

Лабораторна робота № 5

з дисципліни «Алгоритмізація та програмування»
на тему "Реалізація циклічних алгоритмів мовою С ++"
ХАІ.301. 310 група, 7 номер в списку ЛР

Виконала студентка гр. 310

Стеценко

СофіяОлександрівна

(підпис, дата)

(П.И.Б.)

Перевірів к.т.н., доц. Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

Харків 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.

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) - 30 варіант на Рис.1.

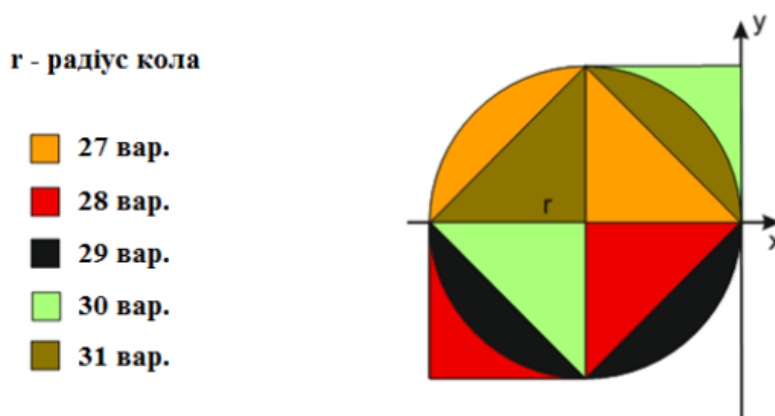


Рис.1 - Завдання 1

Завдання 2.

Дано дійсне число x і натуральне число n . Необхідно:

- а) Обчислити значення виразу при заданих x і n для виразу на Рис. 2.
- б) Вивести: для парних варіантів – значення кожного третього елемента, для непарних – значення кожного четвертого елемента.

$$\sum_{k=0}^n \frac{k(k+1) - x^k}{x^{2k+1}}, 2 \leq x \leq 5$$

Рис.2 - 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 збіжного ряду

($\epsilon = 10^{-5} \dots 10^{-20}$); g – величина для переривання циклу обчислення суми розбіжного ряду ($g = 10^2 \dots 10^5$). Варіант представлений на Рис. 3.

$$\sum_{n=1}^{\infty} \frac{5^n}{\sqrt{n} 2^{n-1}}$$

Рис.3 - Завдання 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).

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі geom30.

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

Дійсні числа – координати точок на площині.

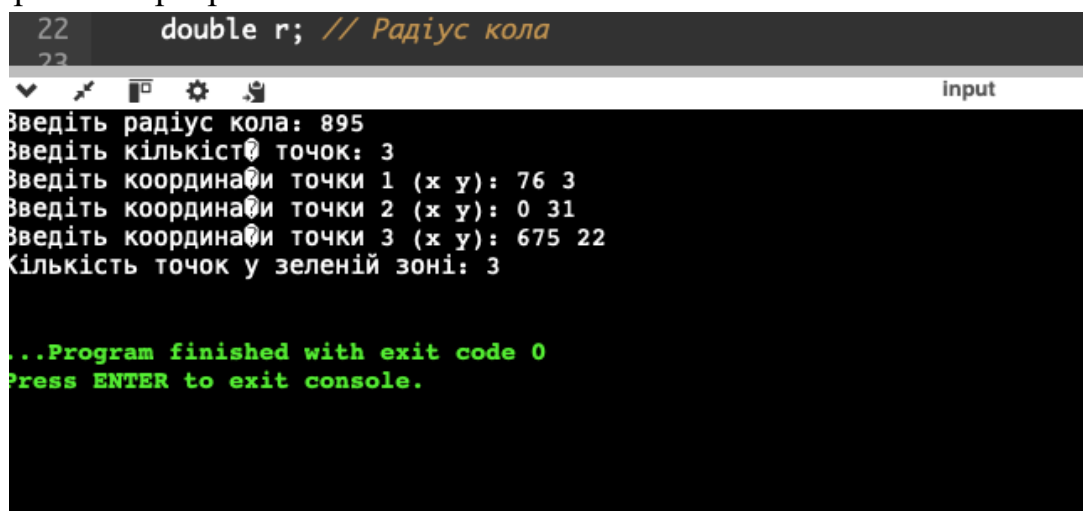
Вихідні дані (ім'я, опис, тип):

Ціле число - кількість точок, що потрапляють в фігуру заданого кольору.

Алгоритм вирішення представлено в дод. А (стор. 7).

Лістинг коду вирішення задачі наведено в дод. Б (стор. 8).

Екран роботи програми показаний на Рис. 4.



```

22 double r; // Радіус кола
23
Введіть радіус кола: 895
Введіть кількість точок: 3
Введіть координати точки 1 (x y): 76 3
Введіть координати точки 2 (x y): 0 31
Введіть координати точки 3 (x y): 675 22
Кількість точок у зеленій зоні: 3

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

Рис. 4 - Екран роботи програми до завдання 1

Завдання 2.

Вирішення задачі 2.

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

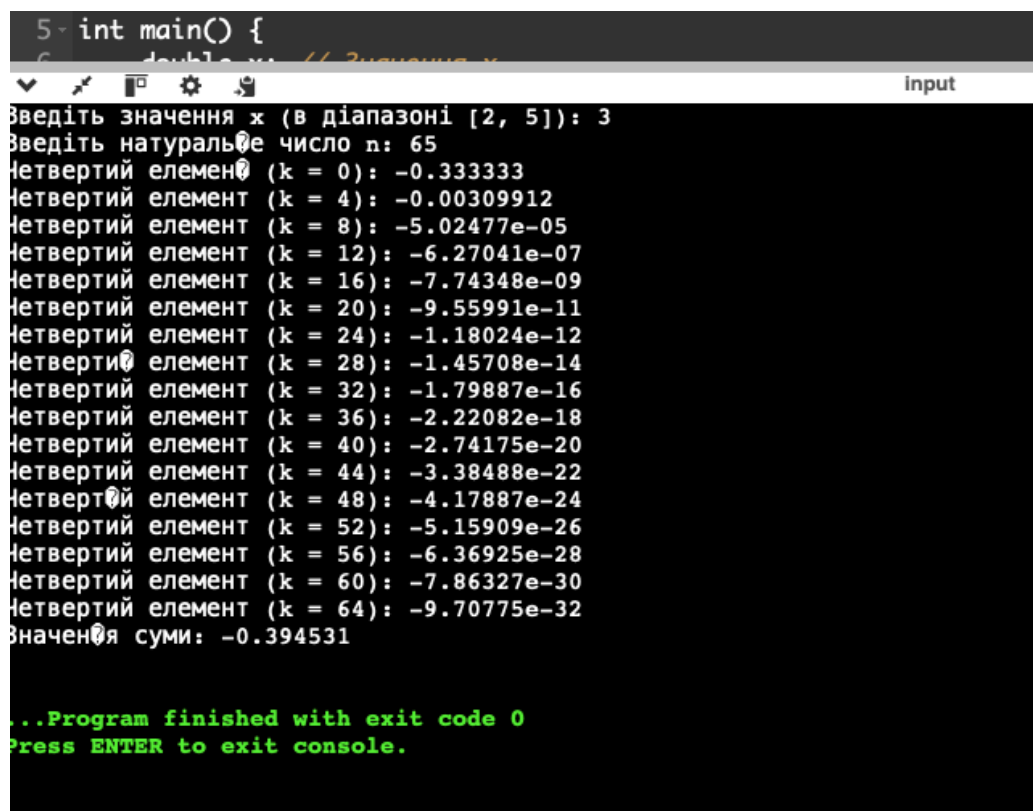
Дійсне число x і натуральне число n .

Вихідні дані (ім'я, опис, тип):

Число - для парних варіантів – значення кожного третього елемента, для непарних – значення кожного четвертого елемента.

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

Екран роботи програми показаний на Рис.5 та Рис.6.



```

5 int main() {
6     double x, // Значення x
7     int n;
8     // Введення даних
9     cout << "Введіть значення x (в діапазоні [2, 5]): ";
10    cin >> x;
11    cout << "Введіть натуральне число n: ";
12    cin >> n;
13    // Виведення результатів
14    for (int k = 0; k < n; k++) {
15        if (k % 4 == 0) {
16            cout << "Четвертий елемент (k = " << k << "): " << x << endl;
17        }
18        if (k % 3 == 0) {
19            cout << "Четвертий елемент (k = " << k << "): " << x << endl;
20        }
21    }
22    // Виведення суми
23    cout << "Значення суми: " << x << endl;
24    return 0;
25 }

```

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

Рис.5 - Екран роботи програми до Завдання 2

```

5 int main() {
6     double x; // Значення x
7     // Введіть значення x (в діапазоні [2, 5]):
8     // Значення x повинні бути в діапазоні [2, 5].
9
10    ...Program finished with exit code 1
11    Press ENTER to exit console.

```

Рис.6 - Екран роботи програми до Завдання 2

Завдання 3.

Вирішення задачі 3.

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

Розбіжний ряд.

Вихідні дані (ім'я, опис, тип):

Алгоритм вирішення показано в дод. Г (стор. 12).

Лістинг коду вирішення задачі наведено в дод. Д (стор. 13).

Екран роботи програми показаний на Рис. 6.

```

22
23
// Ряд розбіжний. Сума перевищує поріг g.
// Кількість членів ряду: 13
24
25
...Program finished with exit code 0
Press ENTER to exit console.

```

Рис.6 - Екран роботи програми до Завдання 3

Завдання 4

Алгоритм вибору користувачем завдання

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

Попередні завдання

Вихідні дані (ім'я, опис, тип):

Програма, що дозволяє користувачу переключатися між завданнями.

Лістинг коду вирішення наведено в дод. Е (стор. 14) .

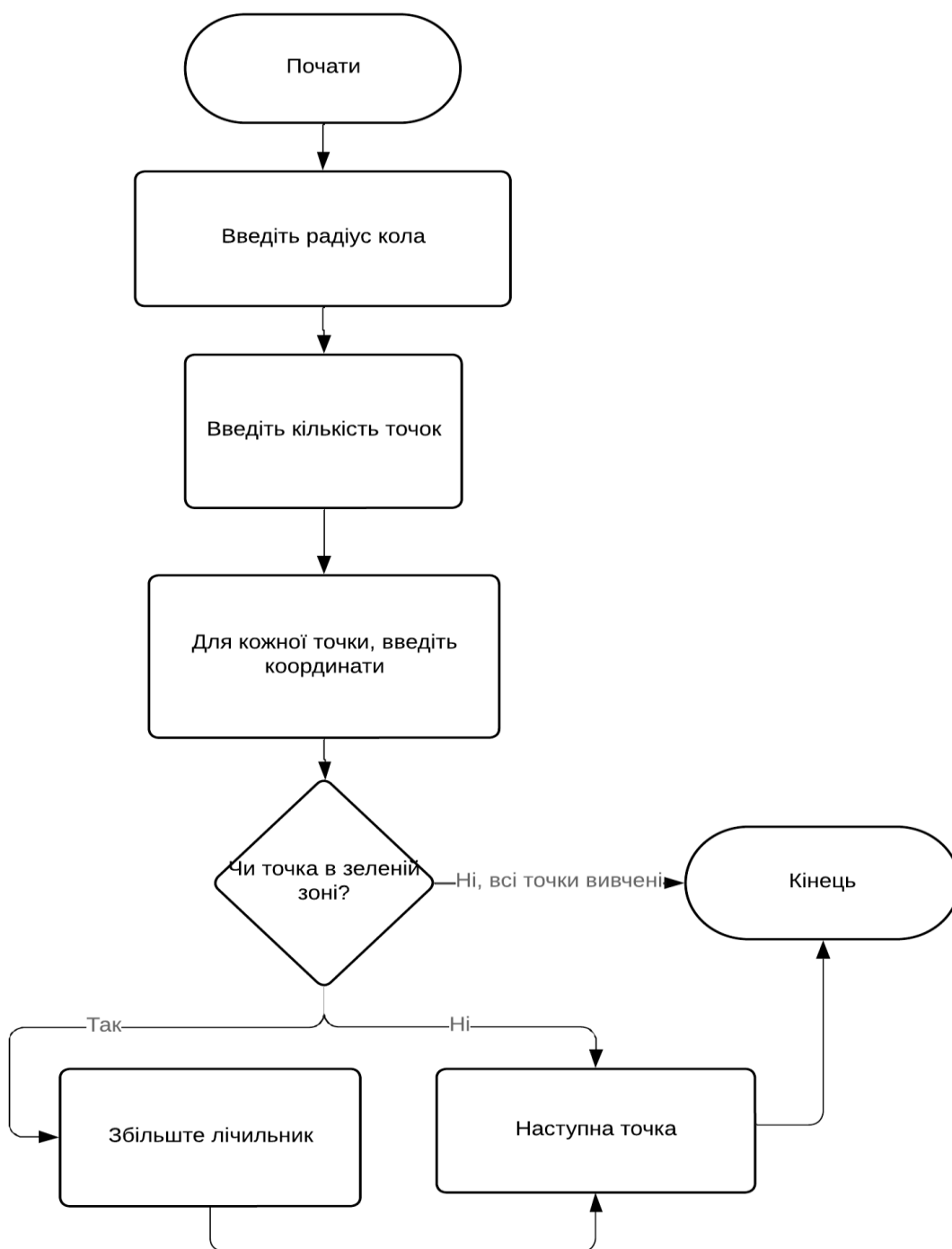
ВИСНОВКИ

У ході лабораторної роботи було вивчено теоретичний матеріал із синтаксису мови C++ та закріплено основи побудови UML-діаграм для моделювання циклічних алгоритмів. Було розглянуто та реалізовано алгоритми з використанням: циклу з передумовою (while), циклу з післяумовою (do-while). Реалізація дала змогу ще раз закріпити знання щодо синтаксису мови програмування C++ та засвоїти основи роботи з інструкціями циклу. UML-діаграми допомогли краще зрозуміти логіку алгоритмів і їхню реалізацію.

Харків 2024

Додаток А

Алгоритм вирішення завдання 1



Додаток Б
Лістинг коду програми виконання завдання 1

// Функція для перевірки, чи точка знаходиться в зеленій області

```
bool isInGreenArea(double x, double y, double r) {
    // Точка повинна бути в межах квадранта  $x > 0, y > 0$ 
    if (x >= 0 && y >= 0) {
        // Перевірка, чи точка всередині кола
        if (x * x + y * y <= r * r) {
            // Перевірка, чи точка поза правим трикутником
            if (y <= r - x) {
                return true;
            }
        }
    }
    return false;
}
```

```
int main() {
    int n; // Кількість точок
    double r; // Радіус кола

    cout << "Введіть радіус кола: ";
    cin >> r;
    cout << "Введіть кількість точок: ";
    cin >> n;

    int count = 0; // Лічильник точок у зеленій зоні

    for (int i = 0; i < n; i++) {
        double x, y;
        cout << "Введіть координати точки " << i + 1 << " (x y): ";
        cin >> x >> y;

        if (isInGreenArea(x, y, r)) {
            count++;
        }
    }

    cout << "Кількість точок у зеленій зоні: " << count << endl;

    return 0;
}
```



```
}
```

Додаток В

Лістинг коду програми виконання завдання 2

```
// Завдання 2 реалізація  
void task2() {
```

```

double x; // Значення x
int n;    // Натуральне число n

cout << "Введіть значення x ( $2 \leq x \leq 5$ ): ";
cin >> x; // Введення значення x
if (x < 2 || x > 5) { // Перевірка на коректність x
    cout << "Значення x повинно бути в діапазоні [2, 5]." << endl;
    return; // Завершуємо функцію у разі некоректного введення
}

cout << "Введіть натуральне число n: ";
cin >> n; // Введення натурального числа n
if (n < 1) { // Перевірка на коректність n
    cout << "n повинно бути натуральним числом." << endl;
    return; // Завершуємо функцію у разі некоректного введення
}

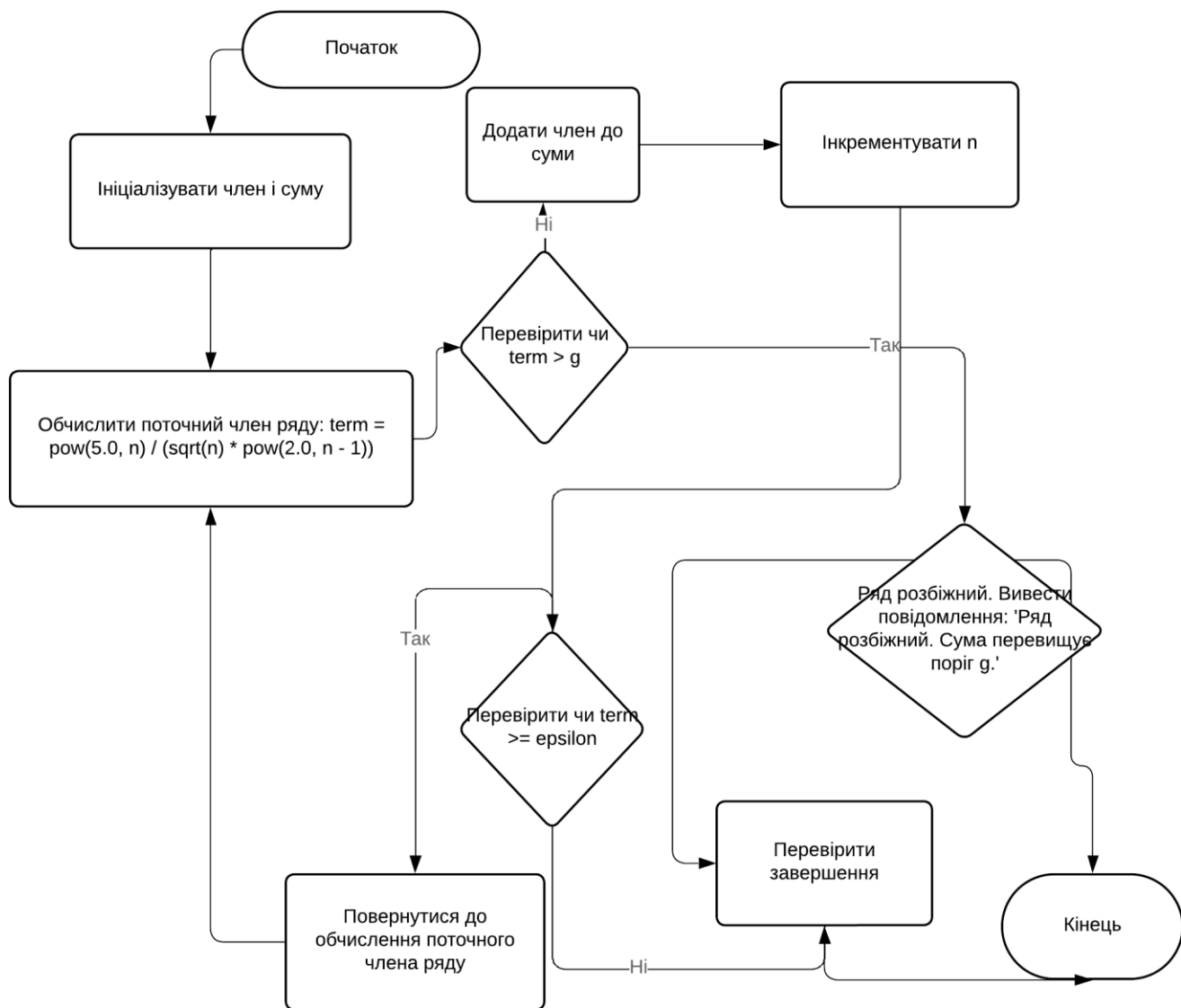
double sum = 0; // Змінна для обчислення суми
for (int k = 0; k <= n; k++) { // Цикл для обчислення членів ряду
    double numerator = k * (k + 1) - pow(x, k); // Чисельник
    double denominator = pow(x, 2 * k + 1);    // Знаменник
    double term = numerator / denominator;     // Поточний член ряду
    sum += term;                               // Додавання члена до суми

    // Виведення кожного третього або четвертого елемента в залежності від
    // парності n
    if (n % 2 == 0 && k % 3 == 0) {
        cout << "Третій елемент (k = " << k << "): " << term << endl;
    }
    if (n % 2 != 0 && k % 4 == 0) {
        cout << "Четвертий елемент (k = " << k << "): " << term << endl;
    }
}

cout << "Значення суми: " << sum << endl; // Виведення результату суми
}

```

Додаток Г
Алгоритм вирішення завдання 3



Додаток Д
Лістинг коду завдання 3

// Виконання завдання 3

```
void task3() {
    const double epsilon = 1e-10; // Мала величина для збіжності ряду ( $\epsilon$ )
    const double g = 1e5;        // Поріг для розбіжності ряду (g)
    double sum = 0.0;            // Поточна сума ряду
    double term;                 // Поточний член ряду (un)
    int n = 1;                   // Лічильник номеру члена ряду

    do {
        term = pow(5.0, n) / (sqrt(n) * pow(2.0, n - 1)); // Обчислення поточного члена
        // ряду

        if (fabs(term) > g) {
            cout << "Ряд розбіжний. Сума перевищує поріг g." << endl;
            break; // Завершуємо цикл у разі розбіжності ряду
        }

        sum += term; // Додавання члена до суми
        n++;        // Інкремент номеру члена ряду
    } while (fabs(term) >= epsilon); // Умова завершення циклу

    if (fabs(term) < epsilon) {
        cout.precision(10); // Встановлення точності виводу результату
        cout << "Ряд збіжний. Сума: " << sum << endl;
    }

    cout << "Кількість членів ряду: " << n - 1 << endl; // Виведення кількості членів
    // ряду
}
```

Додаток Е

Лістинг коду виконання завдання 4

```
int main() {
    int choice; // Змінна для зберігання вибору користувача
```

```
do {  
    cout << "Оберіть завдання: ";  
    cin >> choice; // Вибір користувача  
  
    switch(choice) {  
        case 1:  
            task_geom30(); // Виклик першого завдання  
            break;  
        case 2:  
            task2(); // Виклик другого завдання  
            break;  
        case 3:  
            task3(); // Виклик третього завдання  
            break;  
        default:  
            cout << "Неправильний вибір. Спробуйте ще раз.\n";  
            break;  
    }  
  
} while(choice != 4); // Цикл продовжується, поки користувач не вибере вихід  
  
return 0;  
}
```

Додаток Ж
Діаграма активності цілого коду

