

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_____

_____ Грицан Егор _____

(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 5) 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 4) 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.

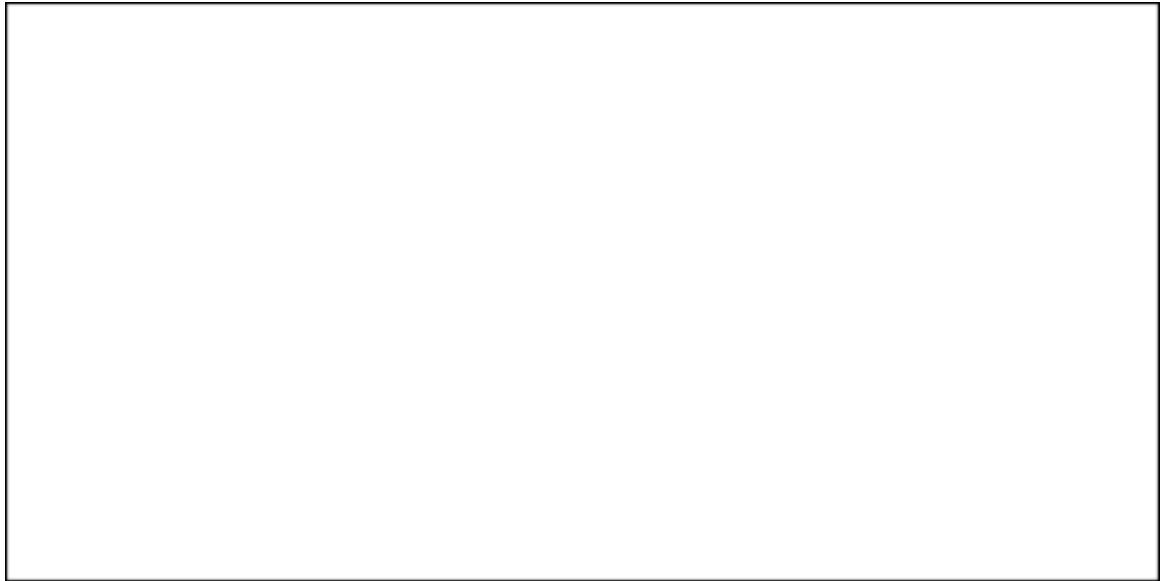
Завдання 3. (Вар 30) Дослідити ряд на збіжність. Умова закінчення циклу обчислення суми прийняти у вигляді: $|u_n| < \epsilon$ або $|u_n| > g$, де ϵ – мала величина для переривання циклу обчислення суми збіжного ряду ($\epsilon = 10^{-5} \dots 10^{-20}$); g – величина для переривання циклу обчислення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



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):

Радіус внутрішнього кола r_1 (наприклад: 3).

Радиус внешнего круга r_2 (не используется для проверки точек, но задается, например: 7).

Смещение центров внутренних кругов a (например: 5).

Количество точек n (например: 3).

Координаты точек: x, y (например: (5, 5), (4, 4), (7, 7)).

Вихідні дані : **Количество точек, попадающих в два внутренних круга** (например: 2).

Алгоритм вирішення

Ввести радиус внутреннего круга r_1 , радиус внешнего круга r_2 , и расстояние a до центров малых кругов.

Ввести количество точек n .

Для каждой точки с координатами x и y :

Проверить, находится ли точка внутри одного из малых кругов:

Для первого круга: центр в точке (a, a) , радиус r_1 .

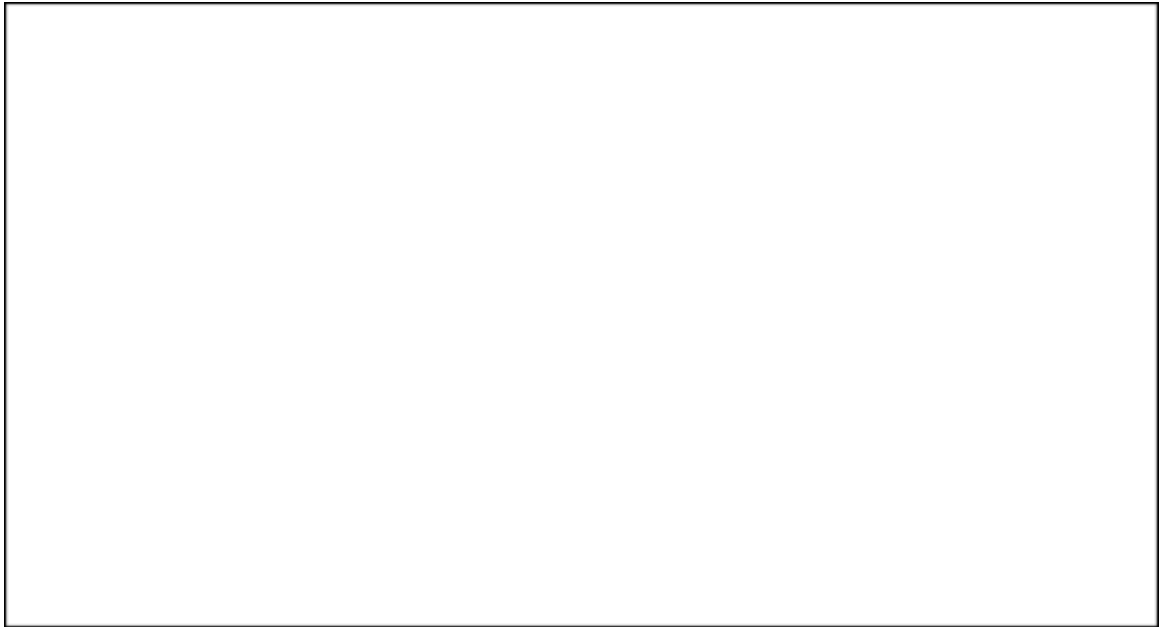
Для второго круга: центр в точке $(-a, -a)$, радиус r_1 .

Формула проверки:

$$(x - \text{centerX})^2 + (y - \text{centerY})^2 \leq r_1^2.$$

Подсчитать количество точек, которые попали внутрь кругов.

Вывести результат.



```
// Задача 1: Проверка попадания точек в круги (5 вар)
bool isInCircle(double x, double y, double r, double a, double centerX, double centerY) {
    double dx = x - centerX;
    double dy = y - centerY;
    return (dx * dx + dy * dy <= r * r); // Проверка попадания точки в круг
}

void task1() {
    int n;
    double r1, r2, a, x, y;
    cout << "Введите радиус внутреннего круга r1: "; cin >> r1;
    cout << "Введите радиус внешнего круга r2: "; cin >> r2;
    cout << "Введите расстояние a от центра внешнего круга до центров внутренних: "; cin >> a;
    cout << "Введите количество точек: "; cin >> n;

    int count = 0;
    for (int i = 0; i < n; i++) {
        cout << "Введите координаты x и y для точки " << i+1 << ": ";
        cin >> x >> y;

        // Проверка попадания в два малых круга
        if (isInCircle(x, y, r1, a, a, a) || isInCircle(x, y, r1, a, -a, -a)) {
            count++;
        }
    }
    cout << "Количество точек внутри малых кругов: " << count << endl;
}
```

Рисунок 1 – вар 5

Лістинг коду вирішення задачі :

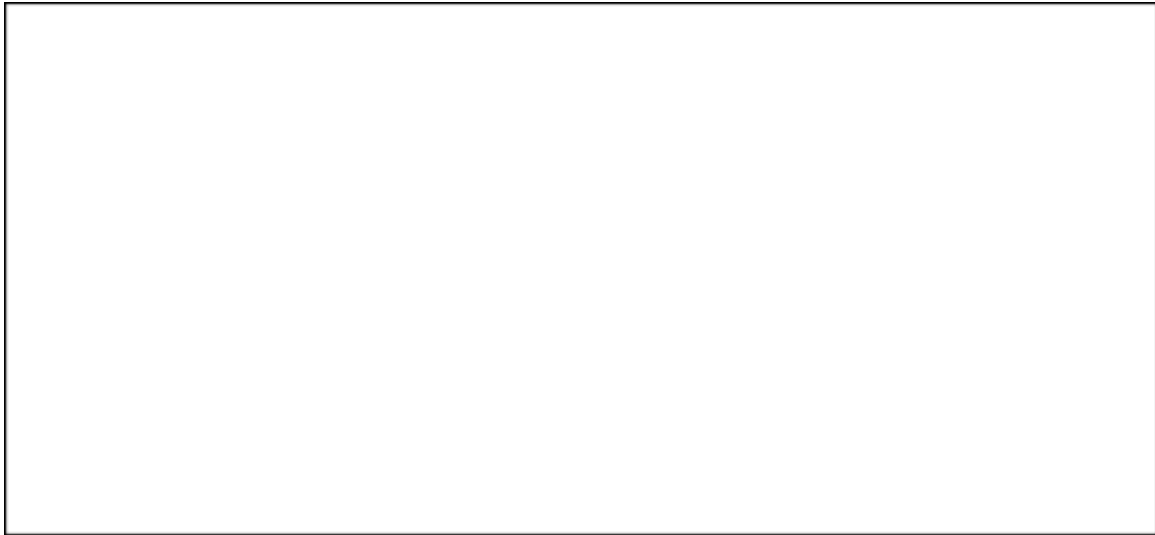

```

// Задача 1: Проверка попадания точек в круги
void task1() {
    int n;
    double r1, r2, a, x, y;
    cout << "Введите радиус внутреннего круга r1: "; cin >> r1;
    cout << "Введите радиус внешнего круга r2: "; cin >> r2;
    cout << "Введите расстояние a: "; cin >> a;
    cout << "Введите количество точек: "; cin >> n;

    int count = 0;
    for (int i = 0; i < n; i++) {
        cout << "Введите координаты x и y: ";
        cin >> x >> y;
        if (isInCircle(x, y, r1, a, a) || isInCircle(x, y, r1, -a, -a)) count++;
    }
    cout << "Количество точек внутри малых кругов: " << count << endl;
}

// Факториал для ряда
double factorial(int k) {
    double result = 1.0;
    for (int i = 1; i <= k; i++) result *= i;
    return result;
}

```



```
Выберите задачу для выполнения:
1. Проверка точек на попадание в круги (5 вар)
2. Вычисление произведения ряда (4 вар)
3. Вычисление суммы ряда (36 вар)
0. Выход
Ваш выбор: 1
Введите радиус внутреннего круга r1: 3
Введите радиус внешнего круга r2: 2
Введите расстояние a от центра внешнего круга до центров внутренних: 5
Введите количество точек: 1
Введите координаты x и y для точки 1: 2
3
Количество точек внутри малых кругов: 0

Выберите задачу для выполнения:
1. Проверка точек на попадание в круги (5 вар)
2. Вычисление произведения ряда (4 вар)
3. Вычисление суммы ряда (36 вар)
0. Выход
Ваш выбор:
```

Екран роботи програми показаний на рис 2.

Завдання 2 (Вар 4)

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.

б) 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.

4	$\prod_{k=1}^n \frac{(1-x)^{k+1} + 1}{(k-1)!}$

Вхідні дані:

Значение x (например: 0.5).

Количество членов ряда n (например: 4).

Вихідні дані:

--

$$\prod_{k=2,4,\dots} \frac{(1-x)^{k+1} + 1}{(k-1)!}$$

Алгоритм вирішення:

Ввести значение x и количество членов ряда n.

Инициализировать переменную для произведения как 1.0.

Для всех чётных значений k от 2 до n (с шагом 2):

Вычислить текущий член ряда по формуле:

$$(1-x)^{k+1} + 1 / (k-1)!$$

Где:

(1 - x) возводится в степень k + 1.

Факториал от (k - 1) вычисляется отдельно.

Умножить текущее значение произведения на этот член.

После завершения цикла вывести итоговое значение произведения.

```
// Задача 2: Произведение ряда (4 вар)
double factorial(int k) {
    double result = 1;
    for (int i = 1; i <= k; i++) result *= i;
    return result;
}

void task2() {
    int n;
    double x, product = 1.0;

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

    for (int k = 2; k <= n; k += 2) { // Члены с шагом 2
        double term = (pow(1 - x, k + 1) + 1) / factorial(k - 1);
        product *= term;
    }

    cout << "Произведение ряда: " << product << endl;
}
```

Рисунок 3- код (вар 4)

Лістинг коду :

```
// Задача 2: Произведение ряда (4 вар)
double factorial(int k) {
    double result = 1;
    for (int i = 1; i <= k; i++) result *= i;
    return result;
}

void task2() {
    int n;
    double x, product = 1.0;

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

    for (int k = 2; k <= n; k += 2) { // Члены с шагом 2
        double term = (pow(1 - x, k + 1) + 1) / factorial(k - 1);
        product *= term;
    }

    cout << "Произведение ряда: " << product << endl;
}
```



```
Выберите задачу для выполнения:
1. Проверка точек на попадание в круги (5 вар)
2. Вычисление произведения ряда (4 вар)
3. Вычисление суммы ряда (36 вар)
0. Выход
Ваш выбор: 2
Введите значение x: 3
Введите количество членов ряда n: 1
Произведение ряда: 1
```

Экран работы програми показаний на рис 4.

Завдання 3 (Вар 36) 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 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): n — кількість членів ряду (наприклад: 3).

Вихідні дані :

$$\text{Сумма} = \sum_{i=1}^n \frac{i!}{7 \cdot 12 \cdot 17 \dots (5i + 2)}$$

Алгоритм вирішення:

Ввести кількість членів ряду n .

Инициализировать переменную для суммы как 0.0.

Для каждого значения i от 1 до n :

Вычислить факториал $i!$.

Факториал — это произведение всех чисел от 1 до i .

Вычислить знаменатель как произведение чисел:

7, 12, 17, ..., $(5 * i + 2)$.

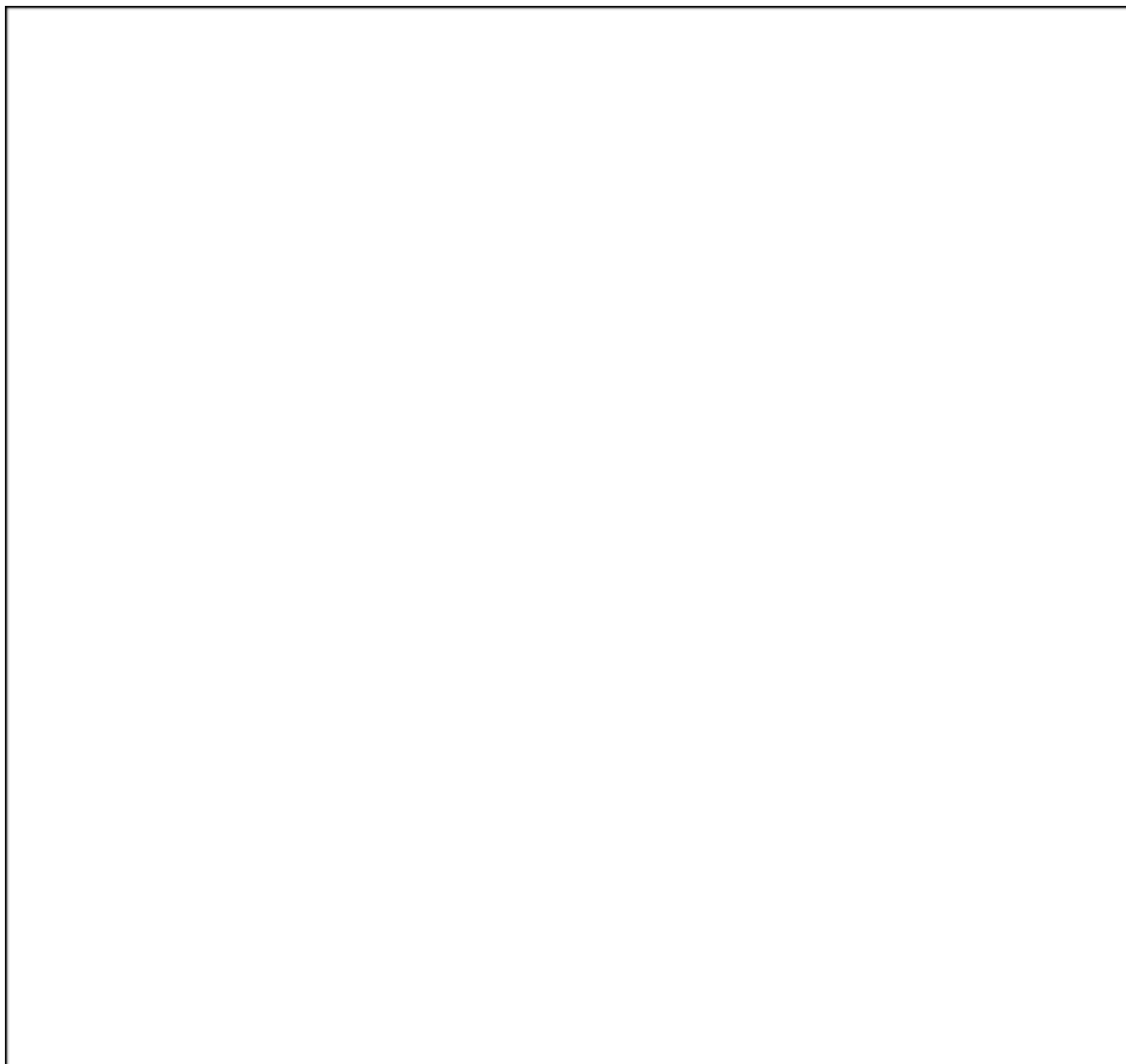
Это последовательность чисел с шагом 5, начиная с 7.

Найти текущий член ряда по формуле:

$i! / \text{знаменатель}$.

Добавить значение текущего члена ряда к переменной суммы.

После завершения цикла вывести итоговое значение суммы.



```

// Задача 3: Сумма ряда с факториалом (36 вар)
void task3() {
    int n;
    double sum = 0.0;

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

    for (int i = 1; i <= n; i++) {
        double denominator = 1;
        for (int j = 7; j <= (5 * i + 2); j += 5) { // последовательные множители
            denominator *= j;
        }
        sum += factorial(i) / denominator;
    }

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

// Главное меню программы
int main() {
    int choice;

    do {
        cout << "\nВыберите задачу для выполнения:\n";
        cout << "1. Проверка точек на попадание в круги (5 вар)\n";
        cout << "2. Вычисление произведения ряда (4 вар)\n";
        cout << "3. Вычисление суммы ряда (36 вар)\n";
        cout << "0. Выход\n";
        cout << "Ваш выбор: ";
        cin >> choice;

        switch (choice) {
            case 1: task1(); break;
            case 2: task2(); break;
            case 3: task3(); break;
            case 0: cout << "Выход из программы. До свидания!\n"; break;
            default: cout << "Ошибка: неверный выбор. Попробуйте снова.\n";
        }
    } while (choice != 0);

    return 0;
}

```

Рисунок 5- код (вар 36) .

Лістинг коду :

```

// Задача 3: Сумма ряда с факториалом (36 вар)
void task3() {
    int n;
    double sum = 0.0;

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

    for (int i = 1; i <= n; i++) {

```



```

        double denominator = 1;
        for (int j = 7; j <= (5 * i + 2); j += 5) { // последовательные множители
            denominator *= j;
        }
        sum += factorial(i) / denominator;
    }

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

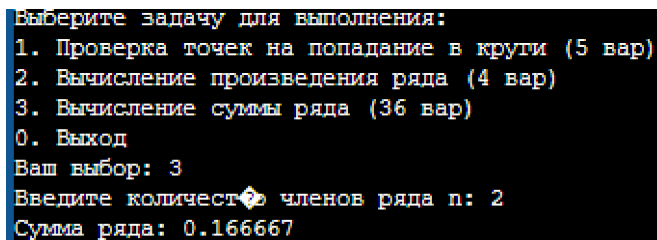
// Главное меню программы
int main() {
    int choice;

    do {
        cout << "\nВыберите задачу для выполнения:\n";
        cout << "1. Проверка точек на попадание в круги (5 вар)\n";
        cout << "2. Вычисление произведения ряда (4 вар)\n";
        cout << "3. Вычисление суммы ряда (36 вар)\n";
        cout << "0. Выход\n";
        cout << "Ваш выбор: ";
        cin >> choice;

        switch (choice) {
            case 1: task1(); break;
            case 2: task2(); break;
            case 3: task3(); break;
            case 0: cout << "Выход из программы. До свидания!\n"; break;
            default: cout << "Ошибка: неверный выбор. Попробуйте снова.\n";
        }
    } while (choice != 0);

    return 0;
}

```



```
Выберите задачу для выполнения:  
1. Проверка точек на попадание в круги (5 вар)  
2. Вычисление произведения ряда (4 вар)  
3. Вычисление суммы ряда (36 вар)  
0. Выход  
Ваш выбор: 3  
Введите количество членов ряда n: 2  
Сумма ряда: 0.166667
```

Екран роботи програми показаний на рис 6.

ВИСНОВКИ

У результаті виконання задачі було обчислено суму ряду, в якому чисельник містить факторіал, а знаменник є добутком послідовних чисел із кроком 5. Результати показали, що значення суми швидко зменшується зі збільшенням кількості членів ряду завдяки зростанню знаменника. Це підтверджує збіжність ряду та ефективність використаного алгоритму.