

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"

XAI.301.173.322.02 ЛР

Виконав студент гр. _____322_____

_____Гусар Анастасія_____

(підпис, дата)

(П.І.Б.)

Перевірив

_____к.т.н., зав. кафедри 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. 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. 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

Proc19 Описати функцію RingS (R1, R2) дійсного типу, яка знаходить площу кільця, укладеного між двома колами із загальним центром і радіусами R1 і R2 (R1 і R2 - речові, $R1 > R2$). З її допомогою знайти площі трьох кілець, для яких дані зовнішні і внутрішні радіуси. Скористатися формулою площі круга радіусу R: $S = \pi \cdot R^2$. Як значення π вважати рівним 3.14.

Вхідні дані:

- 1) R1 — радіус зовнішнього кола
- 2) R2 — радіус внутрішнього кола

Алгоритм вирішення:

```
import math

def RingS(R1, R2):
    """Функція для обчислення площі кільця між двома колами"""
    if R1 <= R2:
        raise ValueError("Радіус зовнішнього кола (R1) повинен бути більшим за радіус внутрішнього кола (R2).")

    # Використовуємо значення  $\pi = 3.14$ 
    pi = 3.14

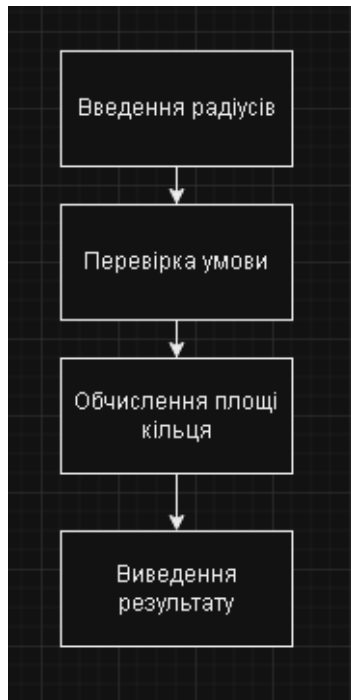
    # Обчислюємо площі двох кіл та їх різницю
    area_ring = pi * (R1**2 - R2**2)

    return area_ring

# Приклад використання функції для трьох кілець:
R1_1, R2_1 = 8, 4
R1_2, R2_2 = 14, 9
R1_3, R2_3 = 22, 14

print(f"Площа першого кільця: {RingS(R1_1, R2_1)}")
print(f"Площа другого кільця: {RingS(R1_2, R2_2)}")
print(f"Площа третього кільця: {RingS(R1_3, R2_3)}")
```

```
Площа першого кільця: 150.72
Площа другого кільця: 361.1
Площа третього кільця: 904.32
```



Matrix 12. У текстовому файлі задана матриця розміру $M \times N$. Знайти мінімальний серед максимальних елементів її стовпців. Відсортувати задану матрицю по рядках по спадаючій.

Вхідні дані:

Матриця

```
3 4 3
3 4 2 2
2 1 5 1
5 2 1 4
```

Алгоритм вирішення:

```
def read_matrix_from_input():
    """Читання матриці з введення користувача"""
    matrix = [
        [3, 4, 3],
        [3, 4, 2, 2],
        [2, 1, 5, 1],
        [5, 2, 1, 4]
    ]
    return matrix

def find_min_of_max_elements(matrix):
    """Знаходимо мінімальний серед максимальних елементів стовпців"""
    if not matrix:
        return None
```

```

# Трансформуємо матрицю для однакових кількостей елементів у стовпцях
num_columns = max(len(row) for row in matrix) # Найбільша кількість стовпців
max_elements = []

for col in range(num_columns):
    # Отримуємо максимальний елемент для кожного стовпця
    column_elements = [matrix[row][col] for row in range(len(matrix)) if col < len(matrix[row])]
    if column_elements:
        max_in_column = max(column_elements)
        max_elements.append(max_in_column)

return min(max_elements)

def sort_matrix_by_rows(matrix):
    """Сортуємо матрицю по рядках за спадяючим порядком"""
    return [sorted(row, reverse=True) for row in matrix]

def print_matrix(matrix):
    """Виведення матриці на екран"""
    for row in matrix:
        print(" ".join(map(str, row)))

def main():
    matrix = read_matrix_from_input()

    if matrix: # Перевіряємо, чи матриця не порожня
        min_of_max = find_min_of_max_elements(matrix)
        print(f"Мінімальний серед максимальних елементів стовпців: {min_of_max}")

        sorted_matrix = sort_matrix_by_rows(matrix)
        print("Відсортована матриця:")
        print_matrix(sorted_matrix)
    else:
        print("Матриця порожня.")

if __name__ == "__main__":
    main()

```

```

Мінімальний серед максимальних елементів стовпців: 4
Відсортована матриця:
4 3 3
4 3 2 2
5 2 1 1
5 4 2 1

```



Висновок

У ході виконання завдання були розроблені функції для обчислення площі кільця між двома колами та для обробки матриці. Перша функція ефективно обчислює площу кільця, враховуючи умови $R_1 > R_2$ та $R_1 > R_2$. Друга функція знаходить мінімум серед максимальних елементів стовпців матриці та сортує її за рядками. Завдання демонструє важливість математичних розрахунків та правильної обробки даних у програмуванні.

