

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 проектування програм»

Тема: "Розробка структурованих програм з
розгалуженням та повтореннями"

ХАІ.301 . 174. 322. № 2ЛР

Виконав студент гр. 322

_____ Безпалова Світлана Вікторівна
(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 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 (x_i, y_i) , $i = 1, 2, \dots, n$, – 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: $|u_n| < \epsilon$ або $|u_n| > G$ де ϵ – мала величина для переривання циклу обчислення суми сходиться ряду ($\epsilon = 10^{-5} \dots 10^{-20}$); G – величина для переривання циклу обчислення суми розходиться ряду ($G = 10^2 \dots 10^5$). Варіанти представлено в табл.3.

ВИКОНАННЯ РОБОТИ

Завдання 1. Дано три числа. Знайти середнє з них (тобто число, розташоване між найменшим і найбільшим).

Вхідні дані:

- Три дійсних числа: a, b, c .
- Вихідні дані: середнє число серед трьох введених.

Алгоритм вирішення на рис 1.

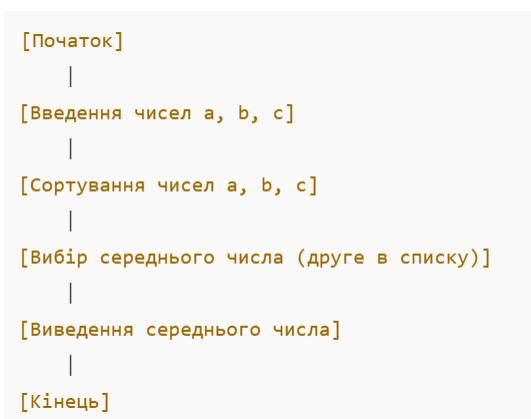
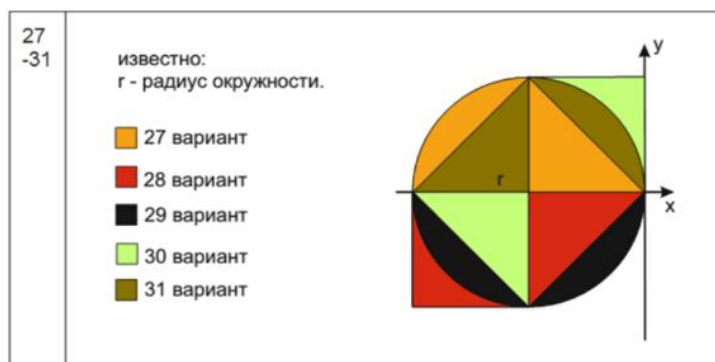


Рисунок 1.

Лістинг коду вирішення задачі наведено в дод. А (стр.7).

Екран роботи програми показаний на рис. Б (стр.9).

Завдання 2. Дано дійсні числа (x_i, y_i) , $i = 1, 2, \dots, n$, – координати точок на площині. Визначити кількість точок, що потрапляють в геометричну область заданого кольору (або групу областей). Варіанти геометричних областей представлені в табл.2. (29 варіант)



Вхідні дані:

- Радіус кола r .
- Кількість точок n .
- Координати кожної точки (x, y) .

Вихідні дані:

- Кількість точок, які належать чорній області (варіант 29).

Алгоритм вирішення на рис 2.

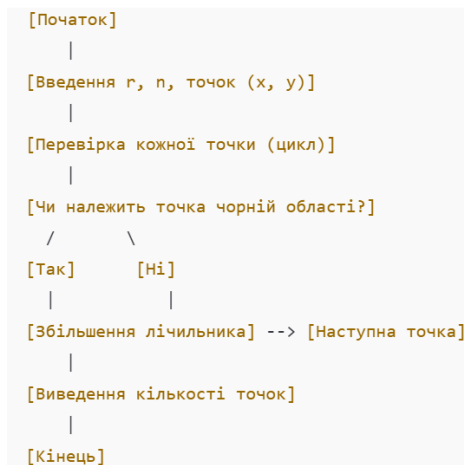


Рисунок 2.

Лістинг коду вирішення задачі наведено в дод. А (стр.7).

Екран роботи програми показаний на рис. Б (стр.9).

Завдання 3. Дослідити ряд на збіжність. Умова закінчення циклу обчислення суми прийняти у вигляді: $|u_n| < \epsilon$ або $|u_n| > G$ де ϵ – мала величина для переривання циклу обчислення суми сходиться ряду ($\epsilon = 10^{-5} \dots 10^{-20}$); G – величина для переривання циклу обчислення суми розходиться ряду ($G = 10^2 \dots 10^5$). Варіанти представлено в табл.3. (варіант 22)

$$\sum_{n=1}^{\infty} \frac{(2+n)!}{x^n 2^{2n+1}}$$

Вхідні дані:

- x — дійсне число, $x > 0$.
- ϵ — мала величина для перевірки сходження ($10^{-5} \leq \epsilon \leq 10^{-20}$).
- G — велика величина для перевірки розходження ($10^2 \leq G \leq 10^5$).

Вихідні дані:

- Значення суми ряду до виконання умови сходження або розходження.

Алгоритм вирішення на рис 3.

```

[Початок]
|
[Введення x, epsilon, g]
|
[Ініціалізація n = 1, total_sum = 0]
|
[Обчислення поточного члена ряду]
|
[Перевірка |u_n| < epsilon?] --Так--> [Виведення суми, Кінець]
|
[Перевірка |u_n| > g?] --Так--> [Ряд розходиться, Кінець]
|
[Додати u_n до total_sum]
|
[Інкремент n]
|
[Повернення до обчислення члена ряду]

```

Рисунок 3.

Лістинг коду вирішення задачі наведено в дод. А (стр.8).

Екран роботи програми показаний на рис. Б (стр.9).

ВИСНОВКИ

У ході виконання роботи було розв'язано три завдання за допомогою мови програмування Python, які демонструють роботу з різними типами даних та алгоритмами:

Перше завдання:

- Було розроблено алгоритм для знаходження середнього з трьох чисел.
- Програма коректно обробляє введені дані, сортує числа і знаходить середнє, що є другим за величиною числом у відсортованому списку.

Друге завдання:

- Було реалізовано алгоритм для перевірки приналежності точок до геометричної області (чорної області для варіанту 29).
- Програма успішно обробляє координати точок, перевіряє їх на відповідність умовам перебування в заданій області, використовуючи математичні рівняння.

Третє завдання:

- Реалізовано програму для дослідження ряду на збіжність, з використанням умов завершення обчислень ($|u_n| < \epsilon$ для сходження або $|u_n| > g$ для розходження).

- Програма обчислює члени ряду та виконує перевірки відповідно до заданих параметрів. У результаті визначається, чи ряд сходиться, чи розходиться.

Результати:

Усі три завдання було виконано коректно. Отримані результати відповідають умовам задач. Кожна програма працює стабільно та демонструє необхідний функціонал.

ДОДАТОК А

Лістинг коду для Завдання 1.

```
def task1():
    # Знаходження середнього з трьох чисел
    print("\nЗадача 1: Знайти середнє з трьох чисел.")
    a = float(input("Введіть перше число (a): "))
    b = float(input("Введіть друге число (b): "))
    c = float(input("Введіть третє число (c): "))

    # Визначення середнього
    numbers = [a, b, c]
    numbers.sort() # Сортуюмо числа
    middle = numbers[1] # Середнє - друге в списку після сортування

    print(f"Середнє число: {middle}\n")
```

Лістинг коду для Завдання 2.

```
def task2():
    # Визначення кількості точок у чорній області
    print("\nЗадача 2: Визначити кількість точок у чорній області (варіант 29).")
    r = float(input("Введіть радіус кола (r): "))
    n = int(input("Введіть кількість точок (n): "))

    points = []
    for i in range(n):
        x, y = map(float, input(f"Введіть координати точки {i + 1} (x y): ").split())
        points.append((x, y))

    def is_in_black_area(x, y, r):
        # Умова перебування точки у чорній області (варіант 29)
        in_circle = x**2 + y**2 <= r**2 # Точка всередині кола
        in_lower_half = y <= 0 # Точка у нижній напівплощині
        outside_triangle = y < -x and y < x # Точка поза трикутником
        return in_circle and in_lower_half and outside_triangle

    count = 0
    for x, y in points:
        if is_in_black_area(x, y, r):
            count += 1

    print(f"Кількість точок у чорній області (варіант 29): {count}\n")
```

Лістинг коду для Завдання 3.

```
from math import factorial
def task3():
    # Дослідження ряду на збіжність
    print("\nЗадача 3: Дослідити ряд на збіжність.")
    x = float(input("Введіть значення x (x > 0): "))
    epsilon = float(input("Введіть значення ε (epsilon): "))
    g = float(input("Введіть значення g (g): "))

    n = 1
    total_sum = 0
    while True:
        # Обчислення поточного члена ряду
        term = factorial(2 + n) / (x**n * 2**(2 * n + 1))
        total_sum += term

        # Перевірка умов зупинки
        if abs(term) < epsilon: # Сходження
            print(f"Ряд сходиться. Сума: {total_sum:.10f}")
            break
        elif abs(term) > g: # Розходження
            print("Ряд розходиться.")
            break

    n += 1
```


ДОДАТОК Б

```
Выберите задачу для выполнения:
1: Найти среднее из трёх чисел (задача 1)
2: Определить количество точек в области (задача 2)
3: Исследовать ряд на сходимость (задача 3)
0: Выход
Ваш выбор: 1

Задача 1: Найти среднее из трёх чисел.
Введите первое число: 3
Введите второе число: 4
Введите третье число: 5
Среднее число: 4.0

Выберите задачу для выполнения:
1: Найти среднее из трёх чисел (задача 1)
2: Определить количество точек в области (задача 2)
3: Исследовать ряд на сходимость (задача 3)
0: Выход
Ваш выбор: 2

Задача 2: Определить количество точек в заданной области (чёрная зона, вариант 29) .
Введите радиус окружности (r): 5
Введите количество точек: 3
Введите координаты точки 1 (x y): 0 -3
Введите координаты точки 2 (x y): 2 -4
Введите координаты точки 3 (x y): 4 1
Количество точек в чёрной области (вариант 29): 2

Выберите задачу для выполнения:
1: Найти среднее из трёх чисел (задача 1)
2: Определить количество точек в области (задача 2)
3: Исследовать ряд на сходимость (задача 3)
0: Выход
Ваш выбор: 3

Задача 3: Исследовать ряд на сходимость.
Введите значение x: 2
Введите epsilon (e): 1e-5
Введите g: 1e5
Ряд расходится.
Выберите задачу для выполнения:
1: Найти среднее из трёх чисел (задача 1)
2: Определить количество точек в области (задача 2)
3: Исследовать ряд на сходимость (задача 3)
0: Выход
Ваш выбор: 0
Выход из программы.

...Program finished with exit code 0
Press ENTER to exit console.
```