

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»  
на тему «"Структурування програм з використанням функцій"»

XAI.301.173.310.1 ЛР

Виконав студент гр. \_\_\_\_\_310\_\_\_\_\_

28.11.2024 Андрій КОБИЛЯНСЬКИЙ  
(підпис, дата) (П.І.Б.)

Перевірив

\_\_\_\_\_ к.т.н., доц. Олена ГАВРИЛЕНКО  
(підпис, дата) (П.І.Б.)

2024

## МЕТА РОБОТИ

Вивчити теоретичний матеріал із синтаксису оголошення, визначення і виклику функцій в C++ і реалізувати консольний додаток з використанням функцій з параметрами і поверненням результату на мові програмування C++ в середовищі Visual Studio.

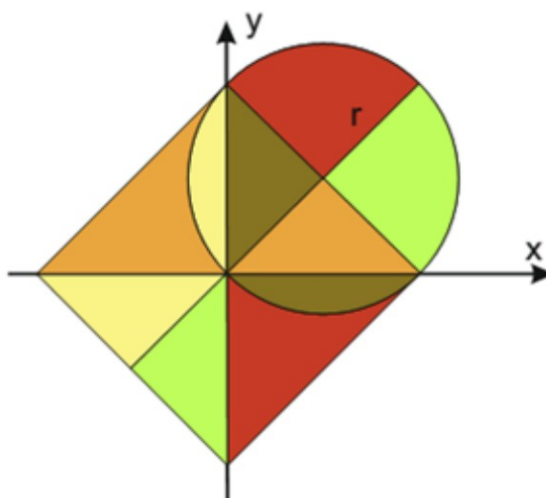
## ПОСТАНОВКА ЗАДАЧІ

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).

Табл.1. Варіант 35, жовтий колір

**r - радіус кола**

- 32 вар.**
- 33 вар.**
- 34 вар.**
- 35 вар.**
- 36 вар.**



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.

Таблиця 2, варіант 2

$$\frac{(x-2)(x-4)\dots(x-2^n)}{(x-1)(x-3)\dots(x-(2^n-1))}$$

Завдання 3. 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 ( $g = 10^2 \dots 10^5$ ).

Таблиця 3, варіант 36

$$\sum_{n=0}^{\infty} \frac{2 * 5 * 8 \dots (3n+2)}{x^n (2n+1)!}$$

Завдання 4. Для вибору користувачем одного з трьох зазначених вище завдань організувати меню з використанням інструкції вибору. Кожне завдання має бути реалізовано у вигляді окремої процедури (функції без параметрів), що містить необхідні оголошення змінних і виклики інших функцій.

## ВИКОНАННЯ РОБОТИ

### Завдання 1

#### Вирішення задачі No35

Вхідні данні (ім'я, опис, тип, обмеження)

Geom 35:  $x, y$  — координати точок, дійсний тип із подвійною точністю.  $R$  — радіус кола, дійсний тип із подвійною точністю.

Визідні данні ((ім'я, опис, тип)

Програма рахує, скільки введених точок знаходяться в площі і виводить результат в консоль.

Алгоритм вирішення:

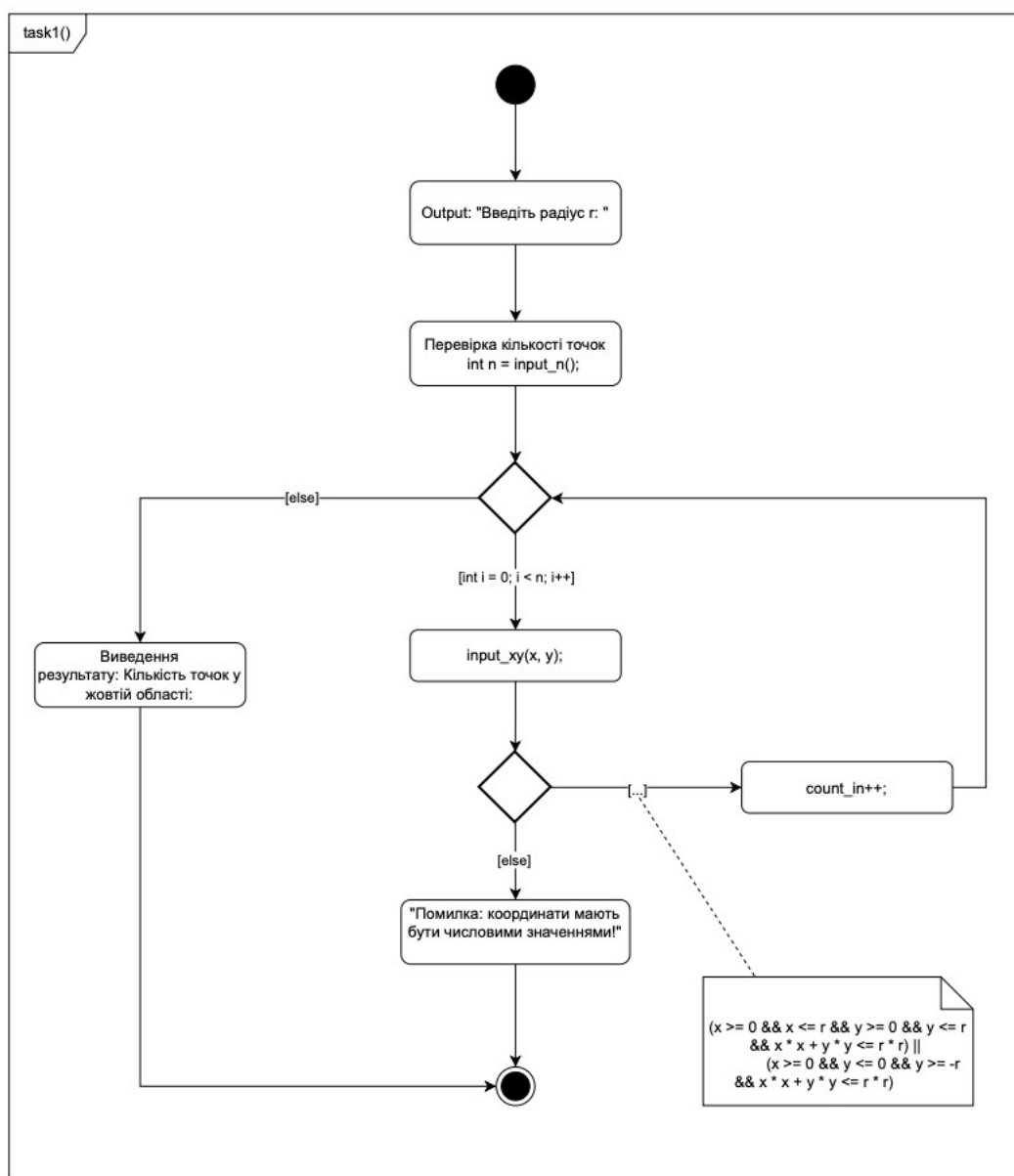


Рисунок 1. Діаграма активності для алгоритму вирішення задачі Geom35 з циклом.

Екран роботи програми показаний на рис. 2

```

_____Task 1 _____
Введіть радіус r: 32
Введіть кількість точок (n): 3
Введіть координати точки (x, y): 2
-5
Введіть координати точки (x, y): 12
23
Введіть координати точки (x, y): 11
-6
Кількість точок у жовтій області: 3
  
```

## Завдання 2

### Вирішення задачі No2

Вхідні данні (ім'я, опис, тип, обмеження)

x — дійсн число, дійсний тип із подвійною точністю. дійсний тип із подвійною точністю.

n — натуральне число, дійсний тип

Вихідні данні ((ім'я, опис, тип)

Кожен четвертий елемент (якщо умова виконується)

Алгоритм вирішення:

- 1) Виведення запрошення до вводу;
- 2) Введення змінної x, n;
- 3) Перевірка коректності введення;
- 4) Обчислення поточного значення виразу;
- 5) Виведення значення в залежності від варіанту;
- 6) Виведення результату.

Лістинг коду вирішення задачі if35 наведено в дод. А (стор. 5).

Екран роботи програми показаний на рис. 3

```
_____Task 2_____
Enter x and n (x as a real number, n as an integer): 32
12
Element 3: 0.96
Element 6: 1.03226
Element 9: 1.00209
Element 12: 1.00025
```

## Завдання 3

### Вирішення задачі No37

Вхідні данні (ім'я, опис, тип, обмеження)

$x$  — дійсне число, дійсний тип із подвійною точністю.

$e = 1E-20$  — дійсне число, дійсний тип із подвійною точністю.

$u1 = 2, u2 = 1, sum = 0$  — дійсне число, дійсний тип із подвійною точністю.

$n = 1$  — ціле число, цілий тип

Вихідні данні (ім'я, опис, тип)

$sum$  — дійсне число, дійсний тип із подвійною точністю.

Лістинг коду вирішення задачі calculate37 наведено в дод. А (стор. 5).

Алгоритм вирішення:

Екран роботи програми показаний на рис. 4

```
-----Task 3-----  
Enter value for x: 12  
u1 = 2  
u2 = 0.138889  
u3 = 0.00462963  
u4 = 0.000101044  
u5 = 1.63728e-06  
u6 = 2.10862e-08  
u7 = 2.2528e-10  
u8 = 2.05612e-12  
u9 = 1.63784e-14  
u10 = 1.15735e-16  
u11 = 7.34822e-19  
Last u: 4.23564e-21  
Sum: 2.14362
```

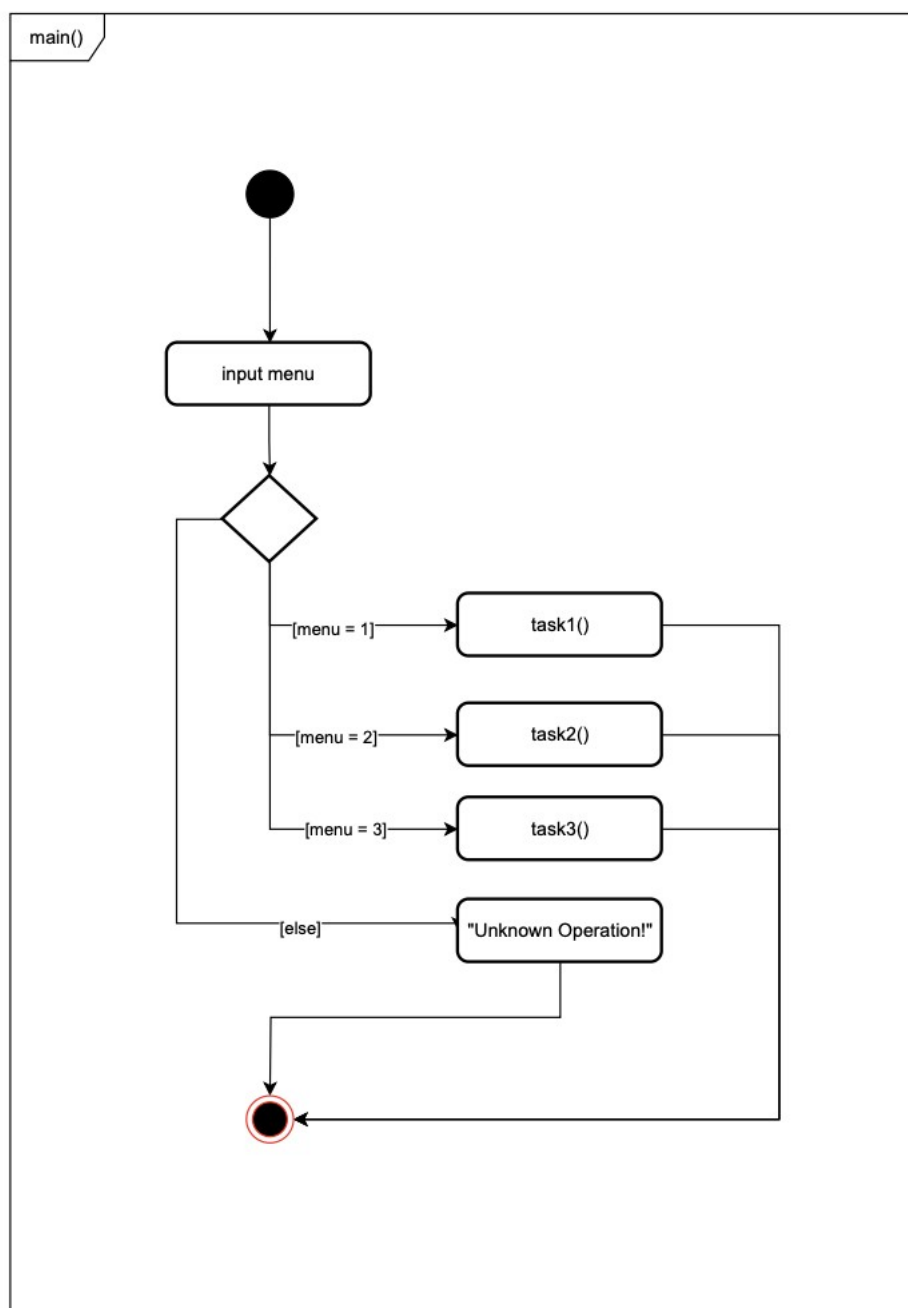


Рисунок 5. Діаграма активності для усієї програми

## ВИСНОВКИ

Було освоєно алгоритми з використання інструкцій циклу з передумовою, циклу післямовою і параметризованого циклу мовою C++ в середовищі Visual Studio. Виведення значення кожного четвертого елемента, було досліджено ряд на збіжність, введення величини для переривання циклу обчислення суми збіжного або розбіжного



## ДОДАТОК А

### Лістинг коду програми.

```
#include <iostream>
#include <cmath>

using namespace std;

void task1();

void task2();

void task3();

int main()
{
    int menu;
    cout << "\nEnter task number: ";
    cin >> menu;
    switch (menu)
    {
        case 1 : task1(); break; //1st Task

        case 2 : task2(); break; //2st Task

        case 3 : task3(); break; //3st Task

        default : cout << "\nUnknown operation!";

    }

    return 0;
}

// Введення координат точки
void input_xy(float& x, float& y)
{
    while (true) {
        cout << "Введіть координати точки (x, y): ";
        cin >> x >> y;
        if (cin.fail()) {
            cout << "Помилка: координати мають бути числовими значеннями!" <<
endl;
            cin.clear();
            cin.ignore(32767, '\n');
```

```

        } else {
            break;
        }
    }
}

// Введення кількості точок
int input_n() {
    int n;
    do {
        cout << "Введіть кількість точок (n): ";
        cin >> n;
        if (cin.fail() || n < 1 || n > 1000) {
            cout << "Помилка: n має бути цілим числом у межах [1, 1000]." <<
endl;
            cin.clear();
            cin.ignore(32767, '\n');
        } else {
            break;
        }
    } while (true);
    return n;
}

```

```

void task1()
{
    cout << "\n_____Task 1 _____" << endl << endl;
    float r, x, y;
    int count_in = 0;

    // Введення радіусу фігури
    cout << "Введіть радіус r: ";
    cin >> r;

    // Перевірка кількості точок
    int n = input_n();

    // Обробка кожної точки
    for (int i = 0; i < n; i++)
    {
        input_xy(x, y);

        // Жовта область (згідно з варіантом 35)
        if (
            (x >= 0 && x <= r && y >= 0 && y <= r && x * x + y * y <= r * r)
        || // Внутрішня чверть кола

```

```

                (x >= 0 && y <= 0 && y >= -r && x * x + y * y <= r *
r)                // Нижня права чверть
            ) {
                count_in++;
            }
        }

        // Виведення результату
        cout << "Кількість точок у жовтій області: " << count_in << endl;
    }

```

/\*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.\*/

```

void input_xy(double &x, int &n) {
    // Введення значень x та n з перевіркою
    while (true) {
        cout << "Enter x and n (x as a real number, n as an integer): ";
        cin >> x >> n;
        if (cin.fail() || n <= 0) {
            cout << "Invalid input. Please try again!" << endl;
            cin.clear();
            cin.ignore(32767, '\n');
        } else {
            break;
        }
    }
}

```

```

void task2()
{
    cout << "\n_____Task 2_____ " << endl << endl;

    // Змінні
    int n;
    double x, prod = 1.0, ps = 0;

    // Введення значень
    input_xy(x, n);

    // Обчислення виразу
    for (int j = 1; j <= n; j++) {
        // Перевірка ділення на 0
    }
}

```

```

        if (x - (pow(2, j) - 1) == 0) {
            cout << "Division by 0!" << endl;
            break;
        }

        // Обчислення поточного значення виразу
        ps = (x - pow(2, j)) / (x - (pow(2, j) - 1));
        prod *= ps;

        // Виведення значення в залежності від варіанту
        if (n % 2 == 0 && j % 3 == 0) {
            cout << "Element " << j << ": " << ps << endl; // Для парних
варіантів
        }
        else if (n % 2 != 0 && j % 4 == 0) {
            cout << "Element " << j << ": " << ps << endl; // Для непарних
варіантів
        }
    }

    // Виведення результату
    cout << "\nTotal product: " << prod << endl;
}

/*Завдання 3. 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 ( $g = 10^2 \dots 10^5$ )
).*/

// Функція для обчислення факторіалу
double factorial(int n)
{
    double result = 1;
    for (int i = 1; i <= n; ++i)
    {
        result *= i;
    }
    return result;
}

// Функція для виконання завдання 3
void task3()
{
    cout << "\n_____Task 3_____ " << endl << endl;
    double x, u1 = 2, u2 = 1, u, sum = 0; // змінні для елементів та суми
    int n = 1;

```

```

double e = 1E-20; // мале число для порівняння

// Введення x з перевіркою на правильність
cout << "Enter value for x: ";
while (true)
{
    cin >> x;
    if (!cin)
    {
        cout << "Invalid input! Please enter a valid number for x: ";
        cin.clear();
        cin.ignore(32767, '\n');
    }
    else
    {
        break;
    }
}

u = u1 / u2; // перший елемент
sum += u; // додаємо до суми

// Обчислення елементів та їх суми
while (fabs(u) > e)
{
    cout << "u" << n << " = " << u << endl; // виведення елемента
    // Обчислення наступного елемента
    u1 *= 3 * n + 2;
    u2 = pow(x, n) * factorial(2 * n + 1);
    if (u2 == 0) {
        cout << "Division by zero!" << endl;
        break;
    }
    u = u1 / u2; // обчислення нового елемента
    sum += u; // додаємо до суми
    n++;
    if (n > 200) break; // обмеження кількості елементів
}

// Виведення останнього елемента та суми
cout << "Last u: " << u << endl;
cout << "Sum: " << sum << endl;
}

```

ДОДАТОК Б  
Скрін-шоти вікна виконання програми

```
_____Begin 3_____
side a = 11
side b = 24
Area of a rectangle = 264
sh: pause: command not found
Perimeter of a rectangle = 70
sh: pause: command not found

_____Begin 11_____
R = 23.7
L = 148.836
S = 1763.71
○ macbook@mbp-macbook-2 Study %
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання  
Begin 3, Begin 11.