МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет
«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 С ++"

ХАІ.301. Електрична інженерія. 319а. 14 ЛР

Виконав студент гр. 319а

03.11.2025 Володимир Івахнін
(підпис, дата) (П.І.Б.)

Перевірив
03.11.2025 асистент Євгеній Пявка
(підпис, дата) (П.І.Б.)

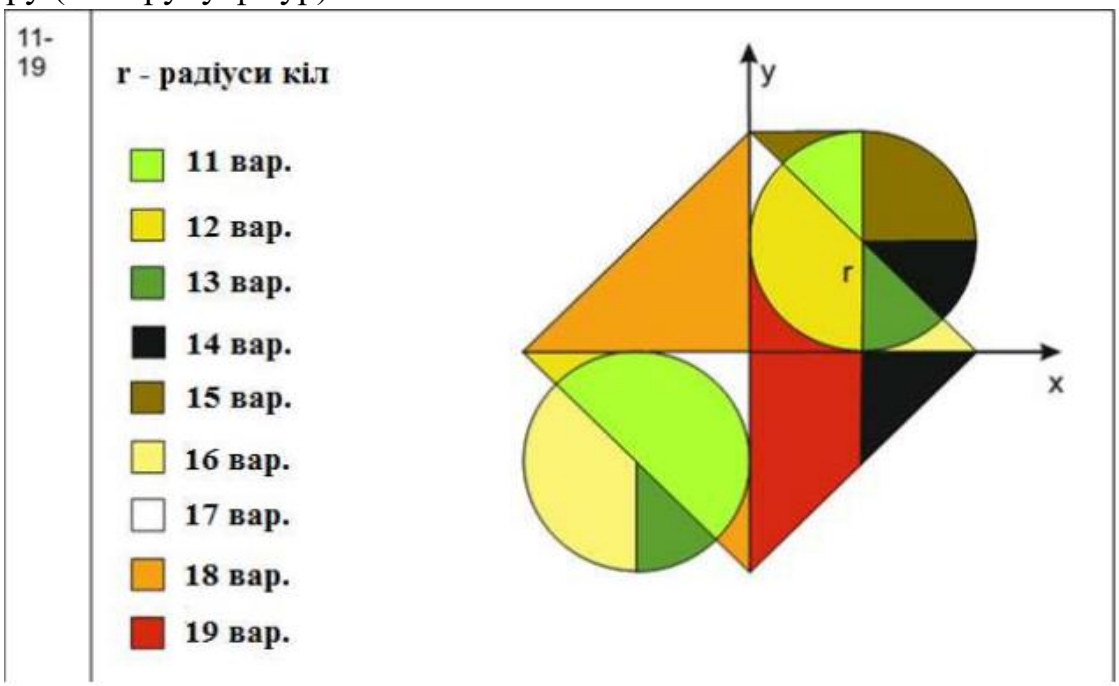
2025

МЕТА РОБОТИ

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

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Дано дійсні числа (xi, yi), i = 1,2, ... 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:
a) 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.

$$\sum_{i=1}^{n} \frac{x^{2i-1}}{(i+2)!}$$

Завдання 3. Дослідити ряд на збіжність. Умова закінчення циклу обчислення суми прийняти у вигляді: | un | < e або | un | > g, де e – мала величина для переривання циклу обчислення суми збіжного ряду (e = 10-5... 10-20); g – величина для переривання циклу обчислення суми розбіжного ряду (g = 102... 105).

$$\sum_{n=1}^{\infty} \frac{2^n (2n-1)!}{\sqrt{n}!}$$

Завдання 4. Організувати меню в командному вікні для багаторазового виконання завдань. *Додати функцію/функції, що вводять з консолі та повертають коректне значення цілого/дійсного типу у відповідності з обмеженнями вхідних даних кожного завдання.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі 1 (Варіант 13).

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

Дано дійсні числа (xi, yi), i = 1,2, ... n, – координати точок на площині.

Вихідні дані (ім’я, опис, тип): Визначити кількість точок, що потрапляють в фігуру заданого кольору (або групу фігур).

Лістинг коду вирішення задачі наведено в дод. А (стор. 6).
Екран роботи програми показаний на рис. Б.1

Завдання 2.

Вирішення задачі 2 (Варіант 14).

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

Дано дійсне число x і натуральне число n.

Вихідні дані (ім’я, опис, тип):

a) Обчислити значення виразу при заданих x і n для виразу
b) Вивести: для парних варіантів – значення кожного третього елемента, для непарних – значення кожного четвертого елемента.

Завдання3.
Вирішення задачі 3 (Варіант 13).

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

Дослідити ряд на збіжність.
Вихідні дані (ім’я, опис, тип):
Умова закінчення циклу обчислення суми прийняти у вигляді: | un | < e або | un | > g, де e – мала величина для переривання циклу обчислення суми збіжного ряду (e = 10-5... 10-20); g – величина для переривання циклу обчислення сумирозбіжного ряду (g = 102... 105).

ВИСНОВКИ

Ми вивчили синтаксис мови C++ для реалізації різних типів циклів: з передумовою, післяумовою та параметризованих. Також освоїли візуалізацію цих алгоритмів за допомогою UML-діаграм. Виконання практичних завдань у середовищі Visual Studio дозволило закріпити знання та набути практичних навичок у програмуванні циклів.

ДОДАТОК А

Лістинг коду програми

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

// Завдання 1 (Варіант 13): Визначити кількість точок всередині кола
void task1() {
    int n;
    cout << "Введіть кількість точок: ";
    cin >> n;

    int count = 0;
    double x, y;
    double r = 5; // Розмір радіусу кола, можна змінити

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

        // Перевірка, чи точка знаходиться всередині кола
        if (pow(x, 2) + pow(y, 2) <= pow(r, 2)) {
            count++;
        }
    }

    cout << "Кількість точок всередині кола: " << count << endl;
}

// Завдання 2 (Варіант 14): Обчислити суму ряду та вивести непарні елементи
void task2() {
    double x;
    int n, variant;

    cout << "Введіть значення x: ";
    cin >> x;
    cout << "Введіть кількість елементів n: ";
    cin >> n;
    cout << "Введіть варіант (парний або непарний): ";
    cin >> variant;

    double sum = 0;

    // Обчислення суми
    for (int i = 1; i <= n; ++i) {
        double term = pow(x, 2 * i - 1) / tgamma(i + 3); // (i+2)! = tgamma(i+3)
        sum += term;
```

```
        // Виведення кожного 3-го або 4-го елемента залежно від варіанту
        if ((variant % 2 == 0 && i % 3 == 0) || (variant % 2 != 0 && i % 4 ==
0)) {
            cout << "Елемент " << i << ": " << term << endl;
        }
    }

    cout << "Загальна сума: " << sum << endl;
}

// Завдання 3 (Варіант 13): Дослідження збіжності ряду
void task3() {
    double sum = 0.0, term;
    double epsilon = 1e-5;  // Мала величина для збіжного ряду
    double g = 1e5;          // Велика величина для розбіжного ряду
 int n = 1;
    double x;

    cout << "Введіть значення x: ";
    cin >> x;

    // Обчислення елементів ряду
    while (true) {
        // Обчислення елементу ряду
        term = (pow(2, 2 * n) * factorial(2 * n - 1)) / sqrt(factorial(n));

        // Перевірка умови збіжності чи розбіжності
        if (fabs(term) < epsilon) {
            break;  // Якщо елемент менший за epsilon, припиняємо для збіжного
ряду
        }
        if (fabs(term) > g) {
            cout << "Ряд розбіжний." << endl;
            break;  // Якщо елемент більший за g, припиняємо для розбіжного ряду
        }
     sum += term;
        n++;
    }
   cout << "Загальна сума: " << sum << endl;
}
// Функція для обчислення факторіалу
long long factorial(int n) {
    long long fact = 1;
    for (int i = 1; i <= n; ++i) {
        fact *= i;
    }
    return fact;
}
int main() {
```

```
    int menu;
    do {
        // Вибір номеру завдання
        cout << "Виберіть завдання (1-3) або -1 для виходу: ";
        cin >> menu;
        switch (menu) {
            case 1: task1(); break;
            case 2: task2(); break;
            case 3: task3(); break;
            case -1: cout << "Вихід з програми." << endl; break;
            default: cout << "Невірний вибір!" << endl; break;
        }
    } while (menu != -1);
  return 0;
}
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

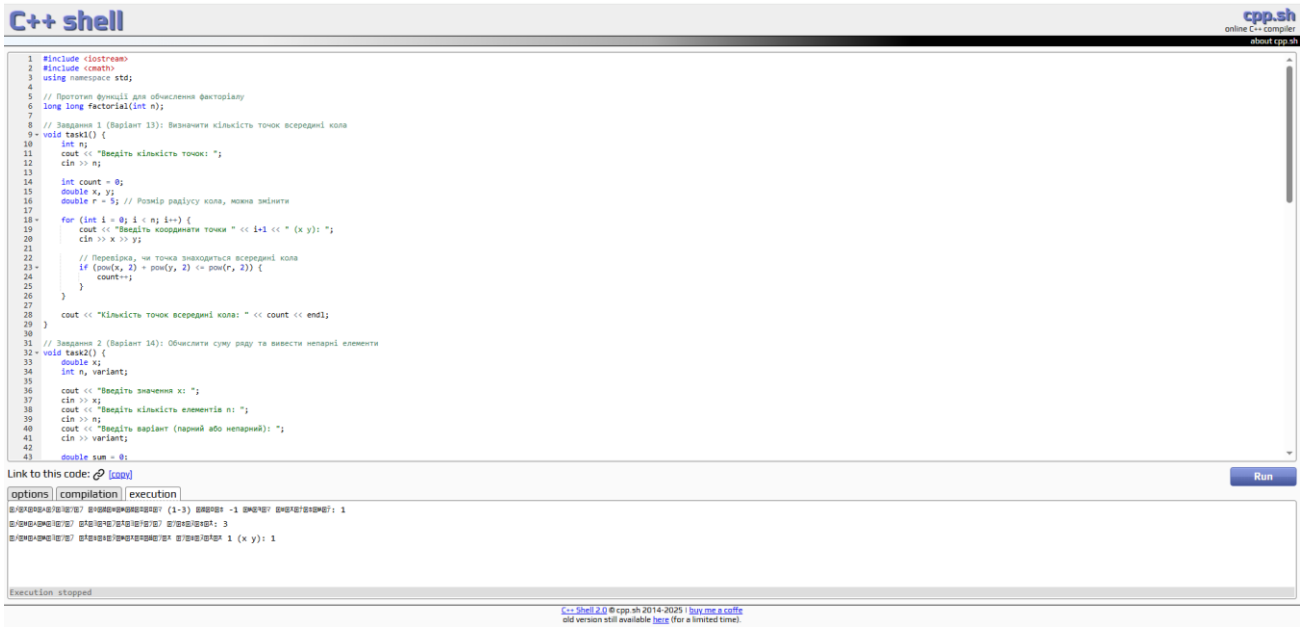


Рисунок Б.1 – Екран виконання програми для вирішення завдання 1 Варіант 13

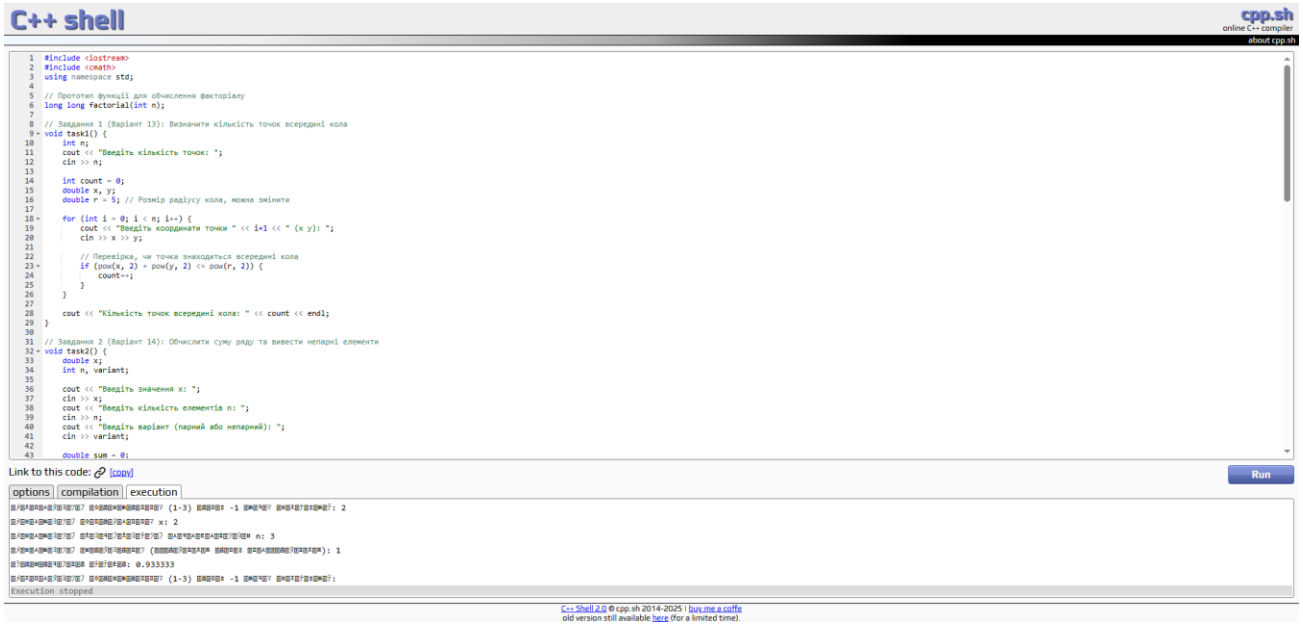


Рисунок Б.2 – Екран виконання програми для вирішення завдання 2 Варіант 14.

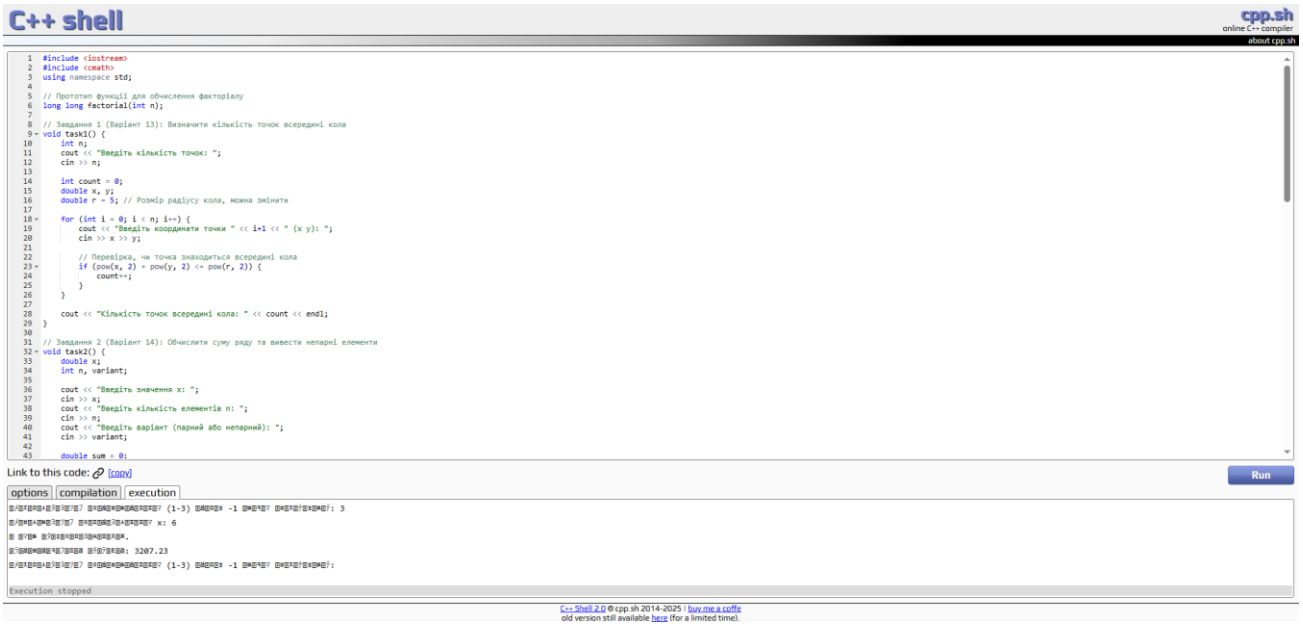
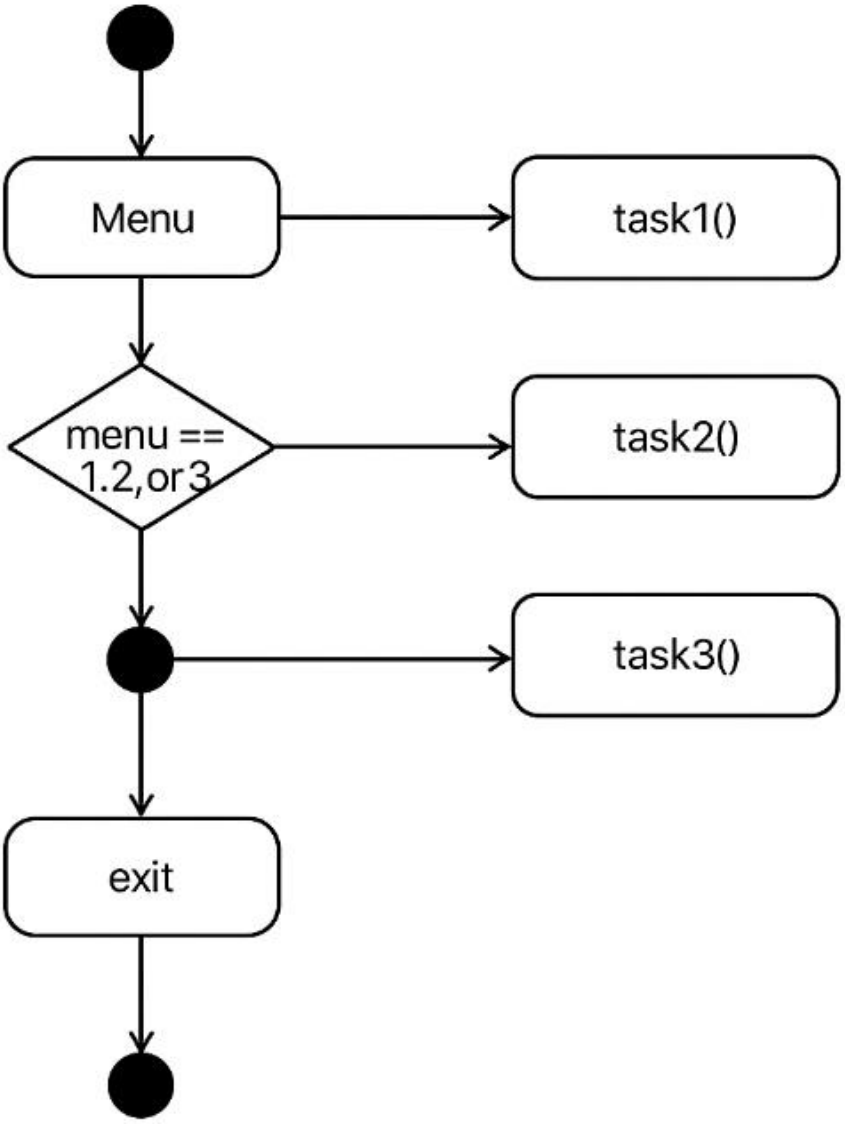


Рисунок Б.3 – Екран виконання програми для вирішення завдання 3 Варіант 13.


Menu
task1()
menu ==
1,2,or3
task2()
task3()
exit

ДОДАТОК В

Що таке параметризований цикл у C++?

Параметризований цикл — це цикл, кількість ітерацій якого визначається змінною (наприклад, в циклі `for`).

Що таке факторіал і як його обчислити в C++?

Факторіал числа — це добуток всіх цілих чисел від 1 до цього числа. Його можна обчислити за допомогою циклу або рекурсії.

Чим відрізняються цикли `while`, `do-while` і `for` у C++?

Цикл `while` перевіряє умову перед виконанням, цикл `do-while` — після, а цикл `for` використовується для ітерацій з лічильником.

Що таке умовне виконання в програмуванні і як воно працює в C++?

Умовне виконання дозволяє виконувати код тільки за певних умов, використовуючи оператори `if`, `else` та `switch`.

Що таке UML-діаграма і для чого вона використовується в програмуванні?

UML-діаграма — це візуальне представлення структури системи або процесу, яке допомагає зрозуміти зв'язки між частинами програми.