

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. 174. 319. 12 ЛР

Виконав студент гр. 319

Віталій НОВИКОВ

(підпис, дата)

(П.І.Б.)

Перевірів

к.т.н., доц. Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

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). Фігура представлена на рис.1.

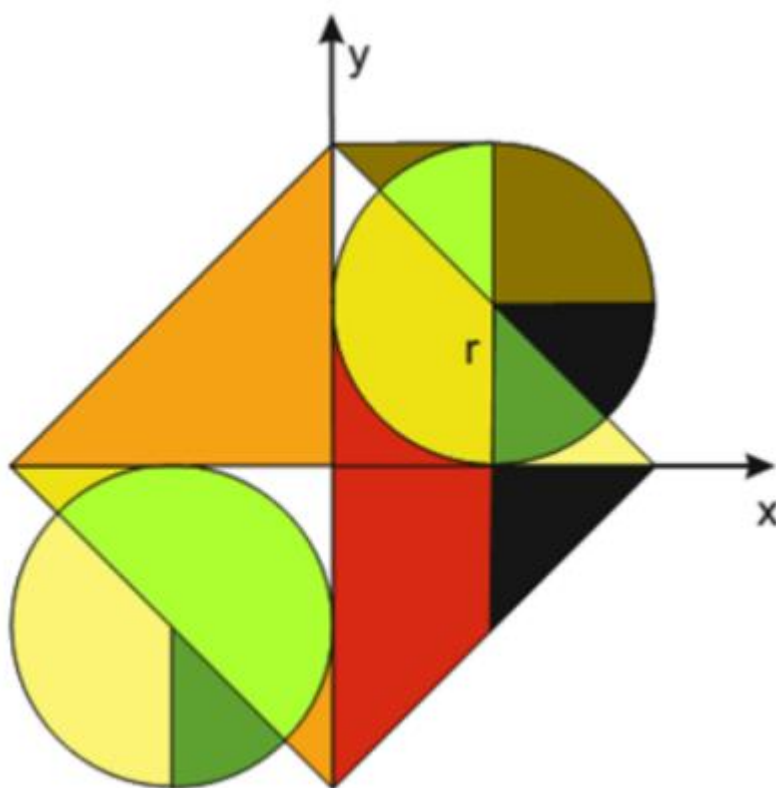


Рис.1 – Фігура для виконання завдання 1.

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

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

$$\sum_{k=1}^n \frac{x^{k-1}}{k!}$$

Рис.2 – Вираз для виконання 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 ($g = 10^2 \dots 10^5$). Рівняння представлено на рис.2.

$$\sum_{n=1}^{\infty} \frac{n^3 e^{2n+1}}{n!}$$

Рис.3 – Рівняння для виконання 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. Визначення кількості точок, які потрапляють у фігуру
Вирішення задачі: task1.

Вхідні дані:

- n : кількість точок (тип `int`, $n > 0$).
- $x[i]$, $y[i]$: координати i -тої точки (тип `double`, діапазон – дійсні числа).

Вихідні дані:

- `countInside`: кількість точок, які потрапили у фігуру (тип `int`).

Алгоритм вирішення представлений у графічному вигляді на рис.4.

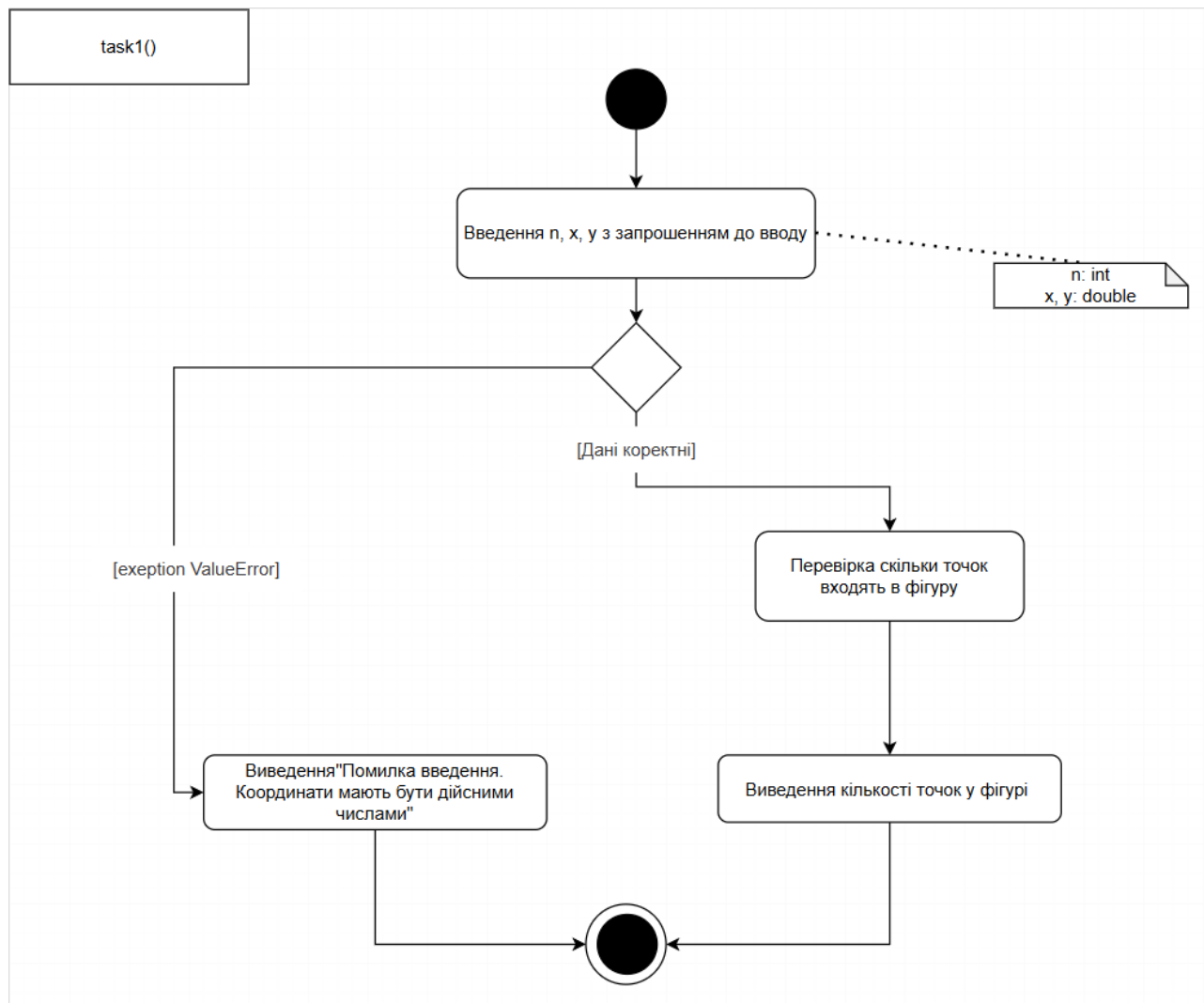


рис.4 – діаграма активності функції `task1()`.

Завдання 2. Обчислення суми ряду та виведення елементів

Вирішення задачі: `task2`.

Вхідні дані:

- `x`: дійсне число (тип `double`, $x > 0$).
- `n`: кількість членів ряду (тип `int`, $n > 0$).

Вихідні дані:

- `sum`: обчислена сума ряду (тип `double`).
- екстове повідомлення зі значеннями кожного четвертого елемента (тип `string`).

Алгоритм вирішення представлений у графічному вигляді на рис.5.

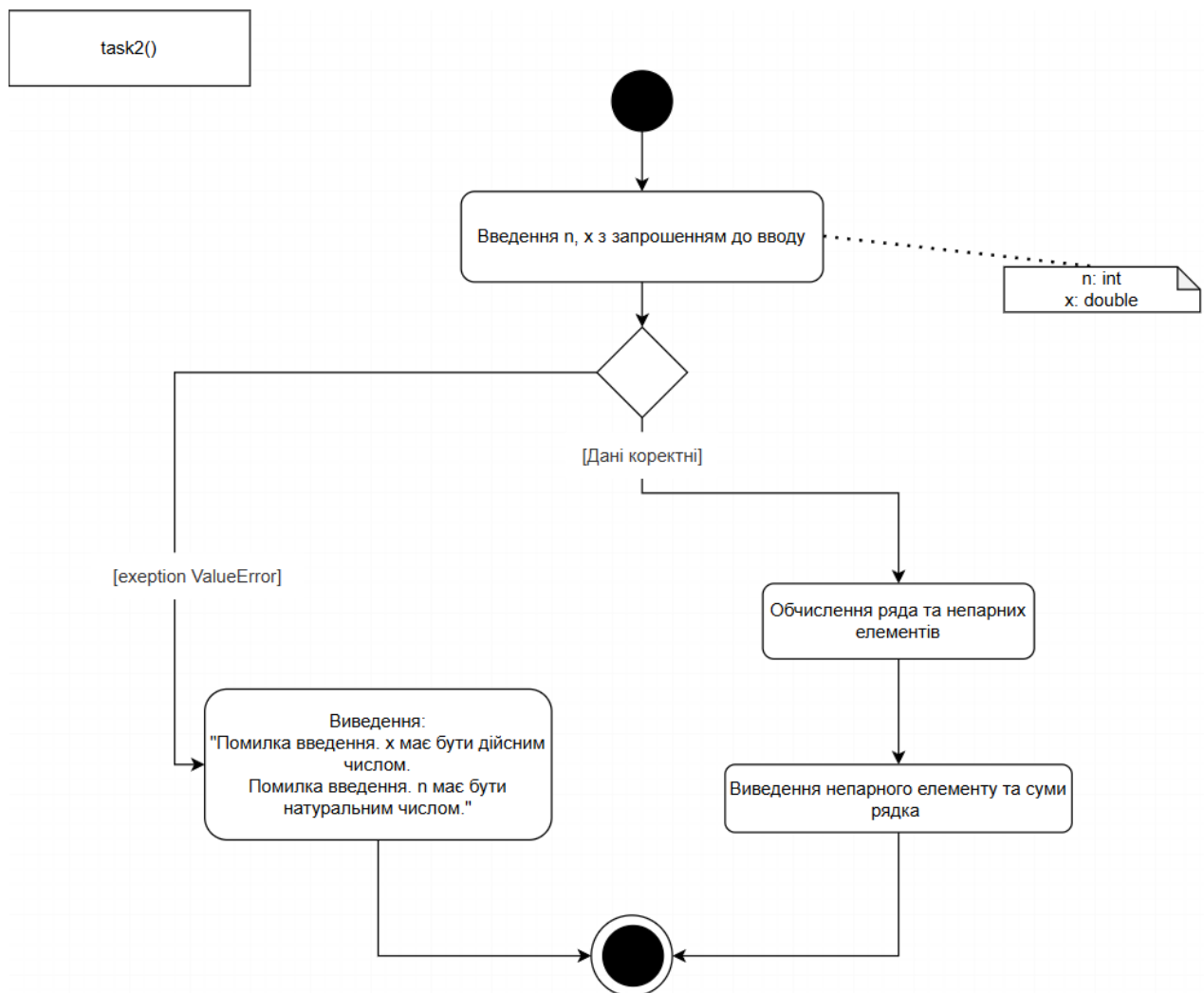


рис.5 – діаграма активності функції `task2 ()`.

Завдання 3. Дослідження збіжності ряду

Вирішення задачі: `task3`.

Вхідні дані:

- `epsilon`: мала величина для перевірки збіжності (тип `double`).
- `g`: величина для перевірки розбіжності (тип `double`).

Вихідні дані:

- `sum`: обчислена сума ряду (тип `double`).
- `lastElement`: значення останнього елемента ряду (тип `double`).

- Текстове повідомлення про тип ряду (збіжний чи розбіжний).
- Алгоритм вирішення представлений у графічному вигляді на рис.6.

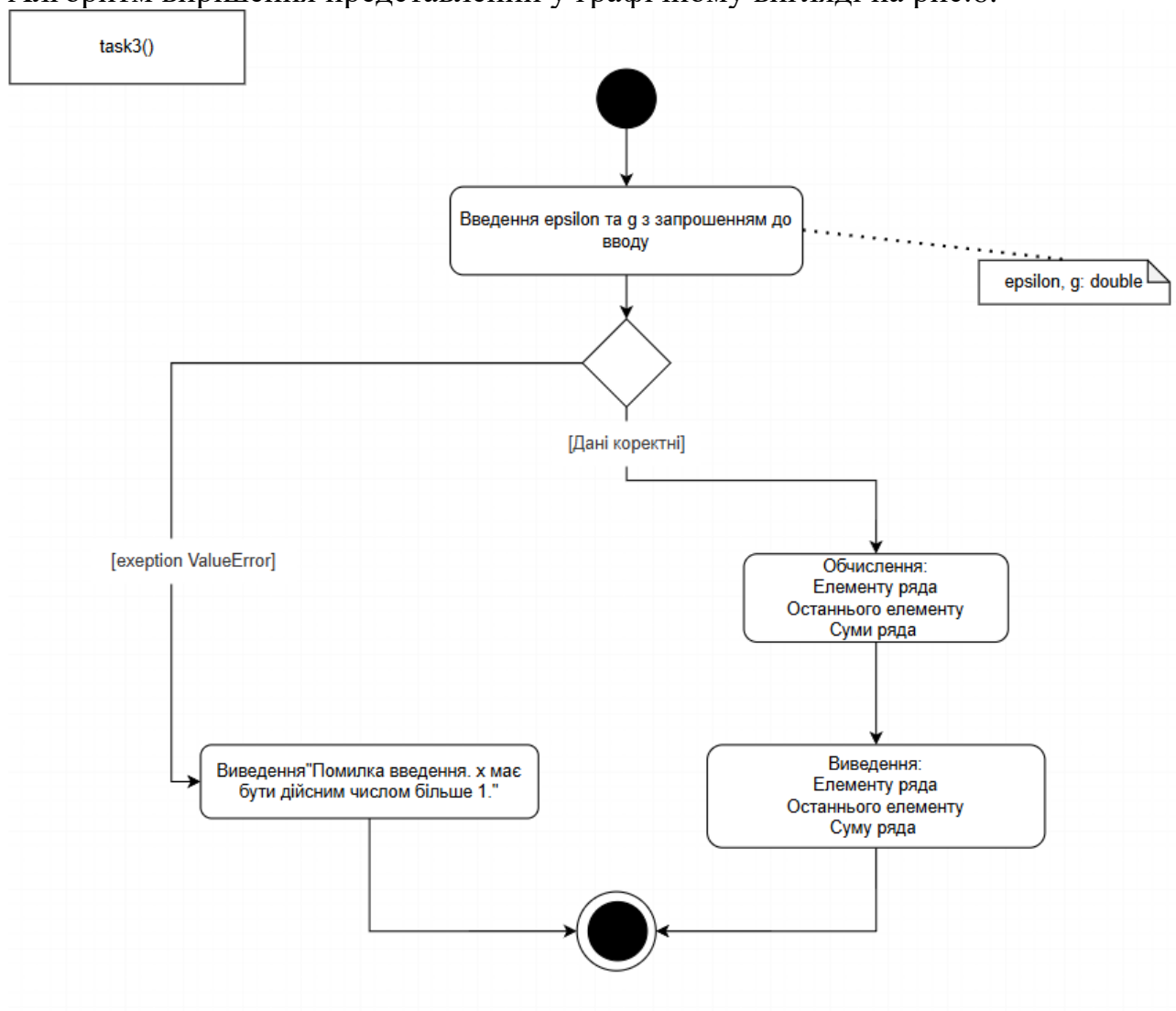


рис.6 – діаграма активності функції task2 ().

Завдання 4. Меню для вибору задачі

Опис задачі: Створити програму, яка реалізує меню для вибору однієї із задач, описаних у попередніх завданнях. Користувач може вибрати одну з доступних задач для виконання, або завершити роботу програми.

Вхідні дані:

- Вибір пункту меню: ціле число (тип `int`).

Вихідні дані:

- Текстові повідомлення про успішне виконання обраного завдання або помилки введення (тип `string`).

Алгоритм вирішення представлений у графічному вигляді на рис.7.

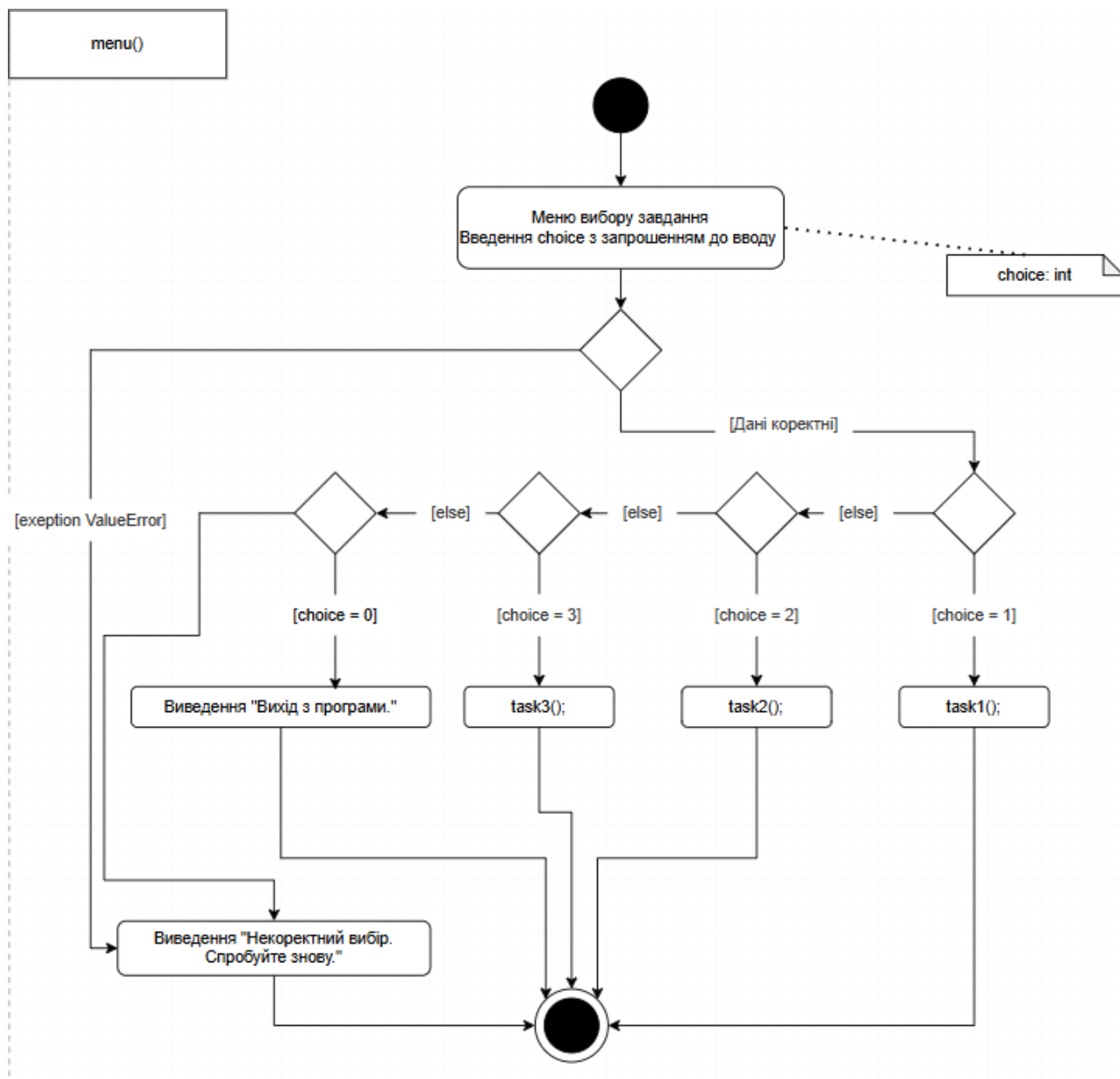


рис.7 – діаграма активності функції menu().

Лістинг коду вирішення задач наведено в додатку А (стор. 8).
Екран роботи програми показаний в додатку Б, рис. Б.1(стор.11).

ВИСНОВКИ

У ході виконання лабораторної роботи була розроблена програма на C++, яка складається з кількох задач. Загалом, робота поглибила знання з програмування, покращила навички алгоритмічного мислення та навчила ефективно взаємодіяти з користувачем.

ДОДАТОК А

Лістинг коду програми

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

// Оголошення функцій
void task1();
void task2();
void task3();
void menu();

bool isValidDouble(double &value);
bool isValidInt(int &value, bool positiveOnly = false);

int main() {
    menu();
    return 0;
}

// Реалізація меню
void menu() {
    while (true) {
        cout << "\nОберіть завдання:\n";
        cout << "1 - Завдання 1\n";
        cout << "2 - Завдання 2\n";
        cout << "3 - Завдання 3\n";
        cout << "4 - Вихід\n";
        cout << "Ваш вибір: ";

        int choice;
        if (!isValidInt(choice)) {
            cout << "Помилка введення. Спробуйте знову.\n";
            continue;
        }

        switch (choice) {
            case 1:
                task1();
                break;
            case 2:
                task2();
                break;
            case 3:
                task3();
                break;
            case 4:
                cout << "Вихід з програми.\n";
                return;
            default:
                cout << "Некоректний вибір. Спробуйте знову.\n";
        }
    }
}

// Завдання 1: Обчислити кількість точок у заданій області
void task1() {
    int n;
    cout << "\nВведіть кількість точок: ";
    if (!isValidInt(n, true)) {
        cout << "Кількість точок має бути натуральним числом.\n";
        return;
    }
}
```



```

int countInside = 0;
for (int i = 0; i < n; ++i) {
    double x, y;
    cout << "Введіть координати точки " << i + 1 << " (x y): ";
    if (!isValidDouble(x) || !isValidDouble(y)) {
        cout << "Помилка введення. Координати мають бути дійсними
числами.\n";
        return;
    }

    if ((x >= 0 && y >= 0 && x * x + y * y <= 1) || (x >= 0 && y <= 0 && y
>= -x)) {
        ++countInside;
    }
}

cout << "Кількість точок у фігурі: " << countInside << "\n";
}

// Завдання 2: Обчислити ряд та вивести непарні елементи
void task2() {
    int n;
    double x, sum, xs;

    cout << "\nВведіть x: ";
    if (!isValidDouble(x)) {
        cout << "Помилка введення. x має бути дійсним числом.\n";
        return;
    }

    cout << "Введіть n: ";
    if (!isValidInt(n, true)) {
        cout << "Помилка введення. n має бути натуральним числом.\n";
        return;
    }

    sum = x;
    xs = 1;
    for (int j = 1; j <= n; j++) {
        xs /= x;
        sum += xs;

        if (j % 2 != 0) {
            cout << "Непарний елемент " << j << ": " << xs << endl;
        }
    }

    cout << "Сума ряду: " << sum << endl;
}

// Завдання 3: Продемонструвати збіжність ряду
void task3() {
    double x, u, sum, e = 1E-20;

    cout << "\nx (x > 1): ";
    if (!isValidDouble(x) || x <= 1) {
        cout << "Помилка введення. x має бути дійсним числом більше 1.\n";
        return;
    }

    u = 1 / x;
    sum = u;
    while (fabs(u) >= e) {
        cout << "Елемент ряду: " << u << endl;
        x *= x;
    }
}

```

```
        u = 1 / x;
        sum += u;
    }

    cout << "Останній елемент: " << u << endl;
    cout << "Сума ряду: " << sum << endl;
}

// Перевірка введення дійсного числа
bool isValidDouble(double &value) {
    cin >> value;
    if (cin.fail()) {
        cin.clear();
        cin.ignore(10000, '\n');
        return false;
    }
    return true;
}

// Перевірка введення цілого числа
bool isValidInt(int &value, bool positiveOnly) {
    cin >> value;
    if (cin.fail() || (positiveOnly && value <= 0)) {
        cin.clear();
        cin.ignore(10000, '\n');
        return false;
    }
    return true;
}
```

ДОДАТОК Б

Скріншот вікна виконання всієї програми представлено на рисунках Б.1

```
Оберіть завдання:
1 - Завдання 1
2 - Завдання 2
3 - Завдання 3
4 - Вихід
Ваш вибір: 1

Введіть кількість точок: 1
Введіть координати точки 1 (x y): 1 1
Кількість точок у фігурі: 0

Оберіть завдання:
1 - Завдання 1
2 - Завдання 2
3 - Завдання 3
4 - Вихід
Ваш вибір: 2

Введіть x: 1
Введіть n: 1
Непарний елемент 1: 1
Сума ряду: 2

Оберіть завдання:
1 - Завдання 1
2 - Завдання 2
3 - Завдання 3
4 - Вихід
Ваш вибір: 3

x (x > 1): 2
Елемент ряду: 0.5
Елемент ряду: 0.25
Елемент ряду: 0.0625
Елемент ряду: 0.00390625
Елемент ряду: 1.52588e-05
Елемент ряду: 2.32831e-10
Елемент ряду: 5.42101e-20
Останній елемент: 2.93874e-39
Сума ряду: 0.816422
```

Рисунок Б.1 – Екран виконання всієї програми