МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 2

«Об'єктно-орієнтоване програмування авіаційно-транспортних систем»

Тема: "Розробка структурованих програм з розгалуженням та повтореннями"

ХАІ.301.173.3-92ав(і).2ЛР

Виконав студентка гр. \_\_\_\_3-92ав(і)\_\_\_\_\_\_\_

Микитас Анна Олексіївна (П.І.Б.)

Перевірив:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доц. О. В. Гавриленко

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ас.  В. О. Білозерський

(підпис, дата) (П.І.Б.)

2023

# МЕТА РОБОТИ

Вивчити теоретичний матеріал щодо синтаксису на мові Python і поданням у вигляді UML діаграм діяльності алгоритмів з розгалуження та циклами, а також навчитися використовувати функції, інструкції умовного переходу і циклів для реалізації інженерних обчислень.

# ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на алгоритми з розгалуженням. Завдання представлено в табл.1.

Завдання 2. Дано дійсні числа (xi , yi), i = 1,2, ... n, – координати точок на площині. Визначити кількість точок, що потрапляють в геометричну область заданого кольору (або групу областей). Варіанти геометричних областей представлені в табл.2.

Завдання 3. Дослідити ряд на збіжність. Умова закінчення циклу обчислення суми прийняти у вигляді: | un | <E або | un | > G де е – мал величина для переривання циклу обчислення суми сходиться ряду (е = 10-5 ... 10-20); g – величина для переривання циклу обчислення суми розходиться ряду (g = 102 ... 105). Варіанти представлено в табл.3.

Завдання 4. Для багаторазового виконання будь-якого з трьох зазначених вище завдань на вибір розробити циклічний алгоритм організації меню в командному вікні.

# ВИКОНАННЯ РОБОТИ

Завдання 1. If14

Вхідні дані (ім’я, опис, тип, обмеження):

num1: Перше число.

Опис: Це перше введене число.

Тип: float.

num2: Друге число.

Опис: Це друге введене число.

Тип: float.

num3: Третє число.

Опис: Це третє введене число.

Тип: float.

Вихідні дані (ім’я, опис, тип):

min\_num: Найменше число серед введених.

Опис: Це найменше з трьох введених чисел.

Тип: float.

max\_num: Найбільше число серед введених.

Опис: Це найбільше з трьох введених чисел.

Тип: float.

Алгоритм вирішення показано нижче

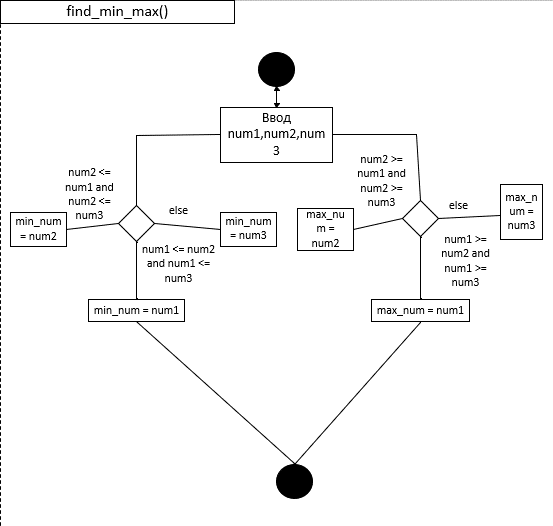


Рис 1-Алгоритм роботи функції find\_min\_max()

Лістинг коду вирішення задачі наведено в дод. А (стор.9). Екран роботи програми показаний на рис. Б.1.

Завдання 2. 12

Вхідні дані:

n: Кількість точок.

Опис: Кількість точок, які будуть введені.

Тип: int.

Вихідні дані:

count: Кількість точок в жовтій геометричній області.

Опис: Кількість точок, які потрапляють в область.

Тип: int.

Алгоритм вирішення показано нижче

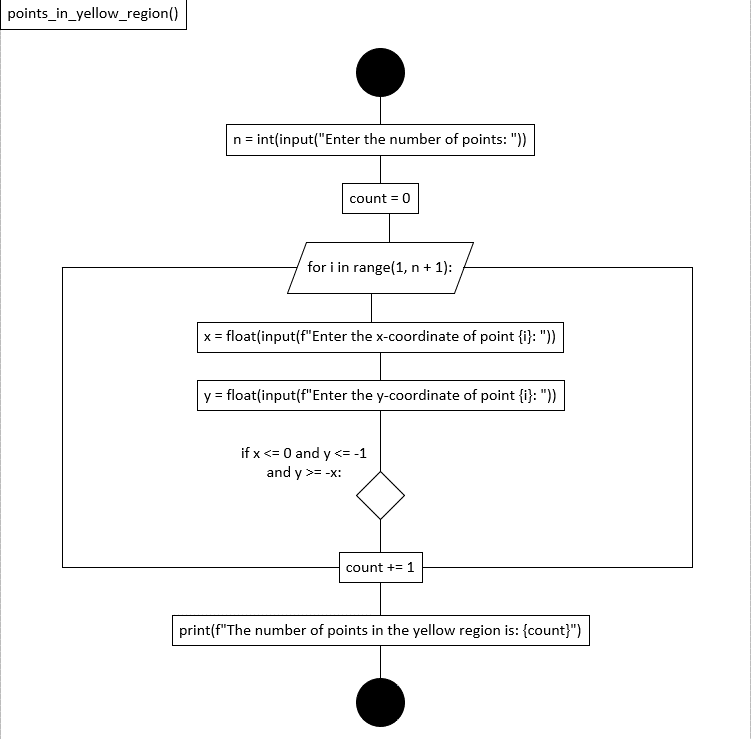


Рисунок 2 – Алгоритм роботи функції points\_in\_yellow\_region()

Лістинг коду вирішення задачі наведено в дод. А (стор.9). Екран роботи програми показаний на рис. Б.2.

Завдання 3. 10

Функція calculate\_un:

Вхідні дані:

n: Ціле число.

Опис: Параметр для обчислення послідовності.

Тип: int.

Вихідні дані:

Результат обчислення формули для послідовності.

Функція check\_convergence:

Вхідні дані:

Відсутні, але внутрішні константи e та g можна розглядати як константи з обмеженнями.

Тип: float.

Вихідні дані:

n: Ціле число.

Опис: Мінімальне значення, при якому відношення un+1/un за модулем менше за e або більше за g.

Тип: int.

Алгоритм вирішення показано нижче

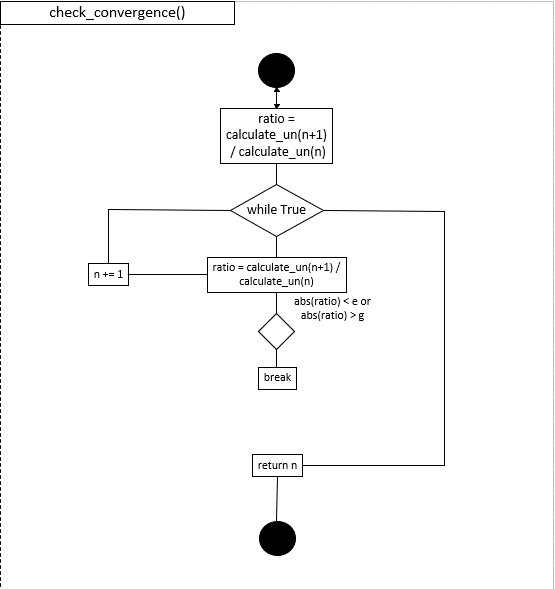


Рис 3-Алгоритм роботи функції check\_convergence()

Лістинг коду вирішення задачі наведено в дод. А (стор.10). Екран роботи програми показаний на рис. Б.3.

Завдання 4.

Вхідні дані:

choice (тип: int) - користувацький вибір завдання (1-3, 0 для виходу).

Вихідні дані:

result (тип: int) - результат обчислення для завдання 3 (n - кількість членів послідовності).

Алгоритм вирішення показано нижче

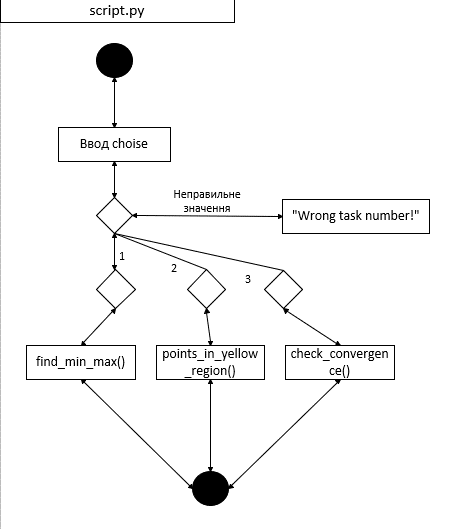


Рис 4-Алгоритм роботи файлу script.py

Лістинг коду вирішення задачі наведено в дод. А (стор.10).

# ВИСНОВКИ

У результаті виконання лабораторної роботи "Розробка структурованих програм з розгалуженням та повтореннями в Python", було вивчено основні конструкції мови програмування Python для роботи з умовами (розгалуженням) та циклами (повтореннями). Завдяки практичним завданням, в яких реалізовано пошук мінімального та максимального числа, визначення кількості точок в геометричній області та перевірка збіжності ряду, учасники лабораторної роботи закріпили отримані знання на практиці. Окрім того, вони отримали навички створення структурованих програм з використанням функцій та модулів, що сприяє поліпшенню організації коду та підтримці читабельності програмного забезпечення.

ДОДАТОК А

Лістинг коду програми до задач 1-3

import math

def find\_min\_max():

    try:

        # Введення трьох чисел

        num1 = float(input("Enter the first number: "))

        num2 = float(input("Enter the second number: "))

        num3 = float(input("Enter the third number: "))

        # Знаходження найменшого числа з використанням операторів if

        if num1 <= num2 and num1 <= num3:

            min\_num = num1

        elif num2 <= num1 and num2 <= num3:

            min\_num = num2

        else:

            min\_num = num3

        # Знаходження найбільшого числа з використанням операторів if

        if num1 >= num2 and num1 >= num3:

            max\_num = num1

        elif num2 >= num1 and num2 >= num3:

            max\_num = num2

        else:

            max\_num = num3

        # Виведення результатів

        print("The smallest number is:", min\_num)

        print("The largest number is:", max\_num)

    except ValueError:

        print("ERROR: Please enter valid numeric values.")

def points\_in\_yellow\_region():

    try:

        # Введення кількості точок

        n = int(input("Enter the number of points: "))

        # Ініціалізація лічильника точок, що потрапляють в геометричну область

        count = 0

        # Зчитування координат точок та перевірка, чи вони потрапляють в область

        for i in range(1, n + 1):

            x = float(input(f"Enter the x-coordinate of point {i}: "))

            y = float(input(f"Enter the y-coordinate of point {i}: "))

            # Перевірка, чи точка потрапляє в жовту геометричну область

            if x <= 0 and y <= -1 and y >= -x:

                count += 1

        # Виведення результату

        print(f"The number of points in the yellow region is: {count}")

    except ValueError:

        print("ERROR: Please enter valid numeric values.")

def calculate\_un(n):

    return (math.factorial(n) - 3\*\*n) / (n\*\*n)

def check\_convergence():

    e = 1e-5  # мала величина для переривання циклу обчислення суми сходиться ряду

    g = 1e2   # величина для переривання циклу обчислення суми розходиться ряду

    n = 1

    while True:

        ratio = calculate\_un(n+1) / calculate\_un(n)

        if abs(ratio) < e or abs(ratio) > g:

            break

        n += 1

    return n

Лістинг коду програми до задач 4

import module

choice = int(input("Please, choose the task 1-3 (0-EXIT): "))

while choice:

    if choice==1:

        module.find\_min\_max();

    elif choice==2:

        module.points\_in\_yellow\_region();

    elif choice==3:

        result = module.check\_convergence()

        print(f"The series converges up to n = {result}")

    else:

        print("Wrong task number!")

    choice = int(input("Please, choose the task again (0-EXIT): "))

print("Good by!")

ДОДАТОК Б

Скрін-шоти вікна виконання програми

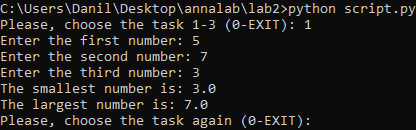


Рисунок Б.1 – Екран виконання програми для вирішення завдання   
1

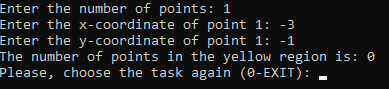


Рисунок Б.2 – Екран виконання програми для вирішення завдання   
2



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