

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів  
Кафедра систем управління літальних апаратів

## Лабораторна робота № 5

з дисципліни «Алгоритмізація та програмування»

Тема: "Структурування програм з використанням функцій"

XAI.301. 141. 319a. 19 ЛР

Виконав студент гр. \_\_\_\_\_ 319a \_\_\_\_\_

\_\_\_\_\_ Овчинніков Станіслав \_\_\_\_\_

(підпис, 15.12.2024)

(П.І.Б.)

Перевірив

\_\_\_\_\_ к.т.н., доц. Олена

ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

2024

## МЕТА РОБОТИ

Вивчити теоретичний матеріал із синтаксису мовою C ++ і поданням у вигляді UML діаграм циклічних алгоритмів і реалізувати алгоритми з використанням інструкцій циклу з передумовою, циклу з післяумовою і параметризованого циклу мовою C ++ в середовищі Visual Studio.

## ПОСТАНОВКА ЗАДАЧІ

Завдання 1. ( Вар 34 ) Дано дійсні числа  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$ , – координати точок на площині. Визначити кількість точок, що потрапляють в фігуру заданого кольору (або групу фігур). Варіанти фігур представлено в табл.1.

Завдання 2. ( Вар 10 ) Дано дійсне число  $x$  і натуральне число  $n$ . Необхідно:

а) Обчислити значення виразу при заданих  $x$  і  $n$  для виразу з табл.2.

б) Вивести: для парних варіантів – значення кожного третього елемента, для непарних – значення кожного четвертого елемента.

Завдання 3. ( Вар 30 ) Дослідити ряд на збіжність. Умова закінчення циклу обчислення суми прийняти у вигляді:  $|u_n| < \epsilon$  або  $|u_n| > g$ , де  $\epsilon$  – мала величина для переривання циклу обчислення суми збіжного ряду ( $\epsilon = 10^{-5} \dots 10^{-20}$ );  $g$  – величина для переривання циклу обчислення суми розбіжного ряду ( $g = 10^2 \dots 10^5$ ). Варіанти представлено в табл.3.

Завдання 4. Організувати меню в командному вікні для багаторазового виконання завдань \*та для перевірки вхідних даних на коректність описати функції, що повертають логічне значення (true – в разі коректного значення переданих параметрів і false – в іншому випадку).

2

## ВИКОНАННЯ РОБОТИ

Завдання 1 Вар 34

32  
-36

$r$  - радіус кола

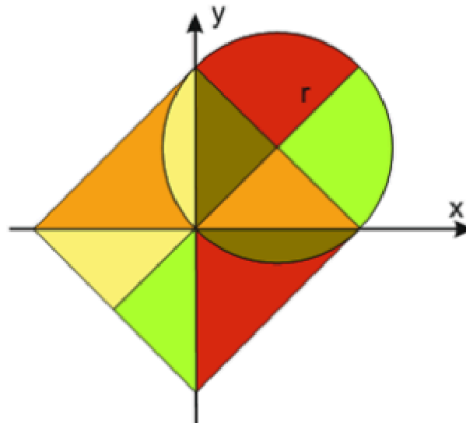
32 вар.

33 вар.

34 вар.

35 вар.

36 вар.



Дано дійсні числа  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$ , – координати точок на площині.  
Визначити кількість точок, що потрапляють в фігуру заданого кольору (або групу фігур).

Вхідні дані (ім'я, опис, тип):

Кількість точок  $n$  (наприклад: 3).

Координати кожної точки  $x$  та  $y$  (наприклад: (1, 2), (0, 0), (3, 4)).

Радіус фігури  $r$  (наприклад: 5).

Вихідні дані : Кількість точок, що потрапляють у фігуру (наприклад: 2).

### Алгоритм вирішення

Ввести кількість точок  $n$  і радіус фігури  $r$ .

Для кожної точки:

Ввести координати  $x$  та  $y$ .

Перевірити умову належності до чверті кола:

$x \geq 0, y \geq 0 \text{ і } x^2 + y^2 \leq r^2$ .

Перевірити умову належності до трикутника:

$x \geq 0 \text{ і } y \geq -x$ .

Якщо хоча б одна з умов виконується, збільшити лічильник точок.

Після перевірки всіх точок вивести кількість точок, що потрапляють у фігуру.

```
#include <iostream>
#include <cmath>
using namespace std;

// Функція для перевірки попадання точки у фігуру (34 вар)
bool isInFigure(double x, double y, double r) {
    bool inQuarterCircle = (x >= 0 && y >= 0 && (x * x + y * y <= r * r)); // Чверть кола
    bool inTriangle = (x >= 0 && y >= -x); // Трикутник
    return inQuarterCircle || inTriangle;
}

void task34() {
    int n;
    double x, y, r; // Радіус фігури
    cout << "Введіть кількість точок: "; cin >> n;
    cout << "Введіть радіус фігури r: "; cin >> r;

    int count = 0; // Лічильник точок
    for (int i = 0; i < n; i++) {
        cout << "Введіть координати x і y для точки " << i + 1 << ": ";
        cin >> x >> y;
        if (isInFigure(x, y, r)) {
            count++;
        }
    }
    cout << "Кількість точок, попадаючих в фігуру: " << count << endl;
}
```

Рисунок 1 – вар 34

Лістинг коду вирішення задачі :

```
#include <iostream>
#include <cmath>
using namespace std;

// Функція для перевірки попадання точки у фігуру (34 вар)
bool isInFigure(double x, double y, double r) {
    bool inQuarterCircle = (x >= 0 && y >= 0 && (x * x + y * y <= r * r)); // Чверть кола
    bool inTriangle = (x >= 0 && y >= -x); // Трикутник
    return inQuarterCircle || inTriangle;
}
```

```

}

void task34() {
    int n;
    double x, y, r; // Радіус фігури
    cout << "Введіть кількість точок: "; cin >> n;
    cout << "Введіть радіус фігури r: "; cin >> r;

    int count = 0; // Лічильник точок
    for (int i = 0; i < n; i++) {
        cout << "Введіть координати x и y для точки " << i + 1 << ": ";
        cin >> x >> y;
        if (isInFigure(x, y, r)) {
            count++;
        }
    }
    cout << "Кількість точок, попадаючих в фігуру: " << count << endl;
}

```

```

Виберіть задачу для виконання:
1. Перевірка точок на попадання в фігуру (34 вар)
2. Виведення суми ряду (10 вар)
3. Виведення суми ряду з факторіалом (30 вар)
0. Вихід
Ваш вибір: 1
Введіть кількість точок: 2
Введіть радіус фігури r: 3
Введіть координати x и y для точки 1: 7
9
Введіть координати x и y для точки 2: 2 5
Кількість точок, попадаючих в фігуру: 2

```

Екран роботи програми показаний на рис 2.

Завдання 2 ( Вар 10 )

Завдання 2. Дано дійсне число  $x$  і натуральне число  $n$ . Необхідно:

а) Обчислити значення виразу при заданих  $x$  і  $n$  для виразу з табл.2.



b) Вивести: для парних варіантів – значення кожного третього елемента, для непарних – значення кожного четвертого елемента. Вхідні дані (ім'я, опис, тип, обмеження): Три целых числа A, B, C, тип int.

10	$S = \sum_{i=0}^n \frac{x^{i+2}}{5^i + i},  x  < 4$
----	---

Вхідні дані:

Значення x (наприклад: 2).

Кількість членів ряду n (наприклад: 4).

Вихідні дані:

Значення суми ряду S (наприклад: 6.8).

Алгоритм вирішення:

Ввести значення x та кількість членів ряду n.

Ініціалізувати змінну sum = 0.

Для кожного значення i від 0 до n:

Обчислити чисельник:  $x^{i+2}$ .

Обчислити знаменник:  $5^i + i$ .

Обчислити член ряду:

$term = x^{i+2} / (5^i + i)$ .

Додати term до змінної sum.

Після завершення циклу вивести значення суми sum.

```

// Функція для обчислення суми ряду (10 вар)
void task10() {
    int n;
    double x, sum = 0.0;

    cout << "Введите значение x: "; cin >> x;
    cout << "Введите количество членов ряда n: "; cin >> n;

    for (int i = 0; i <= n; i++) {
        double numerator = pow(x, i + 2);           // Чисельник:  $x^{(i+2)}$ 
        double denominator = pow(5, i) + i;         // Знаменник:  $5^i + i$ 
        sum += numerator / denominator;
    }

    cout << "Сумма ряда: " << sum << endl;
}

```

Рисунок 3- код ( вар 10 )

Лістинг коду :

```

// Функція для обчислення суми ряду (10 вар)
void task10() {
    int n;
    double x, sum = 0.0;

    cout << "Введите значение x: "; cin >> x;
    cout << "Введите количество членов ряда n: "; cin >> n;

    for (int i = 0; i <= n; i++) {
        double numerator = pow(x, i + 2);           // Чисельник:  $x^{(i+2)}$ 
        double denominator = pow(5, i) + i;         // Знаменник:  $5^i + i$ 
        sum += numerator / denominator;
    }

    cout << "Сумма ряда: " << sum << endl;
}

```

```
Выберите задачу для выполнения:
1. Проверка точек на попадание в фигуру (34 вар)
2. Вычисление суммы ряда (10 вар)
3. Вычисление суммы ряда с факториалом (30 вар)
0. Выход
Ваш выбор: 2
Введите значение x: 3
Введите количество членов ряда n: 4
Сумма ряда: 19.5574
```

Экран роботи програми показаний на рис 4.

Завдання 3 ( Вар 30) Дослідити ряд на збіжність. Умова закінчення циклу обчислення суми прийняти у вигляді:  $|u_n| < \epsilon$  або  $|u_n| > g$ , де  $\epsilon$  – мала величина для переривання циклу обчислення суми збіжного ряду ( $\epsilon = 10^{-5} \dots 10^{-20}$ );  $g$  – величина для переривання циклу обчислення суми розбіжного ряду ( $g = 10^2 \dots 10^5$ ).

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

Значення  $x$  (наприклад: 1.5).

Кількість членів ряду  $n$  (наприклад: 5).

Вихідні дані :

Значення суми ряду  $S$  (наприклад: 4.75).

Алгоритм вирішення:

Ввести значення  $x$  та кількість членів ряду  $n$ .

Ініціалізувати змінну  $sum = 0$ .

Для кожного значення  $i$  від 1 до  $n$ :

Обчислити чисельник:  $x^i * i!$ .

Для обчислення факторіала  $i!$  послідовно перемножити числа від 1 до  $i$ .

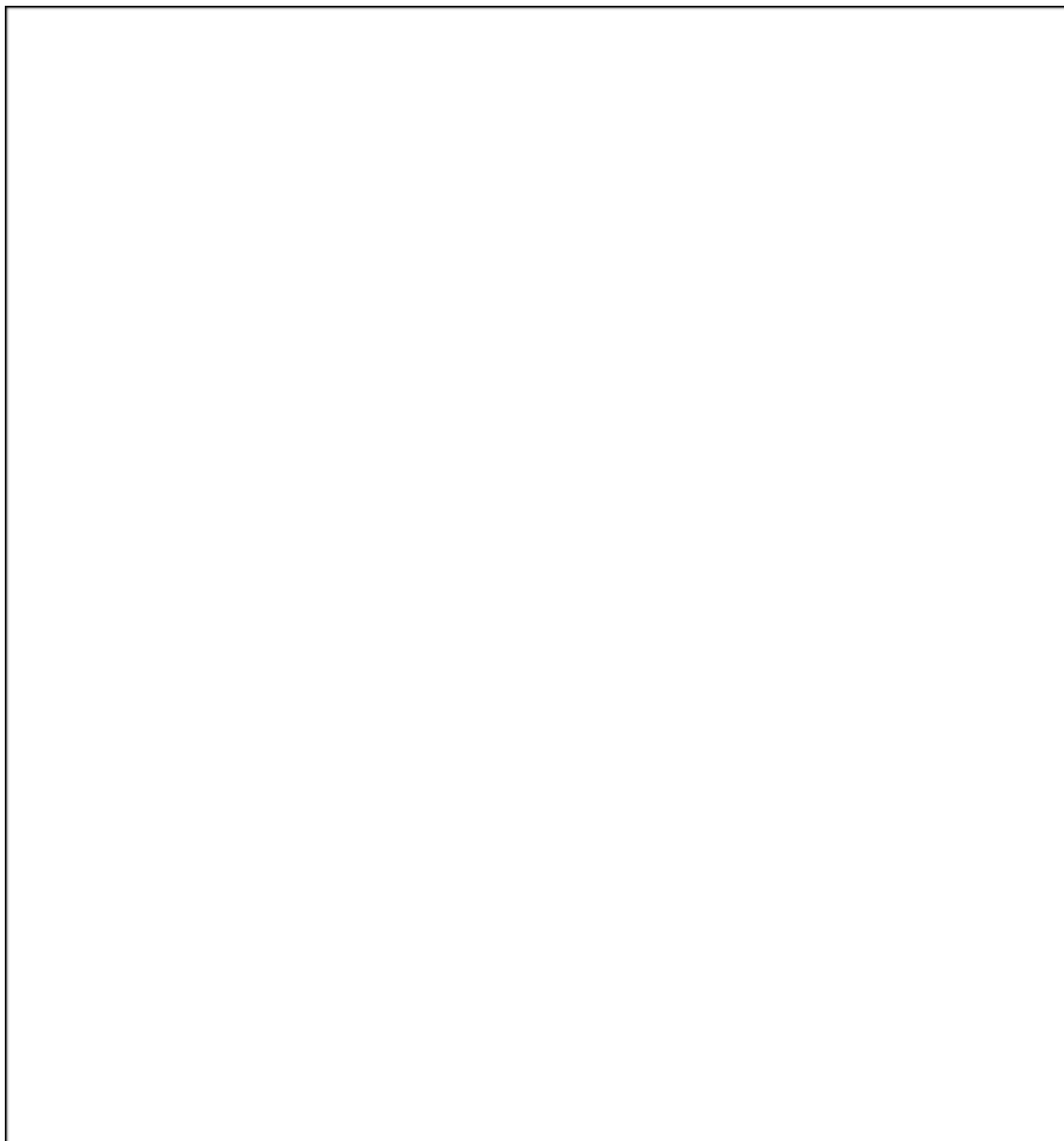
Обчислити знаменник:  $i^{(i/2)}$ .

Обчислити член ряду:

$\text{term} = (x^i * i!) / i^{(i/2)}$ .

Додати  $\text{term}$  до змінної  $\text{sum}$ .

Після завершення циклу вивести значення суми  $\text{sum}$ .



```

// Функція для обчислення суми ряду з факторіалом (30 вар)
double factorial(int n) {
    double result = 1.0;
    for (int i = 1; i <= n; i++) {
        result *= i;
    }
    return result;
}

void task30() {
    int n;
    double x, sum = 0.0;

    cout << "Введіть значення x: "; cin >> x;
    cout << "Введіть кількість членів ряду n: "; cin >> n;

    for (int i = 1; i <= n; i++) {
        double numerator = pow(x, i) * factorial(i); // Чисельник:  $x^i * i!$ 
        double denominator = pow(i, i / 2.0); // Знаменник:  $i^{(i/2)}$ 
        sum += numerator / denominator;
    }

    cout << "Сума ряду: " << sum << endl;
}

// Головне меню програми
int main() {
    int choice;

    do {
        cout << "\nВиберіть задачу для виконання:\n";
        cout << "1. Перевірка точок на попадання в фігуру (34 вар)\n";
        cout << "2. Виведення суми ряду (10 вар)\n";
        cout << "3. Виведення суми ряду з факторіалом (30 вар)\n";
        cout << "0. Вихід\n";
        cout << "Ваш вибір: ";
        cin >> choice;

        switch (choice) {
            case 1: task34(); break;
            case 2: task10(); break;
            case 3: task30(); break;
            case 0: cout << "Вихід із програми. До зустрічі!\n"; break;
            default: cout << "Невірний вибір, спробуйте знову.\n";
        }
    } while (choice != 0);

    return 0;
}

```

Рисунок 5- код ( вар 30 ) .

Лістинг коду :

```

// Функція для обчислення суми ряду з факторіалом (30 вар)
double factorial(int n) {
    double result = 1.0;

```

```

    for (int i = 1; i <= n; i++) {
        result *= i;
    }
    return result;
}

void task30() {
    int n;
    double x, sum = 0.0;

    cout << "Введите значение x: "; cin >> x;
    cout << "Введите количество членов ряда n: "; cin >> n;

    for (int i = 1; i <= n; i++) {
        double numerator = pow(x, i) * factorial(i); // Чисельник:  $x^i \cdot i!$ 
        double denominator = pow(i, i / 2.0); // Знаменник:  $i^{(i/2)}$ 
        sum += numerator / denominator;
    }

    cout << "Сумма ряда: " << sum << endl;
}

```



```

Выберите задачу для выполнения:
1. Проверка точек на попадание в фигуру (34 вар)
2. Вычисление суммы ряда (10 вар)
3. Вычисление суммы ряда с факториалом (30 вар)
0. Выход
Ваш выбор: 3
Введите значение x: 2
Введите количество членов ряда n: 5
Сумма ряда: 107.93

```

Экран роботи програми показаний на рис 6.

## ВИСНОВКИ

Усі три задачі реалізовано успішно з використанням різних математичних підходів: геометричних перевірок, степеневих функцій та обчислення факторіалів. Алгоритми показали свою ефективність і точність при роботі з обчисленнями та перевіркою умов.