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

Кафедра систем управління літальними апаратами

Лабораторна робота № 2

з дисципліни «Об'єктно-орієнтоване програмування СУ»

Тема: «Структурування програм з використанням функцій»

ХАІ.301 . 172.519.СТ.5

Виконав студент гр. _____519СТ_____

_____Пащенко Нікіта___
(підпис, дата) (П.І.Б.)

Перевірив

_____ к.т.н., доц. О. В. Гавриленко
(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. 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 на рис.1.

Proc20	Описати функцію TriangleP (a, h), що знаходить периметр рівнобедреного трикутника по його основи a і висоті h, проведеної до основи (a і h - речові). За допомогою цієї функції знайти периметри трьох трикутників, для яких дані підстави і висоти. Для знаходження збоку b трикутника використовувати теорему Піфагора: b2 = (a / 2) 2 + h2.

Рис 1 - Завдання 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 на рис.2.

Matrix 6. У текстовому файлі задана матриця розміру M × N. Для кожного стовпця матриці з парним номером (2, 4, ...) знайти середнє арифметичне і найбільший з його елементів. Знайти векторний добуток заданої матриці з матрицею того ж розміру, заповненої випадковими числами.

Рис 2 - Завдання 2

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі Proc20

Вхідні дані:

a – основа трикутника.

Тип: float.

Діапазон: $a > 0$

h – висота трикутника, проведена до основи.

Тип: float.

Діапазон: $h > 0$

Вихідні дані:

perimeter – периметр трикутника.

Тип: float.

Алгоритм вирішення:

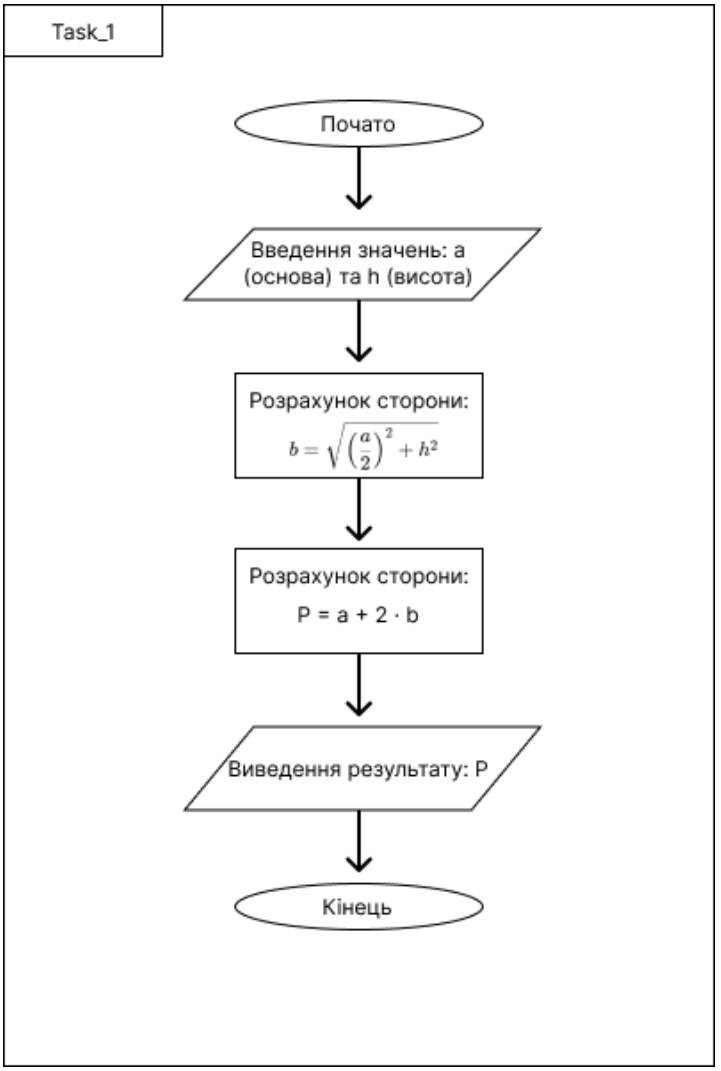

Рисунок 1 - Блок схема до Завдання 1

Завдання 2. Вирішення задачі Matrix 6

Вхідні дані:

filename – ім'я текстового файлу з матрицею.

Тип: str.

Обмеження: файл повинен існувати, матриця складається з дійсних чисел.

matrix – матриця, завантажена з файлу.

Тип: 2D-масив (numpy.ndarray).

Діапазон: $M > 0$, $N > 0$

Вихідні дані:

results – список середніх значень і максимумів для парних стовпців.

Тип: список (list).

product – векторний добуток матриць.

Тип: 2D-масив (numpy.ndarray).

Алгоритм вирішення:

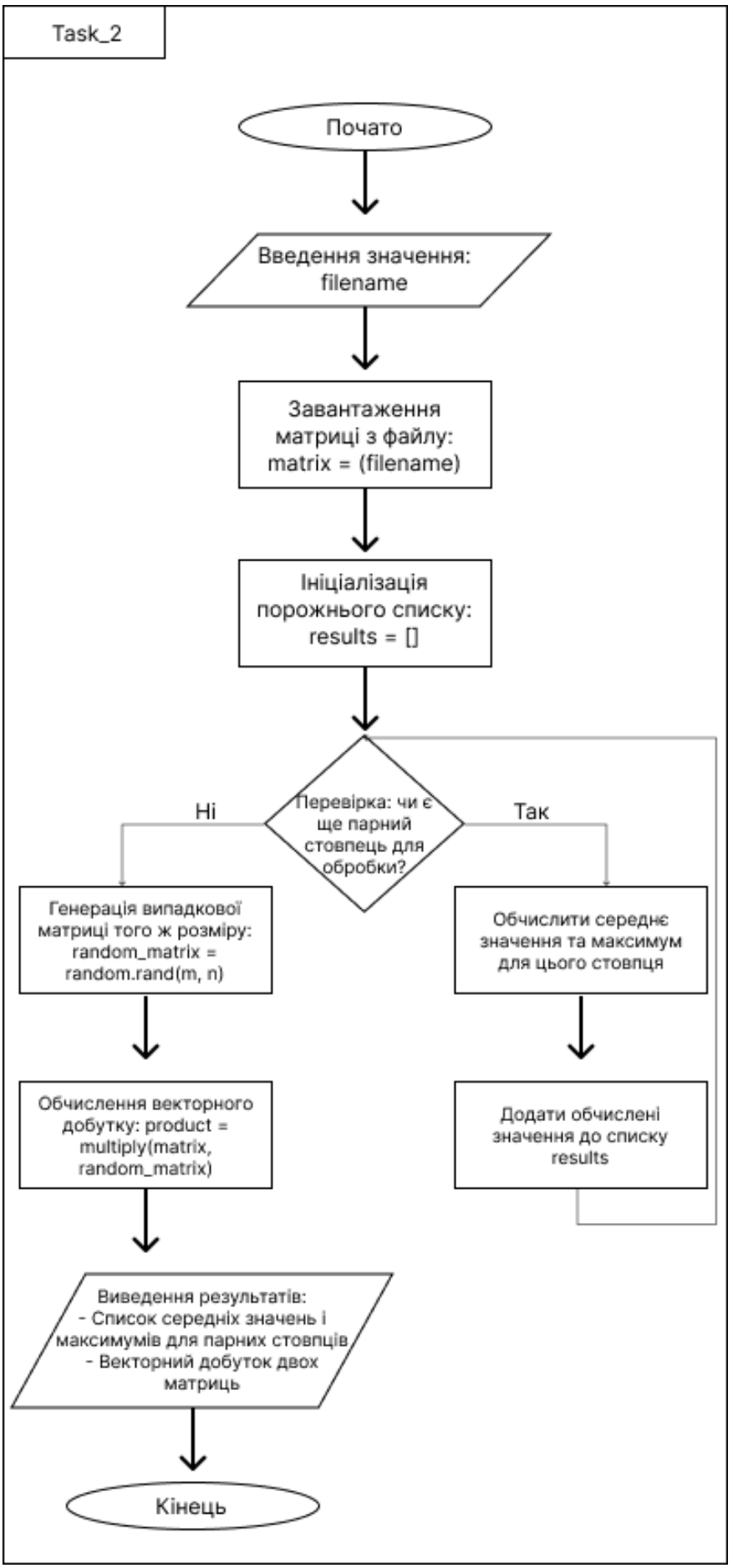

Рисунок 2 - Блок-схема до Завдання 2

Лістинг коду вирішення задачі наведено в дод. А (стор. 7). Екран роботи програми показаний на рис. Б. (стор. 7).

ВИСНОВКИ

У результаті виконання завдання було реалізовано дві функції для обчислень. Перша функція дозволяє знаходити периметр рівнобедреного трикутника за основою та висотою, а друга аналізувати парні стовпці матриці, обчислювати середні значення, максимуми та виконувати векторний добуток з випадковою матрицею.

ДОДАТОК А

Лістинг коду програми до задачі Proc20

```
<
   import math


def TriangleP(a, h):
    """Обчислення периметра рівнобедреного трикутника за основою a і висотою h."""
    b = math.sqrt((a / 2) ** 2 + h ** 2)  # Знаходимо сторону b за теоремою Піфагора
   return a + 2 * b  # Периметр трикутника


def find_perimeters(base_heights):
   """Повертає список периметрів для заданого списку основ і висот."""
   perimeters = []
   for a, h in base_heights:
       perimeters.append(TriangleP(a, h))
   return perimeters


def task1():
   """Введення даних, виклик функції та виведення результатів."""
   base_heights = []
   for i in range(3):
       a = float(input(f"Введіть основу трикутника {i + 1}: "))
       h = float(input(f"Введіть висоту трикутника {i + 1}: "))
       base_heights.append((a, h))

   perimeters = find_perimeters(base_heights)
   print("Периметри трикутників:", perimeters)

task1()

>
```

Лістинг коду програми до задачі Matix 6

```
<
   import numpy as np


def process_matrix(filename):
    """Обробка матриці: обчислення середнього та максимуму парних стовпців і векторний добуток."""
   matrix = np.loadtxt(filename)  # Завантаження матриці з файлу
   m, n = matrix.shape
```

```
    # Обчислення середнього та максимуму для парних стовпців
    results = []
    for col in range(1, n, 2):  # Парні стовпці (2, 4, ...)
        avg = np.mean(matrix[:, col])
        max_val = np.max(matrix[:, col])
          results.append((float(avg), float(max_val)))   # Перетворення в звичайні
числа

    # Генерація випадкової матриці того ж розміру
    random_matrix = np.random.rand(m, n)

    # Векторний добуток
    product = np.multiply(matrix, random_matrix)

    return results, product


def task2():
    """Зовнішня функція для виклику обробки матриці."""
    filename = input("Введіть ім'я файлу з матрицею: ")
    results, product = process_matrix(filename)

    print("Середнє та максимум для парних стовпців:")
    for i, (avg, max_val) in enumerate(results, start=1):
        print(f"Стовпець {2 * i}: Середнє = {avg:.2f}, Максимум = {max_val:.2f}")

    print("\nВекторний добуток матриць:\n", product)

task2()

>
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

```
Введіть основу трикутника 1: 51
Введіть висоту трикутника 1: 54
Введіть основу трикутника 2: 32
Введіть висоту трикутника 2: 21
Введіть основу трикутника 3: 11
Введіть висоту трикутника 3: 23
Периметри трикутників: [170.4361754243663, 84.80151512977633, 58.29693436154187]
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання 1. Proc20

```
Введіть ім'я файлу з матрицею: matrix.txt
Середнє та максимум для парних стовпців:
Стовпець 2: Середнє = 9.00, Максимум = 13.00
Стовпець 4: Середнє = 12.33, Максимум = 17.00

Векторний добуток матриць:
 [[ 3.82270293  0.03601532 10.05479625  1.27775476]
 [ 1.46301617  4.65918743 11.19991782  4.74540008]
 [ 1.17330233  8.35662101  2.65210178  4.94113179]]
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання 2. Matix 6