

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 проєктування систем
управління»

Тема: "Розробка структурованих програм з
розгалуженням та повтореннями"

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 щодо синтаксису на мові Python і поданням 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.

If20. [На числовій осі розташовані три точки: A, B, C. Визначити, яка з двох останніх точок (B або C) розташована ближче до A, і вивести цю точку і її відстань від точки A.

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).

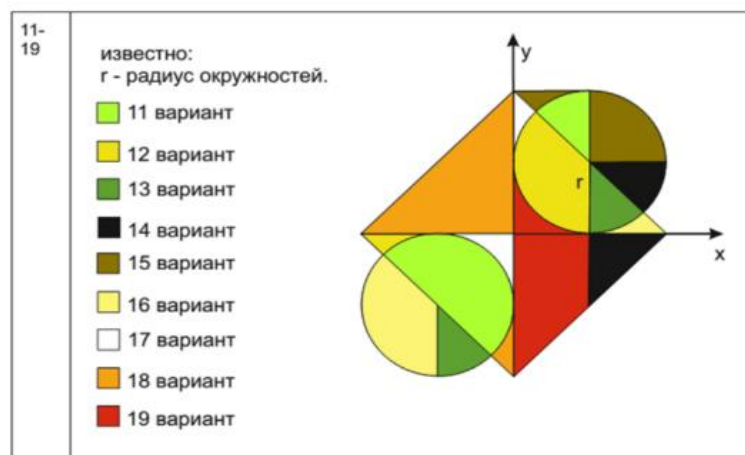


Рис.1 Завдання 2

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.

| | |
|----|--|
| 10 | $\sum_{n=1}^{\infty} \frac{n! - 3^n}{n^n}$ |
|----|--|

Рис.2 Завдання 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: Визначення останньої та середньої цифри тризначного числа.

Вхідні дані: Тризначне ціле число, яке лежить в діапазоні від 100 до 999.

Вихідні дані: Остання та середня цифри цього числа.

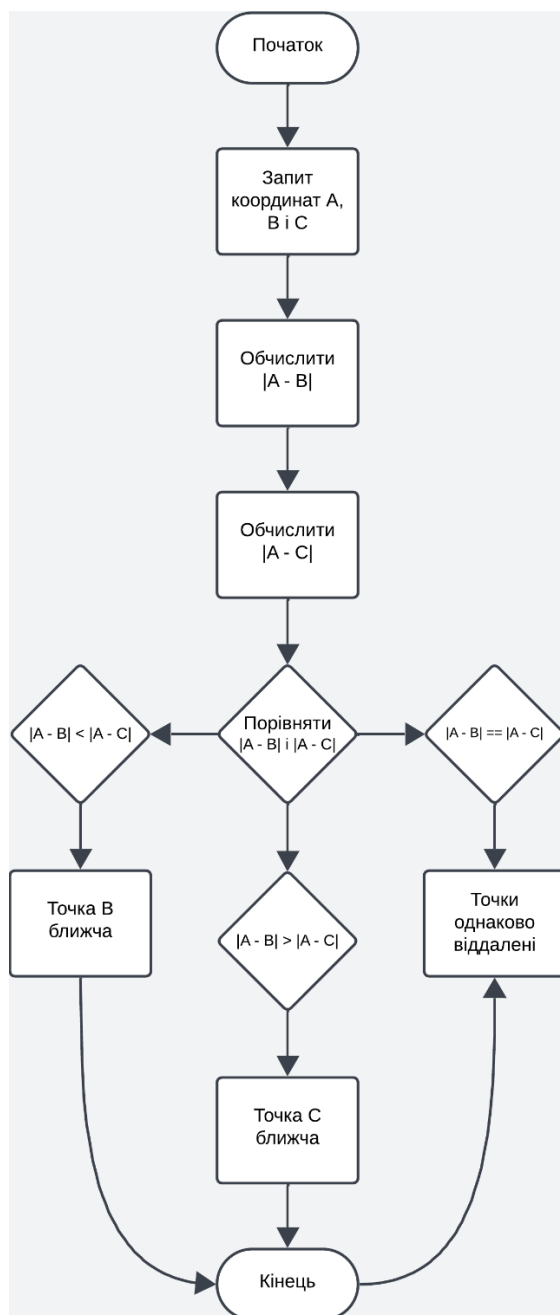


Рис. 3 діаграма Завдання 1: Визначення найближчої точки до A

Завдання 2: Обчислення математичного виразу.

Вхідні дані: Дійсне число x , яке не повинно дорівнювати нулю.

Результат: Обчислене значення виразу.

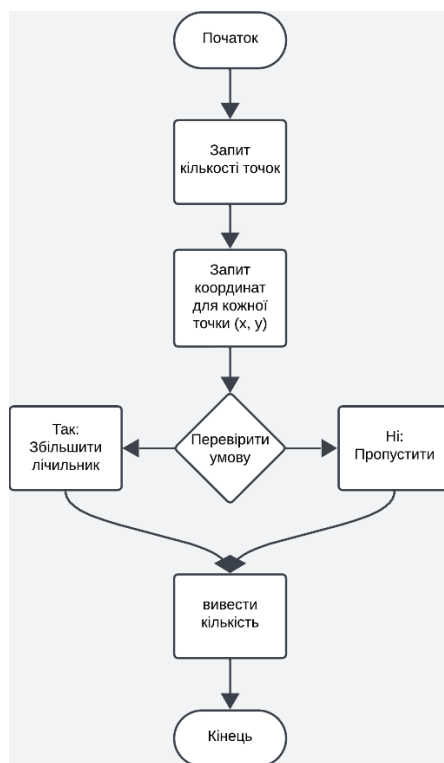


Рис. 4 діаграма Завдання 2: Визначення кількості точок в зеленій області

Завдання 3: Перевірка, чи всі введені числа є додатними.

Вхідні дані: Три цілі числа.

Результат: Логічне значення (істина або хибність), яке вказує на те, чи всі числа є додатними.

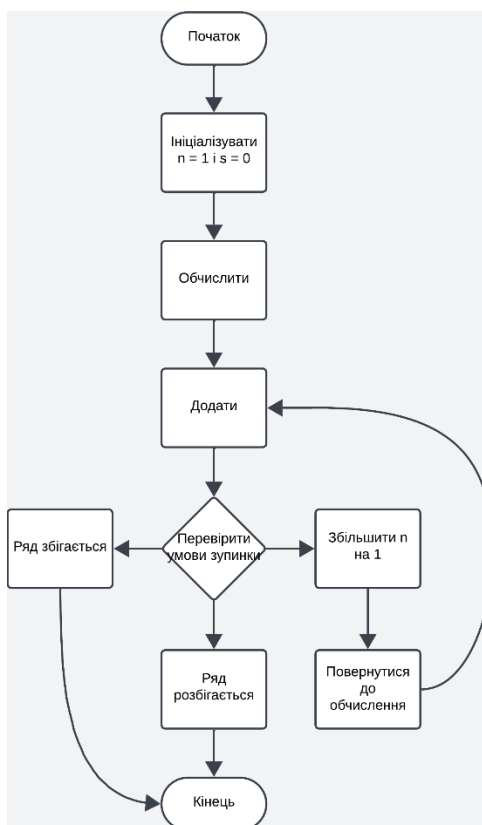


Рис. 4 діаграма Завдання 3: Дослідження збіжності ряду

Завдання 4: Циклічний алгоритм організації меню для вибору завдань

Вхідні дані:

Номер завдання: Користувач вводить число від 0 до 3 для вибору відповідного завдання:

Додаткові вхідні дані для кожного завдання:

Для завдання 1: Координати точок А, В, С.

Для завдання 2: Кількість точок та їх координати.

Для завдання 3: Значення для завершення обчислень.

ДОДАТОК А

Лістинг коду програми до завдання 1-3

```
import math

# Завдання 1: Визначення найближчої точки до А
def task_if20():

    # Визначає, яка з точок В або С знаходиться ближче до точки А.

    try:
        A = float(input("Введіть координату точки А: "))
        B = float(input("Введіть координату точки В: "))
        C = float(input("Введіть координату точки С: "))

        distance_B = abs(A - B)
        distance_C = abs(A - C)

        if distance_B < distance_C:
            print(f"Точка В ближча до точки А. Відстань: {distance_B}")
        elif distance_C < distance_B:
            print(f"Точка С ближча до точки А. Відстань: {distance_C}")
        else:
            print(f"Точки В і С однаково віддалені від точки А. Відстань: {distance_B}")
    except ValueError:
        print("введіть правильні числа.")

# Завдання 2: Визначення кількості точок в зеленій області (варіант 11)
def task_geometry11(points):

    #Підраховує кількість точок, що потрапляють в зелену область.
    #points - список точок (кожна точка має координати (x, y)).
```

```

count = 0
r = 1
for x, y in points:
    # Умова для попадання точки в зелену область
    if (x - r) ** 2 + (y - r) ** 2 <= r ** 2:
        count += 1
return count

```

Завдання 3: Дослідження збіжності ряду

```
def task_series10(epsilon=1e-10, g=1e10):
```

```
    # epsilon - маленька величина для завершення циклу при збіжності.
```

```
    n = 1
```

```
    s = 0
```

```
    while True:
```

```
        term = (math.factorial(n) - 3 ** n) / n ** n
```

```
        s += term
```

```
        if abs(term) < epsilon:
```

```
            print(f"Ряд збігається. Сума: {s}")
```

```
            break
```

```
        elif abs(term) > g:
```

```
            print(f"Ряд розбігається. Сума: {s}")
```

```
            break
```

```
        n += 1
```

ДОДАТОК Б

Лістинг коду програми до завдання 4

```

import sys
from lab_2_oop import task_if20, task_geometry11, task_series10

# Завдання 4: Меню для вибору завдань
def menu():
    while True:
        print("\nВиберіть завдання:")
        print("1. Визначити, яка точка ближче до А.")
        print("2. Порахувати кількість точок в зеленій області.")
        print("3. Дослідити збіжність ряду.")
        print("0. Вихід")

        try:
            choice = int(input("Ваш вибір (0-3): "))

            if choice == 1:
                task_if20()
            elif choice == 2:
                points = [(float(input("x: ")), float(input("y: "))) for _ in
range(int(input("Кількість точок: ")))]
                print(f"Кількість точок у зеленій області:
{task_geometry11(points)}")
            elif choice == 3:
                task_series10()
            elif choice == 0:
                sys.exit(0)
            else:
                print("Невірний вибір.")
        except ValueError:
            print("Помилка введення.")

```



```
# Викликати головне меню:  
if __name__ == "__main__":  
    menu()
```

ВИСНОВКИ

У результаті виконання завдань було реалізовано три функції, які визначають найближчу точку до A , підраховують кількість точок у геометричній області та досліджують збіжність математичного ряду.

ДОДАТОК Б

Скрін-шоти вікна виконання програми

```
Введіть координату точки А: 5  
Введіть координату точки В: 3  
Введіть координату точки С: 8  
Точка В ближча до точки А. Відстань: 2.0
```

Рисунок Б.5 – Екран виконання програми для вирішення завдання 1

```
Кількість точок: 5  
х: 0.5  
у: 0.5  
х: 1.5  
у: 1.5  
х: 0.2  
у: 0.2  
х: 1.0  
у: 0.8  
х: 2.0  
у: 2.0  
Кількість точок у зеленій області: 3
```

Рисунок Б.6 – Екран виконання програми для вирішення завдання 2

```
Ряд збігається. Сума: -4.783041449365289
```

Рисунок Б.7 – Екран виконання програми для вирішення завдання 3