

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 «Реалізація циклічних алгоритмів мовою C ++»

XAI.301. 141. 319a. 16 ЛР

Виконав студент гр. 319a

Моїсеєнко Євген  
(підпис, дата) (П.І.Б.)

Перевішив

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

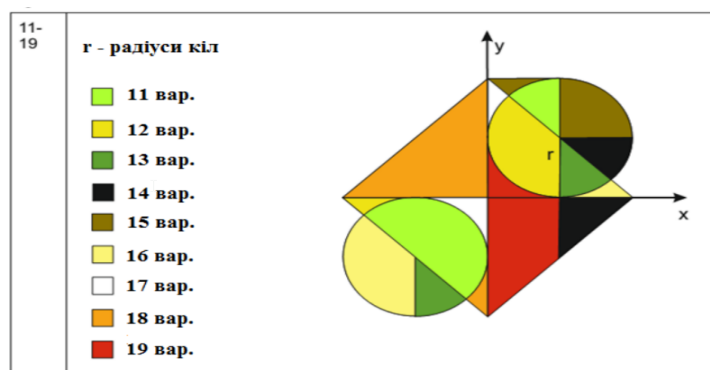
2023

## 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).



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

- Обчислити значення виразу при заданих  $x$  і  $n$  для виразу з табл.2.
- Вивести: для парних варіантів – значення кожного третього елемента, для непарних – значення кожного четвертого елемента.

20	$-\sum_{k=1}^n \frac{(-1)^k (-1 + \cos(x))^k}{k}, \quad  -1 + \cos(x)  < 1$
----	---

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

$$\sum_{n=1}^{\infty} \frac{(3n-2)!}{100 * 102 * \dots (98+2n)}$$

## ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Геометричні області 12

Вхідні дані:

point.x- Тип: int, Координати точки, які перевіряються щодо їхнього розташування відносно фігури.

point.y- тип: int, координати точки, які перевіряються щодо їхнього розташування відносно фігури.

color- тип: int, колір фігури, яка порівнюється з кольорами у масиві фігур.

figures- Тип: Figure[], масивсив структур Figure, який містить дані про різні фігури у випадку одну фігуру з властивостями колір, центр та радіус.

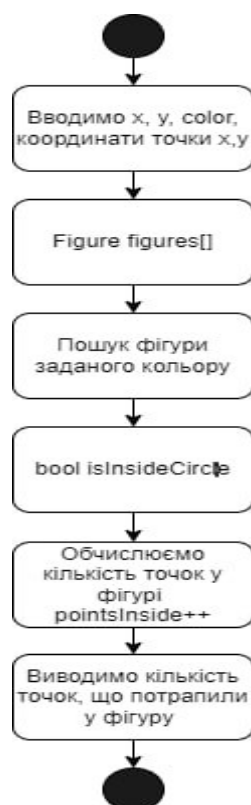
tempPoint.x- тип: int, Координати кожної точки, що перевіряється на потрапляння всередину фігури.

tempPoint.y- тип: int, Координати кожної точки, що перевіряється на потрапляння всередину фігури.

Вихідні дані:

pointsInside- Тип: int, кількість точок, які потрапили в фігуру.

Алгоритм вирішення показано на рис.



## Рисунок 1 – Геометричні області

Лістинг коду вирішення задачі Геометричні області 12 наведено в дод. А (стор. 7-10).

Екран роботи програми показаний на рис. Б.1.

### Завдання 2.

Вирішення задачі Підрахунок рядів 20

Вхідні дані:

x- тип: double, дійсне число.

n- тип: int, ціле число.

Вихідні дані:

sum- тип- double, дійсне число, результат обчислення виразу

Алгоритм вирішення показано на рис.

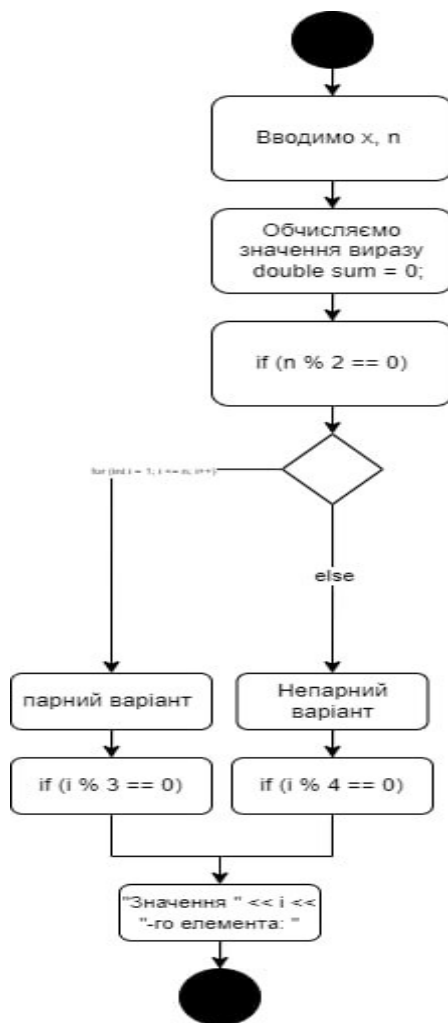


Рисунок 2 – Підрахунок рядів

Лістинг коду вирішення задачі Підрахунок рядів 20 наведено в дод. А (стор. 7-10).

Екран роботи програми показаний на рис. Б.2.

Завдання 3.

Вирішення задачі Збіжність/розбіжність рядів 5

Вхідні дані:

n- тип: int, кількість ітерацій для розрахунку суми ряду.

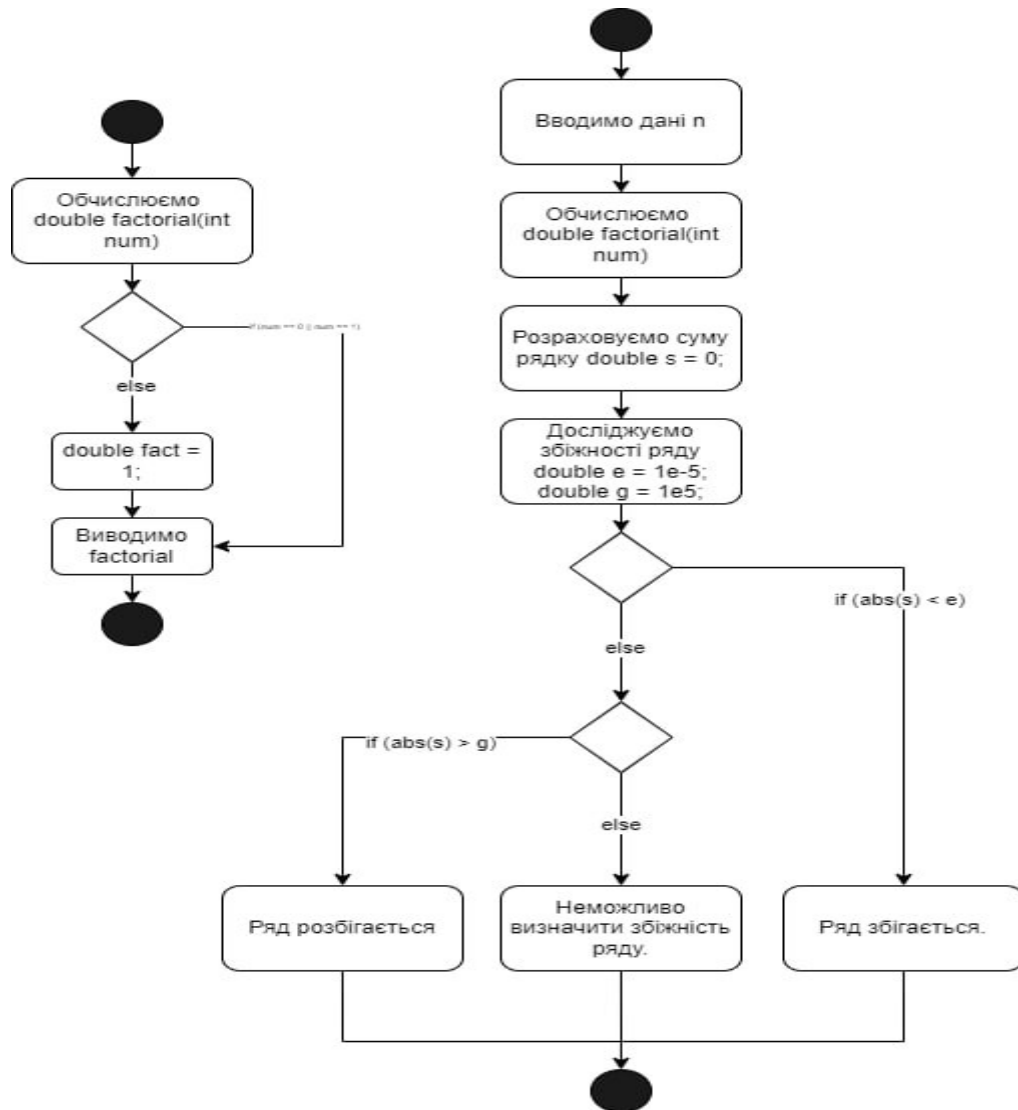
Вихідні дані:

s - тип: double, значення суми ряду.

e -тип: double, - мала величина для збіжного ряду (встановлена на  $1e-5$ ).

g -тип: double, - велика величина для розбіжного ряду (встановлена на  $1e5$ ).

Алгоритм вирішення показано на рис.



Лістинг коду вирішення задачі Збіжність/розбіжність рядів 5 наведено в дод. А (стор. 7-10).

Екран роботи програми показаний на рис. Б.3.

## ВИСНОВКИ

У роботі було вивчено теоретичний матеріал щодо синтаксису мови C++, включаючи представлення циклічних алгоритмів за допомогою UML-діаграм. Реалізація цих алгоритмів відбулася за допомогою інструкцій циклів з передумовою, післяумовою і параметризованих циклів в середовищі Visual Studio. Це дозволило практично застосовувати знання та розуміння синтаксису мови C++, зокрема циклічних конструкцій, для створення функціональних алгоритмів.

## ДОДАТОК А

### Лістинг коду програми

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

using namespace std;

// Структура, яка представляє точку у двовимірному просторі
struct Point {
    int x; // Координата x
    int y; // Координата y
};

// Структура, яка представляє фігуру - коло
struct Figure {
    int color; // Колір фігури
    Point center; // Центр кола
    int radius; // Радіус кола
};

// Функція, яка перевіряє, чи знаходиться точка всередині кола
bool isInsideCircle(Point point, Figure figure) {
    return pow(point.x - figure.center.x, 2) + pow(point.y - figure.center.y, 2)
    <= pow(figure.radius, 2);
}

// Функція, яка обчислює факторіал числа
double factorial(int num) {
    if (num == 0 || num == 1) {
        return 1;
    } else {
        double fact = 1;
        for (int i = 2; i <= num; ++i) {
            fact *= i;
        }
        return fact;
    }
}

// Функція для виконання першого завдання
void taskOne() {
    cout << "Ви обрали завдання 1" << endl;

    Point point;
    cout << "Введіть координати точки: ";
    cin >> point.x >> point.y;
```

```

int color;
cout << "Введіть колір фігури: ";
cin >> color;

// Масив фігур, зараз у ньому є лише одна фігура - коло
Figure figures[] = {
    {12, {5, 5}, 5},
};

int i;
for (i = 0; i < 1; i++) {
    if (figures[i].color == color) {
        break;
    }
}

int pointsInside = 0;

for (int j = 0; j < 5; j++) {
    Point tempPoint;
    cout << "Введіть координати точки " << j + 1 << ": ";
    cin >> tempPoint.x >> tempPoint.y;

    bool isInside = isInsideCircle(tempPoint, figures[i]);
    if (isInside) {
        pointsInside++;
    }
}

cout << "Кількість точок, що потрапили у фігуру: " << pointsInside << endl;
}

// Функція для виконання другого завдання
void taskTwo() {
    cout << "Ви обрали завдання 2" << endl;

    double x;
    int n;

    cout << "Введіть дійсне число x: ";
    cin >> x;

    cout << "Введіть натуральне число n: ";
    cin >> n;

    double sum = 0;
    for (int i = 1; i <= n; i++) {
        sum += pow(-1, i) * pow(-1 + cos(x), i) / i;
    }
}

```



```

cout << "Значення виразу: " << sum << endl;

// Друк коефіцієнтів, що обчислюються на кожній ітерації циклу залежно від
умови
if (n % 2 == 0) {
    for (int i = 1; i <= n; i++) {
        if (i % 3 == 0) {
            cout << "Значення " << i << "-го елемента: " << sum / i << endl;
        }
    }
} else {
    for (int i = 1; i <= n; i++) {
        if (i % 4 == 0) {
            cout << "Значення " << i << "-го елемента: " << sum / i << endl;
        }
    }
}
}

// Функція для виконання третього завдання
void taskThree() {
    cout << "Ви обрали завдання 3" << endl;

    int n;
    cout << "Введіть n: ";
    cin >> n;

    double s = 0;
    for (int i = 1; i <= n; ++i) {
        s += (3 * i - 2) / factorial(100 * (i + 1) + 2);
    }

    double e = 1e-5;
    double g = 1e5;
    if (abs(s) < e) {
        cout << "Ряд збігається." << endl;
    } else if (abs(s) > g) {
        cout << "Ряд розбігається." << endl;
    } else {
        cout << "Неможливо визначити збіжність низки." << endl;
    }
}

// Основна функція програми
int main() {
    int choice;

    do {
        cout << "Виберіть завдання:" << endl;

```

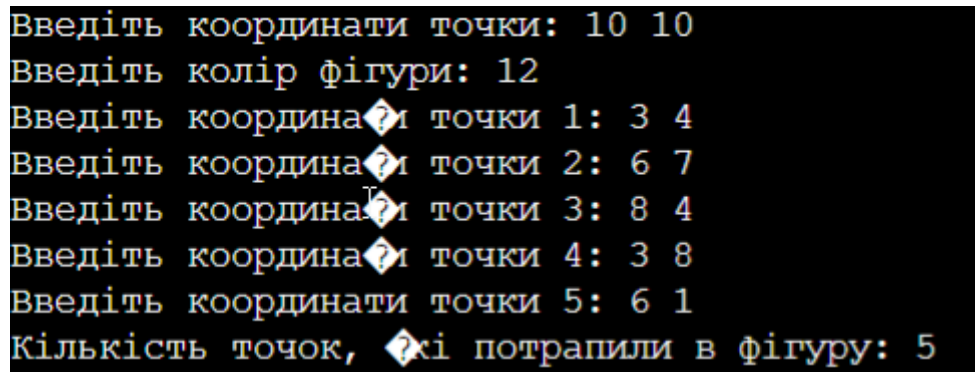
```
cout << "1. Завдання 1" << endl;
cout << "2. Завдання 2" << endl;
cout << "3. Завдання 3" << endl;
cout << "0. Вийти" << endl;
cout << "Ваш вибір: ";
cin >> choice;

switch (choice) {
case 1:
    taskOne();
    break;
case 2:
    taskTwo();
    break;
case 3:
    taskThree();
    break;
case 0:
    cout << "До свідання!" << endl;
    break;
default:
    cout << "Неправильний вибір. Спробуйте ще раз." << endl;
}
} while (choice != 0);

return 0;
}
```

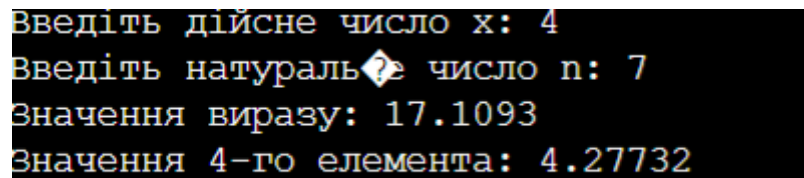
## ДОДАТОК Б

## Скрін-шоти вікна виконання програми



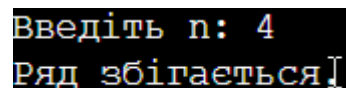
Введіть координати точки: 10 10  
Введіть колір фігури: 12  
Введіть координати точки 1: 3 4  
Введіть координати точки 2: 6 7  
Введіть координати точки 3: 8 4  
Введіть координати точки 4: 3 8  
Введіть координати точки 5: 6 1  
Кількість точок, які потрапили в фігуру: 5

Рисунок Б.1 – Екран виконання програми для вирішення завдання  
Геометричні області 12



Введіть дійсне число x: 4  
Введіть натуральне число n: 7  
Значення виразу: 17.1093  
Значення 4-го елемента: 4.27732

Рисунок Б.2 – Екран виконання програми для вирішення завдання  
Підрахунок рядів 20



Введіть n: 4  
Ряд збігається]

Рисунок Б.3 – Екран виконання програми для вирішення завдання  
Збіжність/розбіжність рядів 5