

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 проєктування систем
управління»

Тема: "Структурування програм з використанням
функцій"

ХАІ.301 . Інженерія мобільних додатків. 312ст.1 ЛР

Виконав студент гр. _____312ст_____

_____Васильєв Б.А._____

(підпис, дата)

(П.І.Б.)

Перевірів

_____к.т.н., доц. О. В. Гавриленко

_____ас. В. О. Білозерський

(підпис, дата)

(П.І.Б.)

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. Варіант 24

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`. Варіант 2

ВИКОНАННЯ РОБОТИ

Завдання 1: Вирішення задачі Proc24

Вхідні дані:

in_data: Опис: Список із 5 цілих чисел, введених користувачем. Тип: list[int]

Вихідні дані:

odd_count: Опис: Кількість непарних чисел у списку. Тип: int

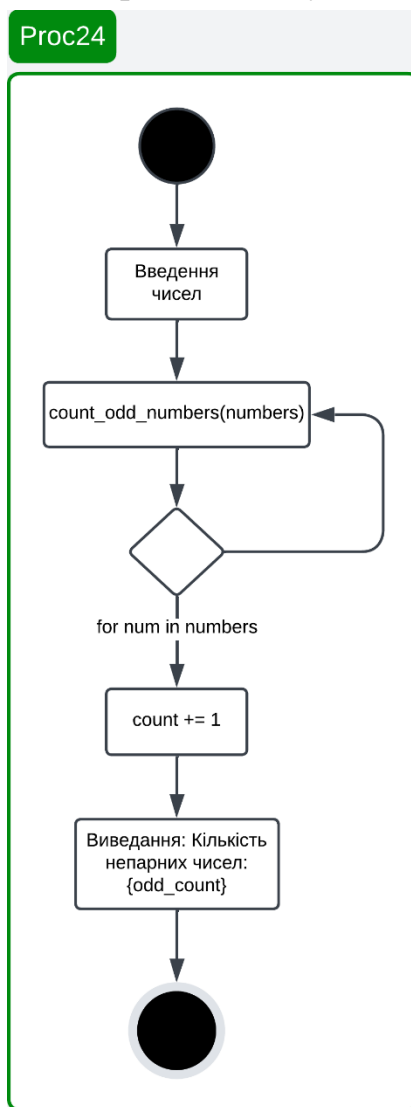


Рис. 1 діаграма Завдання 1: Вирішення задачі Proc24

Завдання 2: Вирішення задачі Matrix 2

Вхідні дані:

file_name: Опис: Ім'я або шлях до файлу з матрицею. Тип: str

K: Опис: Номер стовпця для обробки. Тип: int

Вихідні дані:

column_sum: Опис: Сума елементів K-го стовпця. Тип: int

column_product: Опис: Добуток елементів K-го стовпця. Тип: int

sum_matrix: Опис: Сума вхідної матриці з одиничною матрицею.

Тип: numpy.ndarray

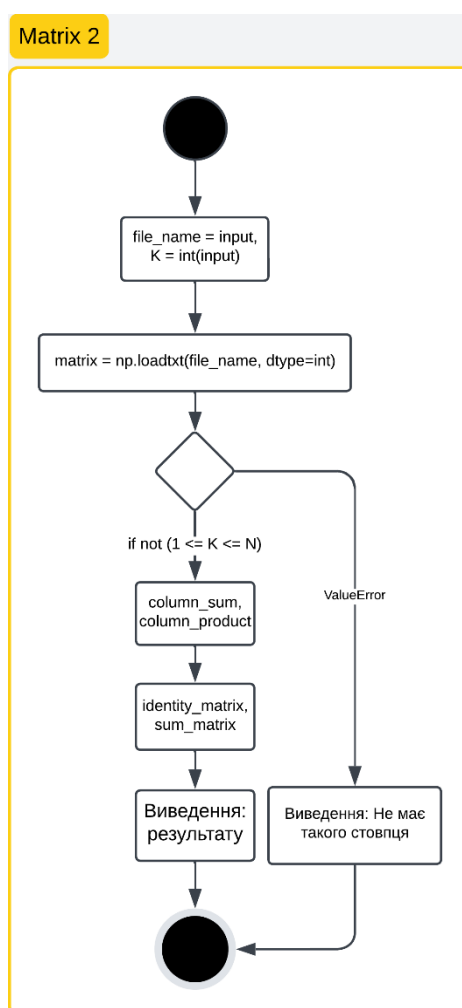


Рис. 2 діаграма Завдання 2: Вирішення задачі Matrix 2

ВИСНОВКИ

У цих двох завданнях я реалізував функції для обробки матриць та перевірки чисел на непарність. Це дало змогу потренуватися зчитувати й обробляти дані, працювати з умовами та циклами, а також використовувати функції для спрощення коду.

ДОДАТОК А

Лістинг коду програми до завдання Proc24

```
def Odd(K):
    "Перевіряє, чи є ціле число K непарним"
    return K % 2 != 0

def count_odd_numbers(numbers):
    "Підраховує кількість непарних чисел у списку"
    count = 0
    for num in numbers:
        if Odd(num):
            count += 1
    return count

def input_and_process():
    "Вводить п'ять чисел, знаходить кількість непарних чисел"
    numbers = []
    for _ in range(5):
        num = int(input("Введіть число: "))
        numbers.append(num)
    odd_count = count_odd_numbers(numbers)
    print(f"Кількість непарних чисел: {odd_count}")
```

ДОДАТОК Б

Лістинг коду програми до завдання Matrix 2

```
import numpy as np

def process_matrix(file_name, K):
    "Обробляє матрицю з файлу, обчислює потрібні параметри."
    matrix = np.loadtxt(file_name, dtype=int)
    M, N = matrix.shape
    if not (1 <= K <= N):
        raise ValueError("Не має такого стовпця")

    column_sum = np.sum(matrix[:, K - 1])
    column_product = np.prod(matrix[:, K - 1])

    identity_matrix = np.eye(M, dtype=int)
    sum_matrix = matrix + identity_matrix

    return column_sum, column_product, sum_matrix

def matrix_task():
    "функція для виклику обробки матриці."
    file_name = input("Введіть ім'я файлу з матрицею: ")
    K = int(input("Введіть номер стовпця K: "))
    column_sum, column_product, sum_matrix = process_matrix(file_name, K)

    print(f"Сума елементів {K}-го стовпця: {column_sum}")
    print(f"Добуток елементів {K}-го стовпця: {column_product}")
    print("Сума матриці з одиничною матрицею:")
    print(sum_matrix)
```

ДОДАТОК В

```
from Proc24 import input_and_process
from Matrix2 import matrix_task

def main():
    while True:
        print("Оберіть завдання для виконання:")
        print("1 - Завдання1 (Proc)")
        print("2 - Завдання 2 (Matrix)")
        print("0 - Вихід")
        choice = input("Ваш вибір: ")

        if choice == "1":
            input_and_process()
        elif choice == "2":
            matrix_task()
        elif choice == "0":
            print("Вихід з програми.")
            break
        else:
            print("Невірний вибір, спробуйте ще раз.")

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

ДОДАТОК Г

Скрін-шоти вікна виконання програми

```
Введіть число: 3
Введіть число: 4
Введіть число: 10
Введіть число: 7
Введіть число: 15
Кількість непарних чисел: 3
```

Рисунок Б.3 – Екран виконання програми для вирішення завдання Pros 24

```
Введіть ім'я файлу з матрицею: matrix.txt
Введіть номер стовпця K: 2
Сума елементів 2-го стовпця: 15
Добуток елементів 2-го стовпця: 80
Сума матриці з одиничною матрицею:
[[ 2  2  3]
 [ 4  6  6]
 [ 7  8 10]]
```

Рисунок Б.4 – Екран виконання програми для вирішення завдання Matrix 2