МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 3

з дисципліни «Об'єктно-орієнтоване проектування СУ»

Тема: "Структурування програм з використанням функцій"

ХАІ.301. 173. 3-91АВ/С. 2 ЛР

Виконав студент гр. _____3-91АВ/С_____

_____Опанасюк Олександр___
(підпис, дата) (П.І.Б.)

Перевірив
_____ к.т.н., доц. О. В. Гавриленко
(підпис, дата) (П.І.Б.)

2024

МЕТА РОБОТИ

Вивчити теоретичний матеріал щодо синтаксису на мові Python і поданням у вигляді UML діаграм діяльності алгоритмів з розгалуження та циклами, а також навчитися використовувати функції, інструкції умовного переходу і циклів для реалізації інженерних обчислень.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Описати функцію відповідно до варіанту. Для виклику функції (друга частина задачі) описати іншу функцію, що на вході має список вхідних даних і повертає список вихідних даних. Введення даних, виклик функції та виведення результатів реалізувати в третій функції без параметрів. Завдання наведено в табл.1.

Завдання 2. Розробити дві вкладені функції для вирішення задачі обробки двовимірних масивів відповідно до варіанту: зовнішня – без параметрів, внутрішня має на вході ім’я файлу з даними, на виході – підраховані параметри матриці (перша частина задачі) та перетворену матрицю (друга частина задачі). Для обробки масивів використати функції бібліотеки numpy. Завдання представлено в табл.2.

ВИКОНАННЯ РОБОТИ

Завдання 1. Обчислення третьої ступені чисел

Вхідні дані:

A – число для піднесення до третьої ступені; тип: float; допустимі значення: будь-яке дійсне число.

in_lst – список із 5 чисел; тип: list[float]; допустимі значення: список із 5 елементів, кожен з яких є дійсним числом.

Вихідні дані:

B – третя ступінь числа A; тип: float.

out_data – список із третіх ступенів чисел із вхідного списку in_lst; тип: list[float].

Текстове повідомлення: "Числа, піднесені до кубу: [список результатів]".

Алгоритм вирішення показано нижче:

- Введення користувачем п’яти чисел.
- Виклик функції PowerA3 для обчислення третьої ступені кожного числа у списку.
- Повернення результату у вигляді списку третіх ступенів.
- Виведення результату на екран.

```
4   def task1():
5   # Взаємодія зі завданням 1
6       in_data = []
7       print("Введіть 5 чисел для обчислення їхніх третіх ступенів:")
8       for _ in range(5):
9           num = float(input("Число: "))
10          in_data.append(num)
11
12      out_data = Power_of5(in_data)
13      print("Числа, піднесені до кубу:", out_data)
```

```
4   # Завдання 1. Функція PowerA3
5   def PowerA3(A):
6       return A ** 3
7
8
9   def Power_of5(in_lst):
10      return [PowerA3(A) for A in in_lst]
```

Рисунок 1 – Алгоритм вирішення завдання 1

Завдання 2. Обробка матриці

Вхідні дані:

filename – ім'я текстового файлу з матрицею; тип: str; допустимі значення: існуючий текстовий файл із числовими даними.

K – номер рядка для обробки; тип: int; допустимі значення: $1 \leq K \leq M$, де M – кількість рядків у матриці.

matrix – матриця розміру $M \times N$; тип: numpy.ndarray; допустимі значення: будь-яка числова матриця.

Вихідні дані:

row_sum – сума елементів K-го рядка матриці; тип: float.

row_product – добуток елементів K-го рядка матриці; тип: float.

matrix_diff – різниця матриці та одиничної матриці того ж розміру; тип: numpy.ndarray.

Текстові повідомлення:

"Сума елементів K-го рядка: [значення]".

"Добуток елементів K-го рядка: [значення]".

"Різниця матриці з одиничною матрицею:\n[матриця]".

Алгоритм вирішення показано нижче:

- Введення користувачем імені файлу та номера рядка K.
- Читання матриці з текстового файлу.
- Перевірка коректності номера рядка K.
- Обчислення:
- Суми та добутку елементів K-го рядка;
- Різниці матриці з одиничною матрицею.
- Виведення результатів на екран.

```python
# Завдання 2. Функції для роботи з матрицею
def process_matrix(filename, K):
    # Читаємо матрицю з файлу
    matrix = np.loadtxt(filename, dtype=float)

    # Перевірка коректності K
    if K < 1 or K > matrix.shape[0]:
        raise ValueError("K повинен бути в межах 1 ≤ K ≤ M")

    # Обчислення параметрів K-го рядка
    row = matrix[K - 1]  # K-й рядок (нумерація з 0)
    row_sum = np.sum(row)
    row_product = np.prod(row)

    # Обчислення різниці матриці з одиничною
    identity_matrix = np.eye(matrix.shape[0], matrix.shape[1])
    matrix_diff = matrix - identity_matrix

    return row_sum, row_product, matrix_diff


def matrix_task():
    filename = input("Введіть ім'я файлу з матрицею: ")
    K = int(input("Введіть номер рядка K (1 ≤ K ≤ M): "))
    try:
        row_sum, row_product, matrix_diff = process_matrix(filename, K)
        print(f"Сума елементів K-го рядка: {row_sum}")
        print(f"Добуток елементів K-го рядка: {row_product}")
        print("Різниця матриці з одиничною матрицею:")
        print(matrix_diff)
    except Exception as e:
        print(f"Помилка: {e}")
```

Рисунок 2 – Алгоритм обробки матриці

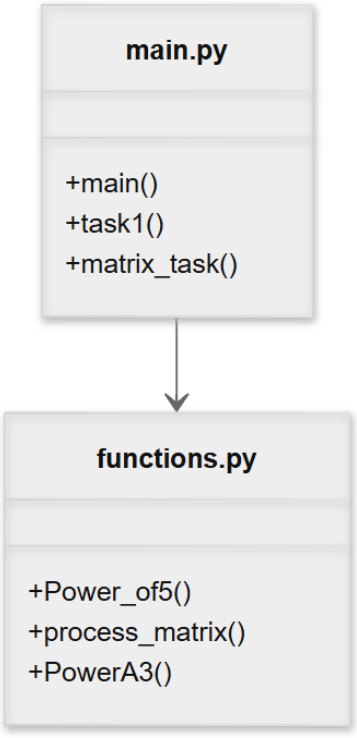

Рисунок 3 – Діаграма ієрархічної структури виклику функцій

ВИСНОВКИ

Було вивчено та закріплено на практиці основи роботи з функціями в Python, включаючи обчислення математичних операцій та обробку даних із текстових файлів. У програмі відпрацьовано обчислення третіх ступенів чисел та базові операції з матрицями, такі як підрахунок характеристик рядків і виконання матричних перетворень. Отримано навички структурованого програмування, зокрема розділення коду на функціональні модулі та основний скрипт.

Додаток А

Лістинг коду програми до задач №1: Обчислення кількості повних метрів, №2: Обчислення математичного виразу, №3: Перевірка позитивності числа

main.py
<
```
from functions import Power_of5, matrix_task


def task1():
# Взаємодія зі завданням 1
    in_data = []
    print("Введіть 5 чисел для обчислення їхніх третіх ступенів:")
    for _ in range(5):
        num = float(input("Число: "))
        in_data.append(num)

    out_data = Power_of5(in_data)
    print("Числа, піднесені до кубу:", out_data)


def main():
# Меню для вибору завдання
    while True:
        print("\nОберіть завдання:")
        print("1. Обчислення третьої ступені чисел (Завдання 1)")
        print("2. Обробка матриці (Завдання 2)")
        print("3. Вихід")

        choice = input("Ваш вибір: ")
        if choice == "1":
            task1()
        elif choice == "2":
            matrix_task()
        elif choice == "3":
            print("До побачення!")
            break
        else:
            print("Невірний вибір. Спробуйте ще раз.")


if __name__ == "__main__":
    main()
```
>

functions.py

<
```
import numpy as np


# Завдання 1. Функція PowerA3
def PowerA3(A):
    return A ** 3


def Power_of5(in_lst):
    return [PowerA3(A) for A in in_lst]


# Завдання 2. Функції для роботи з матрицею
def process_matrix(filename, K):
    # Читаємо матрицю з файлу
    matrix = np.loadtxt(filename, dtype=float)

    # Перевірка коректності K
    if K < 1 or K > matrix.shape[0]:
        raise ValueError("K повинен бути в межах 1 ≤ K ≤ M")

    # Обчислення параметрів K-го рядка
    row = matrix[K - 1]  # K-й рядок (нумерація з 0)
    row_sum = np.sum(row)
    row_product = np.prod(row)

    # Обчислення різниці матриці з одиничною
    identity_matrix = np.eye(matrix.shape[0], matrix.shape[1])
    matrix_diff = matrix - identity_matrix

    return row_sum, row_product, matrix_diff


def matrix_task():
    filename = input("Введіть ім'я файлу з матрицею: ")
    K = int(input("Введіть номер рядка K (1 ≤ K ≤ M): "))
    try:
        row_sum, row_product, matrix_diff = process_matrix(filename, K)
        print(f"Сума елементів K-го рядка: {row_sum}")
        print(f"Добуток елементів K-го рядка: {row_product}")
        print("Різниця матриці з одиничною матрицею:")
        print(matrix_diff)
    except Exception as e:
        print(f"Помилка: {e}")
```
>

ДОДАТОК Б
Скрін-шоти вікна виконання програми

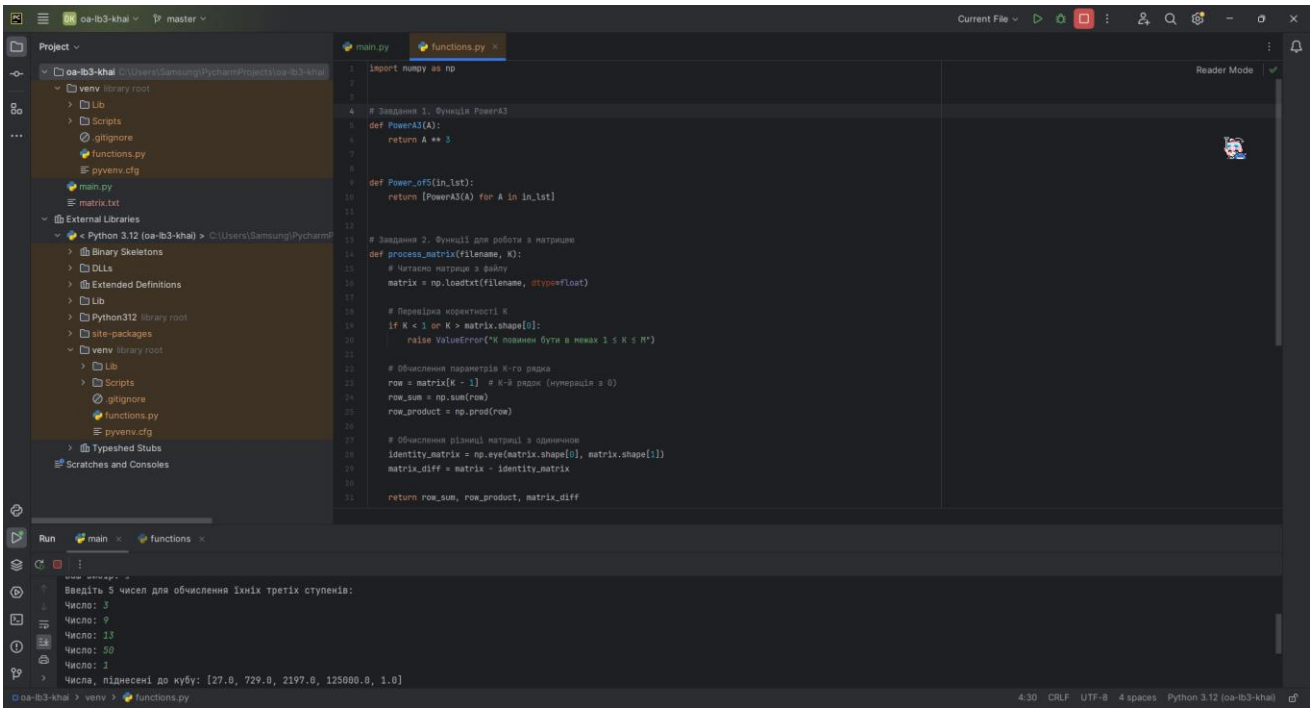

Рисунок Б.1 – Екран виконання програми для вирішення завдання №1. Обчислення третьої ступені чисел.

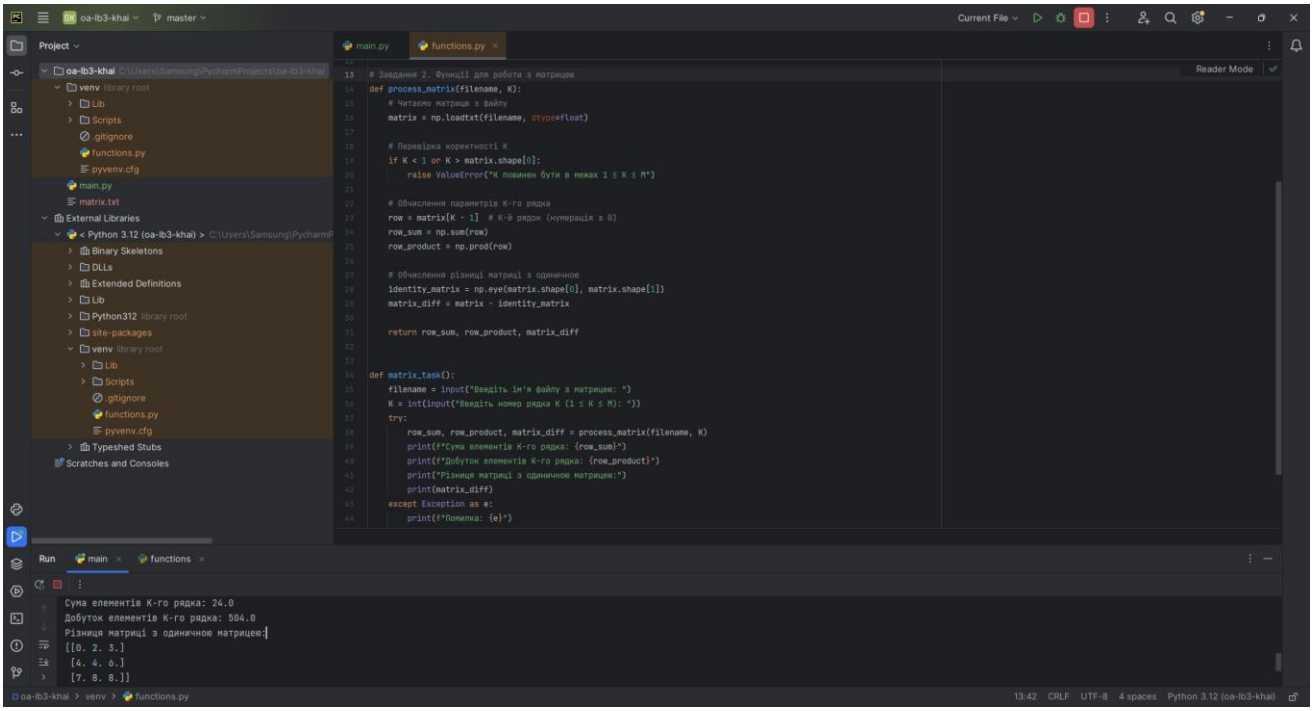

Рисунок Б.2 – Екран виконання програми для вирішення завдання №2. Обробка двовимірної матриці.