

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 програмування СУ»

Тема: «Структурування програм з використанням функцій»

ХАІ.301 .173. 310ст.3 ЛР

Виконав студент гр. _____310ст_____

_____Возвишаєв Олексій Андрійович
(підпис, дата) (П.І.Б.)

Перевірів

_____к.т.н., доц. О. В. Гавриленко
_____ас. В. О. Білозерський
(підпис, дата) (П.І.Б.)

2024

МЕТА РОБОТИ

Вивчити теоретичний матеріал із синтаксису визначення і виклику функцій та особливостей послідовностей у Python, а також документацію бібліотеки `numpy`; отримати навички реалізації бібліотеки функцій з параметрами, що структурують вирішення завдань «згори – до низу»

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. (Proc) 23 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 2. Matrix 3 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`.

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі (Proc) 23

Вхідні дані:

Ім'я	Опис	Тип	Обмеження
a	Число, яке потрібно піднести до третього ступеня	float	Дійсне число
in_list	Список чисел, які потрібно піднести до третього ступеня	list[float]	Довжина списку – рівно 5 елементів

Вихідні дані:

Ім'я	Опис	Тип
result	Число, піднесене до третього ступеня	float
out_data	Список чисел, піднесених до третього ступеня	list[float]

Алгоритм вирішення показано на рис. 1

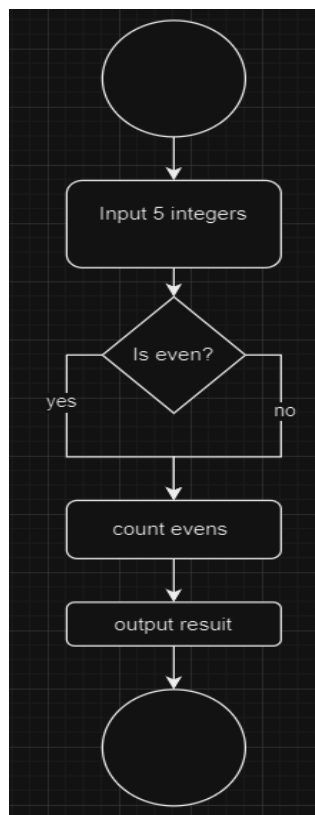


Рисунок 1 - Алгоритм вирішення завдання 1

Лістинг коду вирішення задачі наведено в дод. А стор.6. Екран роботи програми показаний на рис. Б.8.

Завдання 2. Вирішення задачі Matrix 3

Вхідні дані:

Ім'я	Опис	Тип	Обмеження
matrix	Початкова матриця	np.ndarray	Двовимірний масив цілих чисел
filename	Шлях до файлу з матрицею	str	Існуючий файл з коректними даними
rows	Кількість рядків матриці	int	Дійсне ціле число
cols	Кількість стовпців матриці	int	Дійсне ціле число
choice	Метод введення матриці (1 або 2)	str	Значення "1" або "2"

Вихідні дані:

Ім'я	Опис	Тип
row_sums	Суми елементів кожного рядка матриці	np.ndarray
row_products	Добутки елементів кожного рядка матриці	np.ndarray
result_matrix	Різниця між початковою і випадковою матрицями	np.ndarray

Алгоритм вирішення показано на рис. 1.2

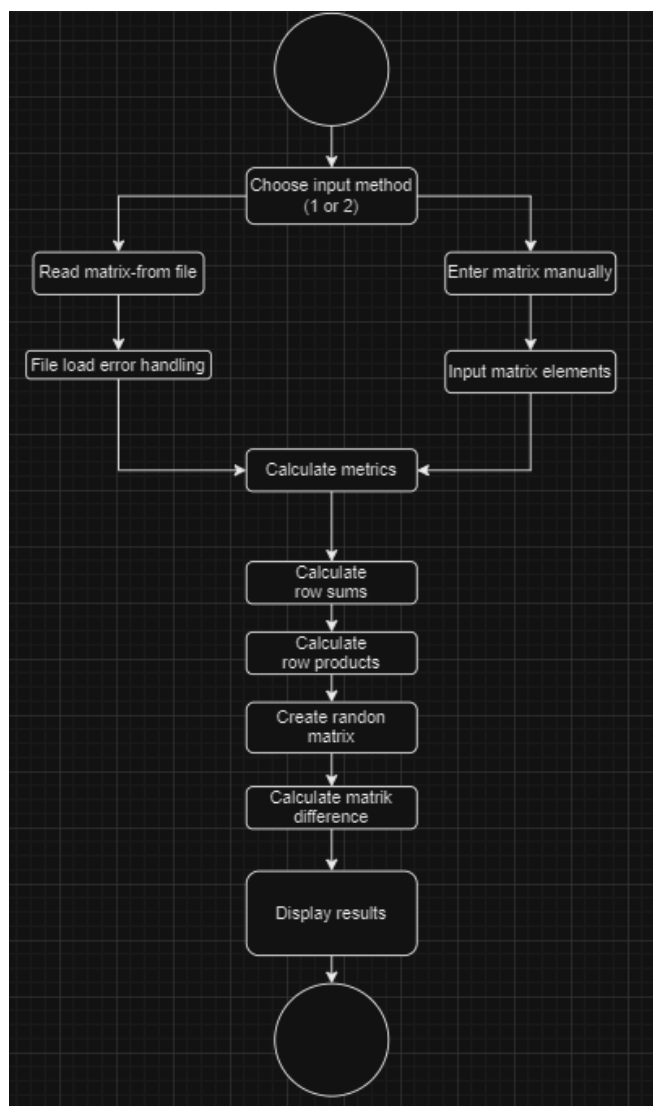


Рисунок 1.2 – Алгоритм вирішення завдання 2

Лістинг коду вирішення задачі наведено в дод. А стор.6. Екран роботи програми показаний на рис. Б.8.

ВИСНОВКИ

У ході виконання лабораторної роботи було вивчено принципи структурування програм з використанням функцій та особливості роботи з двовимірними масивами в Python за допомогою бібліотеки NumPy. Закріплено навички реалізації функцій для виконання окремих завдань, таких як обчислення суми та добутку елементів рядків матриці та їхньої різниці з випадковою матрицею.

ДОДАТОК А

Лістинг коду програми до задач (Proc) 23

```

def Even(K: int) -> bool:
    return K % 2 == 0

# Функція, яка приймає список чисел і повертає кількість парних
def count_even_numbers(numbers: list) -> int:
    return sum(1 for num in numbers if Even(num))

# Функція для введення даних, виклику попередніх функцій і виведення результатів
def main():
    # Введення 5 цілих чисел від користувача
    numbers = []
    for i in range(5):
        num = int(input(f"Введіть число {i + 1}: "))
        numbers.append(num)

    # Виклик функції для підрахунку парних чисел
    even_count = count_even_numbers(numbers)

    # Виведення результату
    print(f"Кількість парних чисел у наборі: {even_count}")

# Виклик основної функції
main()

```

Лістинг коду програми до задачі (Matrix 3)

```

import numpy as np

def calculate_matrix(matrix):
    # Обчислюємо суму елементів по рядках
    row_sums = np.sum(matrix, axis=1)
    # Обчислюємо добуток елементів по рядках
    row_products = np.prod(matrix, axis=1)

    print("Суми елементів рядків:", row_sums)
    print("Добутки елементів рядків:", row_products)

    # Створюємо випадкову матрицю того ж розміру
    random_matrix = np.random.randint(1, 10, size=matrix.shape)

    # Обчислюємо різницю між початковою і випадковою матрицями
    result_matrix = matrix - random_matrix
    print("Різниця початкової і випадкової матриць:\n", result_matrix)

    return row_sums, row_products, result_matrix

def process_matrix():

```

```

choice = input("Оберіть метод введення матриці (1 - з файлу, 2 - вручну): ")

if choice == "1":
    filename = input("Введіть шлях до файлу з матрицею: ")
    try:
        # Завантажуємо матрицю з файлу
        matrix = np.loadtxt(filename, dtype=int)
    except Exception as e:
        print(f"Помилка при завантаженні файлу: {e}")
        return
elif choice == "2":
    try:
        rows = int(input("Введіть кількість рядків матриці: "))
        cols = int(input("Введіть кількість стовпців матриці: "))
        matrix = []
        print("Введіть елементи матриці:")
        for i in range(rows):
            row = list(map(int, input(f"Рядок {i + 1}: ").split()))
            if len(row) != cols:
                print("Кількість елементів у рядку не відповідає вказаній  
кількості стовпців.")
            return
            matrix.append(row)
        matrix = np.array(matrix)
    except ValueError:
        print("Помилка: введіть коректні цілі числа.")
        return
else:
    print("Некоректний вибір. Спробуйте ще раз.")
    return

# Викликаємо функцію обчислення параметрів
calculate_matrix(matrix)

# Виконуємо основну функцію
if __name__ == "__main__":
    process_matrix()

```

ДОДАТОК Б
Скрін-шоти вікна виконання програми

```
Введіть число 1:  
1  
Введіть число 2:  
2  
Введіть число 3:  
3  
Введіть число 4:  
4  
Введіть число 5:  
5  
Кількість парних чисел у наборі: 2
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання
(Proc) 23

```
Оберіть метод введення матриці (1 - з файлу, 2 - вручну):  
2  
Введіть кількість рядків матриці:  
2  
Введіть кількість стовпців матриці:  
2  
Введіть елементи матриці:  
Рядок 1:  
1 2  
Рядок 2:  
3 4  
Суми елементів рядків: [3 7]  
Добутки елементів рядків: [ 2 12]  
Різниця початкової і випадкової матриць:  
[[-6 -4]  
[-1 -1]]
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Matrix 3