

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"

Виконала студентка гр. 321
Гаспарян Марія
Перевірив
_____ к.т.н., доц. О. В.
Гавриленко
_____ ас. В. О. Білозерський
(підпис, дата)

2023

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 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). Завдання представлено в табл.2.

Хід роботи

Завдання 1

Proc20	Описати функцію TriangleP (a, h), що знаходить периметр рівнобедреного трикутника по його основі 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: $b^2 = (a / 2)^2 + h^2$.
---------------	---

Завдання 2

Matrix 11. У текстовому файлі задана матриця розміру $M \times N$. Знайти максимальний серед мінімальних елементів її рядків. Відсортувати задану матрицю по стовпцях по зростанню.
--

Лістинг коду

```
import random

import numpy as np

def TriangleP(a, h):

    # Знайдемо довжину одного зі сторін трикутника (b) за теоремою
    Піфагора

    b = (a / 2) ** 2 + h ** 2

    b = b ** 0.5
```

```
# Обчислюємо периметр трикутника
```

```
perimeter = a + 2 * b
```

```
return perimeter
```

```
# Функція для введення даних, виклику функції та виведення  
результатів
```

```
def proc20():
```

```
    for i in range(3):
```

```
        a = float(input(f'Введіть основу трикутника {i + 1}: '))
```

```
        h = float(input(f'Введіть висоту трикутника {i + 1}: '))
```

```
        perimeter = TriangleP(a,h)
```

```
        print(f'Периметр трикутника {i + 1}: {perimeter}')
```

```
# Зовнішня функція для знаходження максимального серед  
мінімальних елементів рядків
```

```
def outer_function(filename):
```

```
    # Зчитування матриці з файлу
```

```
    matrix = np.loadtxt(filename)
```

```
    # Знаходження мінімальних елементів в кожному рядку
```

```
    min_in_rows = np.min(matrix, axis=1)
```

```
    # Знаходження максимального серед мінімальних елементів
```

```
    max_among_min = np.max(min_in_rows)
```

```
    return max_among_min
```

```
# Внутрішня функція для сортування матриці по стовпцях по  
зростанню
```

```

def inner_function(input_filename, output_filename):
    # Зчитування матриці з файлу
    matrix = np.loadtxt(input_filename)

    # Сортування матриці по стовпцях по зростанню
    sorted_matrix = np.sort(matrix, axis=0)

    # Збереження відсортованої матриці в новому файлі
    np.savetxt(output_filename, sorted_matrix, fmt='%d', delimiter=' ')

    return sorted_matrix

def matrix11():
    # Виклик зовнішньої функції для знаходження максимального
    # серед мінімальних елементів
    max_among_min = outer_function('matrix.txt')
    print("Максимальний серед мінімальних елементів рядків:",
    max_among_min)

    # Виклик внутрішньої функції для сортування матриці
    sorted_matrix = inner_function('matrix.txt', 'sorted_matrix.txt')
    print("Відсортована матриця:")
    print(sorted_matrix)

if __name__ == "__main__":
    while True:
        print("\nОберіть опцію:")
        print("1. Proc 20")
        print("2. Matrix 11")
        print("0. Вийти")

        choice = input("Ваш вибір: ")

        if choice == "1":
            proc20()
        elif choice == "2":

```

```
matrix11()
elif choice == "0":
    break
else:
    print("Невірний вибір. Виберіть 1, 2, 3 або 0.")
```

```
2. Matrix 11
0. Вийти
Ваш вибір: 1
Введіть основу трикутника 1: 6
Введіть висоту трикутника 1: 2
Периметр трикутника 1: 13.21110255092798
Введіть основу трикутника 2: 3
Введіть висоту трикутника 2: 1
Периметр трикутника 2: 6.60555127546399
Введіть основу трикутника 3: 6
Введіть висоту трикутника 3: 3
Периметр трикутника 3: 14.48528137423857

Оберіть опцію:
1. Proc 20
2. Matrix 11
0. Вийти
Ваш вибір: 2
Максимальний серед мінімальних елементів рядків: 5.0
Відсортована матриця:
[[1. 2. 0.]
 [5. 6. 4.]
 [8. 9. 7.]]

Оберіть опцію:
1. Proc 20
2. Matrix 11
0. Вийти
Ваш вибір:
```

Висновок: під час виконання цієї лабораторної роботи я вивчив структурування програм з використанням функцій