у мові Python. Порядок доступу до модулів пакета

Тема 12.6. Імпортування функцій з модулів, відносно імпортування модулів

Тема 12.7. Пакети `docutils` і `distutils`

Лекція 20

Тема 12.8. Установка нових пакетів у мові Python. Система керування пакетами `pip`

Тема 12.9. Використання пакетів `tabulate` та `NetworkX`

Тема 12.10. Основні функції пакета `NetworkX`

Тема 12.11. Генератори графів

Тема 12.12. Пакет `matplotlib` для відображення графіків

Розділ 13. Об'єктно-орієнтоване програмування

Лекція 21

Тема 13.1. Основні означення, що пов'язані з об'єктно-орієнтованим програмуванням. Особливості інкапсуляції, спадкування та поліморфізму для ООП в Python.

Тема 13.2. Визначення класу й створення екземпляра класу. Синтаксис створення атрибуту класу. Спосіб створення методу класу за допомогою інструкції `def`. Застосування змінної `self` для доступу до атрибутів та методів всередині класу.

Тема 13.3. Методи `__init__()` для задавання початкових значень атрибутам екземпляра класу.

Тема 13.4. Метод `__del__()`, як деструктор, що викликається при видаленні екземпляра класу.

Лекція 22

Тема 13.5. Системні атрибути та атрибути користувача. Атрибути екземпляру класу

Тема 13.6. Ідея інкапсуляції та її реалізація в Python за допомогою методів класу. Переавантаження операторів на прикладі методу `__str__`.

Тема 13.7. Спадкування. Синтаксис спадкування. Ієрархія спадкування.

Тема 13.8. Механізм перевизначення методів. Механізм перевизначення атрибутів

Тема 13.9. Поняття базового класу (суперкласу) та похідного класу (підкласу).

Тема 13.10. Множинне спадкування. Атрибут `__mro__`

Лекція 23

Тема 13.11. Особливості реалізації поліморфізму в Python. Переавантаження методів суперкласів їх підкласами. Домішки та їх використання.

Тема 13.12. Застосування функції `getattr()` до значення атрибута по його назві. Функція `setattr()` для задавання значення атрибута за його назвою. Видалення атрибута функцією `delattr()` та перевірка його значення функцією `hasattr()`. Принцип відкритості атрибутів в мові Python

(public). Спеціальні методи екземпляра класу: виклик екземпляра класу як функції (`__call__()`), метод `__getattr__` викликається при доступі до неіснуючого атрибута класу.

Тема 13.13. Метод `__getattribute__()`, який викликається при доступі до будь-якого атрибута класу, метод `__setattr__()`, який викликається при спробі присвоювання значення атрибуту екземпляра класу, метод `__delattr__()`,

Тема 13.14. Інструкція `del` та ряд інших методів, що викликаються при виконанні арифметичних та логічних операцій.

Розділ 14. Робота з файлами.

Лекція 24

Тема 14.1. Основні поняття про файл та типи файлів. Відкриття файлу та формат функції `open()`. Абсолютний та відносний шлях до файлу. Перетворення відносного шляху у абсолютний. Можливі задавання шляхів до файлу та їх модифікація.

Тема 14.2. Встановлення шляху до поточного каталогу. Одержання шляху до файлу, що виконується, за допомогою атрибута `__file__`. Одержання повного шляху до файлу.

Тема 14.3. Модифікатори відкриття файлу. Особливості роботи з механізмом буферизації файлів. Поняття про кодування текстових файлів та особливості роботи з різними кодуваннями.

Тема 14.4. Методи для роботи з файлами. Права доступу до файлів і каталогів. Визначення прав доступу. Функції для маніпулювання файлами. Перевірка наявності файлу, розміру файлу, часу останнього доступу, часу створення, часу останньої зміни. Функції для перетворення шляху до файлу. Перенаправлення вводу/виводу. Функція `print()`. Стандартний ввід і вивід.

Лекція 25

Тема 14.5. Функції модуля `os.path`.

Тема 14.6. Застосування модуля `pathlib`. Визначення параметрів файла.

Тема 14.7. Формування шляху та каталогу

Тема 14.8. Функції для роботи з файлом.

Лекція 26

Тема 14.9. Збереження об'єктів у файлі. Функції модуля `pickle` для роботи з файлами.

Тема 14.10. Класи `Pickler` і `Unpickler`.

Тема 14.11. Перетворення об'єкта у послідовність байтів і відновлення послідовності у об'єкт

Тема 14.12. Використання модуля `shelve` для зберігання даних в файлі по ключу. Функції для роботи з каталогами. Виключення, які виникають при виконанні операцій з файлами.

Тема 14.13. Застосування функції `glob()` до папок

Тема 14.14. Виключення, які виникають при файлових операціях

Розділ 15. Графічний інтерфейс користувача. Бібліотека `tkinter` (віджети).

Лекція 27

Тема 15.1. Підключення бібліотеки `tkinter`. Варіанти застосування інструкції `import` при роботі з `tkinter`. Створення головного вікна. Створення віджета. Задавання властивостей віджетів. Визначення подій та їх обробників. Розміщення віджета за допомогою методу `pack`. Відображення головного вікна.

Тема 15.2. Огляд віджетів, які реалізовані в бібліотеці `tkinter`. Кнопка (`Button`), лейбл (`Label`), днорядкове текстове поле (`Entry`),

Тема 15.3. Багаторядкове текстове поле (`Text`), методи роботи з віджетами

Лекція 28

Тема 15.4. Віджети `Listbox`, фрейм (`Frame`), галочка (`Checkbutton`), галочка з одним вибором (`Radiobutton`), шкала (`Scale`), скролбар (`Scrollbar`).

Тема 15.5. Пакувальник `pack()`. Пакування кількох віджетів одного над одним. Розміщення кількох віджетів поряд. Опції пакувальника `pack()`: `anchor`, `expand`, `fill`, `ipadx` та `ipady`, `padx` та `pady`, `side`.

Тема 15.6.

Пакувальник `grid()`. Аргументи пакувальника `grid`: `row`, `rowspan`, `column`, `columnspan`, `padx` / `pady`, `ipadx` / `ipady`, `sticky`, `in_`.

Лекція 29

Тема 15.7. Додаткові функції пакувальника `grid()`. Методи налаштування комірок.

Використання опцій злиття стовпців або рядків

Тема 15.8. Пакувальник `place`. Синтаксис пакувальника `place` та його опції: `anchor`, `bordermode`, `height`, `width`, `relheight`, `relwidth`, `relx`, `rely`, `x`, `y`.

Тема 15.9. Віджети меню: меню верхнього рівня (`toplevel`), впливаюче вікно (`pop-up`), та випадаюче меню (`pull-down`). Просунуті віджети типу `OptionMenu`. Синтаксис та властивості меню. Методи доступу на об'єктах меню. Кнопка меню (`Menubutton`), синтаксис та властивості.

Лекція 30

Тема 15.10. Застосування меню типу `Toplevel`: синтаксис, властивості та методи.

Тема 15.11. Віджет, що задає тло для малювання (`Canvas`): синтаксис, властивості та методи. Прийоми малювання на `Canvas`.

Тема 15.12. Віджет-контейнер (`PaneWindow`): синтаксис, властивості та методи. Модуль `messagebox` та його застосування для відображення вікон повідомлень у прикладних програмах. Події та зв'язування. Розгляд об'єкта `Event` та його атрибутів: `widget`, `x`, `y`, `x_root`, `y_root`, `char`, `keysym`, `keycode`, `num`, `width`, `height`, `type`. Зв'язування віджета з подіями за допомогою `bind()`. Представлення подій та основні види подій. Формати подій. Екземпляри класів та зв'язування класів. Методи зв'язування класів. Інші `callback`-подібні методи. Протоколи та їх застосування.

4. Навчальні матеріали та ресурси

4.1. Базова література

1. Новотарський, М. А. 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 ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 17.93 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 701с. <https://ela.kpi.ua/handle/123456789/49913>
2. Яковенко А.В. 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. <https://ela.kpi.ua/handle/123456789/25111>
3. Анісімов А. В. та ін. Програмування числових методів мовою Python : підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с. https://ist.kpi.ua/wp-content/uploads/2017/05/%D0%94%D0%BE%D1%80%D0%BE%D0%B3%D0%B8%D0%B9_%D0%9F%D1%96%D0%B4%D1%80%D1%83%D1%87%D0%BD%D0%B8%D0%BA.pdf
4. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП

Баликіна С.М., 2020. 180 с.

<https://epub.chnpu.edu.ua/jspui/bitstream/123456789/5584/1/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%BC%D0%BE%D0%B2%D0%BE%D1%8E%20Python.pdf>

5. Невлюдов І. Ш. Проектування мобільних робіт на базі одноплатних комп'ютерів (Raspberry Pi) і мови Python 3.6 : підручник / І. Ш. Невлюдов, А. О. Андрусевич, В. В. Євсєєв – Харків, 2020. - 257 с. <https://openarchive.nure.ua/handle/document/20558>

4.2. Додаткова література

1. Phillips D. Python 3 Object-oriented Programming. – Packt Publishing, 2015. – 460 p.
2. Christopher A. Jones, Fred L. Drake Jr. Python & XML. – O'Reilly Media, 2001. – 450 p.
3. Chun W.J. Core Python Programming, Second Edition. – Prentice Hall, 2006. – 1120 p.
4. Rappin N., Dunn R. wxPython in Action. – Manning Publications Co., 2006. – 552 p.
5. Summerfield M. Rapid GUI Programming with Python and Qt. – Prentice Hall, 2007. – 628 p.
6. Alchin M. Pro Python. – Apress, 2010. – 341 p.
7. Foundations of Python Network Programming: The comprehensive guide to building network applications with Python (Books for Professionals by Professionals) 2nd ed. Edition by John Goerzen (Author), Tim Bower (Author), Brandon Rhodes (Author).

4.3. Інформаційні ресурси:

1. Новотарський М. А. Відео лекції з курсу «Програмування. Частина 1»: https://www.youtube.com/@KPI_OT/courses
2. Новотарський М. А. Лабораторний з курсу курсу «Програмування»// Методичні вказівки, <https://cloud.comsys.kpi.ua/s/dxWmzzMTdo66NGc>
3. Установка Python: <https://www.python.org>
4. Установка Anaconda: <https://www.anaconda.com/download>
5. Установка PyCharm: <https://www.jetbrains.com/pycharm/download/?section=windows>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Розподіл навчального часу за видами занять і завдань з дисципліни згідно з робочим навчальним планом: на кредитний модуль виділено 180 годин та 6 кредитів – 30 лекцій (60 години), 8 лабораторних робіт (30 годин), самостійна робота (90 годин), підготовка до поточних тестувань та поточні тестування (20 годин). Підготовка до залікового тесту та заліковий тест (10 годин).

Лекційні заняття

Для досягнення мети навчальної дисципліни слід зосередитись на ключових аспектах методології оволодіння скриптовою алгоритмічною мовою програмування. Особливу увагу варто приділити застосуванню методів та функцій для типів даних мови програмування Python, особливостям застосування об'єктно-орієнтованого стилю програмування та принципів використання файлової системи. Це забезпечить студентам всебічну підготовку до розробки застосунків мовою Python.

Деталізація тем лекційних занять, конспект лекцій та відеоматеріали розміщено на платформі дистанційного навчання «Сікорській»

<https://classroom.google.com/c/NDE3ODIzNzU2MDk1?cjc=nnenp23>

Лабораторний практикум побудований за принципом поступового ускладнення: від програмування простих алгоритмів з використанням базових типів даних до створення повноцінних застосунків, що включають використання елементів графічного інтерфейсу користувача, побудованого з використанням бібліотеки tkinter. Такий підхід сприяє кращому засвоєнню матеріалу та формуванню компетентностей, необхідних для професійної діяльності у сфері інформаційних технологій. Мета лабораторних робіт — набуття практичних навичок у розробці програмних застосунків алгоритмічною мовою Python:

- вивчення основних складових частин персональних та професійних комп'ютерів, рівнів програмного забезпечення сучасних комп'ютерів. Інсталяція середовища розробки PyCharm & Anaconda.
- вивчення типів даних, які використовуються в мові програмування Python. Змінні та правила їх іменування, операції над змінними. Оператори та їх застосування.
- вивчення способів створення рядків та даних типу bytes і bytearray, операції над ними. Форматування рядків. Функції та методи роботи з рядками. Налаштування локалі.
- вивчення способів створення списків, кортежів, множин та задавання діапазонів. Операції над списками, кортежами та діапазонами. Функції для перетворень списків, кортежів та множин.
- вивчення способів створення словників та функцій користувача. Операції над словниками. Методи для роботи зі словниками. Генератори словників Функції користувача, особливості їх створення та використання.
- вивчення способів створення та підключення модулів та пакетів. Основи ООП. Методи і атрибути класів та робота з ними. Побудова програми у стилі ООП.
- вивчення основних способів роботи з виключеннями. Виключення користувача. Відкриття файлів, зчитування та запис у файл. Шляхи доступу до файлів. Функції, методи та атрибути для роботи з файлами.
- ознайомлення з організацією графічного інтерфейсу на основі бібліотеки tkinter. Графічний інтерфейс (GUI) та його елементи. Модуль tkinter.

Тематика лабораторних робіт

Теми лабораторних робіт та кількість годин на їх перевірку для однієї навчальної групи наведено в таблиці 1.

Таблиця 1.

№ з/п	Назва лабораторної роботи	Кількість ауд.годин
1	Лабораторна робота №1. Загальні поняття про технічні характеристики та програмне забезпечення сучасних комп'ютерів .	4
2	Лабораторна робота №2. Базові типи даних, змінні та оператори мови програмування Python .	6
3	Лабораторна робота №3. Робота з даними типу str, bytes та bytearray .	4
4	Лабораторна робота №4. Списки, кортежі, множини і діапазони в Python.	4
5	Лабораторна робота №5. Словники та функції користувача (Тема 10, Тема 11).	6
6	Лабораторна робота №6. Об'єктно-орієнтоване програмування на Python. Класи та екземпляри класів (Тема 13).	6
7	Лабораторна робота №7. Робота з файлами на Python. Запис у файл, зчитування з файлу, модифікація.	6
8	Лабораторна робота №8. Розробка програми з графічним інтерфейсом на основі бібліотеки tkinter .	Бонусна робота

6. Самостійна робота здобувача вищої освіти денної форми навчання

Види самостійної роботи для студентів денної форми навчання (90 годин):

- підготовка до аудиторних занять, виконання поточних домашніх завдань та опрацювання матеріалів лекцій (0,5 годин x 30 лекцій = 15 годин);
- виконання індивідуального завдання до лабораторних робіт, розв'язок задач, оформлення протоколу, підготовка та оброблення проведення розрахунків за первинними даними, отриманими на лабораторних заняттях, оформлення звіту до лабораторної роботи (2 години x 7 лаб = 14 годин);
- підготовка до поточного закритого тесту (2 години x 5=10)
- підготовка до екзамену (30 годин);
- самостійне опрацювання теоретичного матеріалу (21 годин).

Теми для самостійного опрацювання

Розділ 6. Форматування рядків. Функції та методи для роботи з рядками.

Тема 6.3. Літеральна інтерполяція рядків (f-рядки) (2 год)

Тема 6.4. Шаблонізація (Templating) (1 год)

Розділ 8. Списки, кортежі, множини і діапазони в мові Python

Тема 8.2. Особливості створення списків та їх копіювання (2 год)

Розділ 9. Означення, властивості та застосування кортежів множин та діапазонів. Ітератори.

Тема 9.3. Методи для роботи з множинами. Генератори множин та їх синтаксис. (2 год)

Розділ 11. Визначення та властивості функцій користувача в Python

Тема 11.11. Функції-генератори. (1 год)

Тема 11.12. Виклик функції-генератора з функції-генератора за допомогою ключового слова yield from. (1 год)

Тема 11.13. Декоратори функцій та їх застосування. (2 год)

Розділ 12. Модулі і пакети

Тема 12.4. Шляхи пошуку модулів, модифікація змінної PYTHONPATH, файли з розширенням *.pth (2 год)

Тема 12.7. Пакети docutils і distutils (2 год)

Тема 12.10. Основні функції пакета NetworkX (2 год)

Розділ 13. Об'єктно-орієнтоване програмування

Тема 13.7. Спадкування. Синтаксис спадкування. Ієрархія спадкування. (2 год)

Розділ 15. Графічний інтерфейс користувача. Бібліотека tkinter (віджети).

Тема 15.9. Віджети меню: меню верхнього рівня (toplevel), випливаюче вікно (pop-up), та випадаюче меню (pull-down). росунуті віджети типу OptionMenu. Синтаксис та властивості меню. Методи доступу на об'єктах меню. Кнопка меню (Menubutton), синтаксис та властивості. (2 год)

Матеріали для самостійного опрацювання, теми, питання для самоконтролю та контрольні запитання знаходяться на платформі дистанційного навчання «Сікорський» (Classroom).

<https://classroom.google.com/c/NDE3ODIzNzU2MDk1?cjc=nnenp23>

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Дедлайни. Для виконання практичних робіт встановлюються дедлайни. Дедлайни уточнюються і оприлюднюються викладачем на початку семестру.

Заохочувальні бали (бонуси) виставляються за: активну участь на лекціях, виконання додаткової лабораторної роботи. Кількість заохочуваних балів встановлює викладач, але не більше 10.

Лабораторні роботи здаються особисто з перевіркою теоретичних знань, які необхідні для виконання лабораторної роботи. Перевірка практичних результатів включає перевірку коду, виконання тестових завдань та відповідей на запитання по темі лабораторної роботи.

В процесі навчання викладач має право нарахувати заохочувальний бал за дострокове виконання лабораторної роботи, за проявлений творчий підхід при виконанні індивідуального завдання або за активну участь у обговоренні питань, що пов'язані з тематикою лекції або практичного заняття.

Правила відвідування занять. Відвідування занять є вільним, бали за присутність на лекціях та практичних заняттях не додаються, відповідно штрафні бали не передбачаються. Втім, вагома частина рейтингу формується через активну участь у заходах на практичних заняттях.

Дистанційний режим навчання. У разі запровадження обмежень на відвідування університету, пов'язаних з введенням карантину або режиму воєнного стану в державі, освітній процес здійснюється у дистанційному режимі відповідно до Положення про дистанційне навчання в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/index.php/node/188>), Регламенту організації освітнього процесу в дистанційному режимі (<https://profkom.kpi.ua/reglament-organizatsiyi-osvitnogo-protsesu-v-distantsiynomu-rezhimi>) та Регламенту проведення семестрового контролю в дистанційному режимі (<https://osvita.kpi.ua/node/148>). У режимі дистанційного навчання заняття відбуваються у вигляді онлайн-конференції на платформах BigBlueButton, Google Meet, Zoom. Посилання на конференцію видається на початку семестру і розміщується в АС «Електронний кампус». З метою забезпечення якісної підготовки здобувачів, дистанційний курс дисципліни розміщено на Платформа дистанційного навчання «Сікорський» (<https://www.sikorsky-distance.org>). Результати оцінювання висвітлюють у АС «Електронний кампус» на особистій сторінці здобувача (<https://ecampus.kpi.ua>).

Правила поведінки на заняттях. На заняттях слід дотримуватись норм етичної поведінки визначених у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>). На території університету здобувачі мають поводити себе відповідно до Правил внутрішнього розпорядку (<https://kpi.ua/admin-rule>).

Визнання результатів навчання, набутих у неформальній/інформальній освіті. Положення про визнання результатів навчання, набутих у неформальній / інформальній освіті (<https://osvita.kpi.ua/index.php/node/179>) регламентує визнання результатів навчання, набутих у неформальній/інформальній освіті:

- у разі навчання на неформальній освіті, яку студент обрав самостійно і хоче зарахувати результати навчання в рамках дисципліни, проходиться процедура валідації, що передбачає подання здобувачем заяви на ім'я декана, декларації підтверджувальних документів; рішення про визнання чи не визнання приймається комісією у складі завідувача кафедри, викладача, гаранта освітньо-професійної програми.

- за рішенням викладача окремі змістовні модулі або теми дисципліни можуть бути зараховані на підставі результатів проходження здобувачем онлайн-курсу або іншого елементу неформальної освіти; перелік завдань та елементів неформальної освіти визначається викладачем заздалегідь і розміщується на платформі дистанційного навчання «Сікорський», про що студенти повідомляються на початку семестру; у цьому випадку здобувач звільняється від виконання відповідних завдань і отримує максимальний бал згідно з рейтинговою системою оцінювання.

Політика використання штучного інтелекту. Використання штучного інтелекту регламентується «Політикою використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського» (<https://osvita.kpi.ua/node/1225>). Усі завдання, як під час виконання навчальних завдань з дисципліни, так і індивідуальні завдання, мають бути результатом власної оригінальної роботи здобувача. Використання ШІ для автоматичної генерації відповідей без подальшого їх аналізу та доопрацювання заборонено.

Порушення академічної доброчесності: використання плагіату. У разі виявлення фактів використання плагіату під час виконання навчальних завдань, студент втрачає право на подальше виконання поточних завдань у межах навчальної дисципліни до з'ясування обставин порушення.

Студент зобов'язаний аргументовано пояснити причини використання плагіату, а також засвідчити готовність дотримуватись принципів академічної доброчесності в подальшому навчанні (письмово у вигляді пояснювальної записки); повторно виконати завдання або, за потреби,

виконати додаткове завдання відповідно до бачення викладача з урахуванням вимог щодо самостійності.

Зазначені дії регламентуються Кодексом честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>).

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

Види контролю з навчальної дисципліни «Програмування» включають:

Лабораторні роботи:

Заплановано самостійне виконання семи лабораторних робіт.

Теми лабораторних робіт узгоджені у часі та за змістом з темами лекцій. Виконання лабораторних робіт у повному обсязі дозволяє набути практичних навичок програмування алгоритмічною мовою Python та оволодіти сучасними технологіями програмування.

Поточний контроль:

Передбачено 5 поточних тестувань закритими тестами у системі TCXAM, які повністю охоплюють тематику даної навчальної дисципліни. Кожний поточний закритий тест містить 10 питань та триває 10 хв. У випадку дистанційного навчання закритий поточний тест проводиться у терміни, які визначаються викладачем. При очному навчанні час чергового поточного тестування призначається викладачем за узгодженням зі студентами.

Екзамен проводиться у вигляді екзаменаційного тесту у системі TCXAM.

Тест проводиться в екзаменаційну сесію, складається з 40 питань.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою, яка наведена у таблиці 3

Таблиця 3.

Відповідність рейтингових балів

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

Оскільки навчальна дисципліна «Програмування» має семестрову атестацію у вигляді екзамену, рейтингова система оцінювання побудована за типом PCO – 2. Семестровий рейтинг студента складається з балів, які він отримує за види робіт відповідно до таблиці 4.

Таблиця 4

Оцінювання окремих видів навчальної роботи студента (у балах)

Вид навчальної роботи	Форма контролю	Кількість балів	Допуск до семестрового контролю	Всього балів
Лабораторна робота № 1	Звіт	4	3	10
	Опитування	3		
	Сирцевий код	3		
Лабораторна робота № 2	Звіт	4	3	10
	Опитування	3		

Вид навчальної роботи	Форма контролю	Кількість балів	Допуск до семестрового контролю	Всього балів
	Сирцевий код	3		
Лабораторна робота № 3	Звіт	4	3	10
	Опитування	3		
	Сирцевий код	3		
Лабораторна робота № 4	Звіт	4	3	10
	Опитування	3		
	Сирцевий код	3		
Лабораторна робота № 5	Звіт	4	3	10
	Опитування	3		
	Сирцевий код	3		
Лабораторна робота № 6	Звіт	4	3	10
	Опитування	3		
	Сирцевий код	3		
Лабораторна робота № 7	Звіт	4	3	10
	Опитування	3		
	Сирцевий код	3		
Результати поточного тестування (5 тестувань x 6 балів)	Поточний тест 1	6	9	30
	Поточний тест 2	6		
	Поточний тест 3	6		
	Поточний тест 4	6		
	Поточний тест 5	6		
Rп			30	100
Визначення семестрового рейтингу за формулою: $R_c = R_p \times 0.6 = 100 \times 0.6$ Rc			18	60
Екзамен (Re)			12	40
Усього за семестр (R=Rc+Re)			60	100

Індивідуальний поточний рейтинг студента (**Rп**) складається з балів, які він отримує за:

1) Поточний контроль виконується у системі TCEXAM.

2) Виконання лабораторних робіт. Протягом семестру студенти виконують 7 лабораторних робіт. Максимальна кількість балів за кожну лабораторну роботу – 10. Бали нараховуються за правилом

- Звіт – 4 бали,
- Опитування – 3 бали
- Сирцевий код – 3 бали

Максимальний можливий бал за лабораторну роботу – 10 балів.

Максимальна кількість балів за всі лабораторні роботи $7 \times 10 = 70$ балів.

Розрахунок розміру шкали (R) рейтингу.

Сума вагових балів контрольних заходів протягом семестру становить:

$R = R_c + R_e$, де R_c – семестровий рейтинг студента (контрольні, лабораторні роботи).

Оцінювання поточних тестувань:

$5 \times 6 = 30$ балів

Обчислення семестрового рейтингу відбувається на основі поточного рейтингу R_p за

формулою $R_c = \frac{R_p}{100} \cdot 60\%$. Максимальне значення семестрового рейтингу $R_c = R_p \times 0.6 = 100 \times 0.6 = 60$ балів;

Екзамен проводиться у вигляді тесту у системі TCXAM. Тест складається з 40 питань.

R_e – ваговий бал екзамену, $R_e = 40$ балів.

Розмір рейтингової шкали для навчальної дисципліни становить:

$R = R_c + R_e = 60 + 40 = 100$ балів.

Необхідною умовою допуску студента до екзамену є його індивідуальний семестровий рейтинг (R_c), не менший, ніж 30 балів, та відсутність заборгованості з лабораторних робіт. При невиконанні згаданих вимог студент до екзамену не допускається.

Порядок ліквідації студентами академічної заборгованості

Для студентів, що не виконали вчасно індивідуальний навчальний план деканатом призначається додаткова сесія і надається можливість двох перескладань. Час першого та другого перескладань додатково узгоджується. Для студентів, яким призначена додаткова сесія поточний рейтинг анулюється і призначаються такі умови:

Для отримання позитивного перескладання студент має:

1. виконати всі лабораторні роботи,
2. пройти семестровий контроль і надати більше 60% правильних відповідей.

Залікова оцінка, яка отримана у додаткову сесію, не може перевищувати «достатньо»

9. Додаткова інформація з дисципліни (освітнього компонента)

Викладання дисципліни «Програмування» для напряму «Комп'ютерна інженерія» має свою специфіку, яка пов'язана з тим, що сфера застосування методів машинного аналізу даних постійно розширюється. Разом з тим, мова програмування Python, яка вивчається в рамках даної навчальної дисципліни, є основною алгоритмічною мовою, яку використовують в таких задачах. Вибір алгоритмічної мови програмування Python, як першої мови у рамках спеціальності 123 – «Комп'ютерна інженерія» обґрунтовано також тим, що дана мова має низький поріг входження, а також протягом кількох останніх років займає перші місця у рейтингу з популярності алгоритмічних мов програмування.

Робочу програму навчальної дисципліни (силабус):

Складено: д.т.н. проф. Новотарський Михайло Анатолійович

Ухвалено кафедрою ОТ (протокол №18 від 24.06.2026)

Погоджено: Методичною комісією ФІОТ (протокол № 12 від 26.06.2026)