

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.

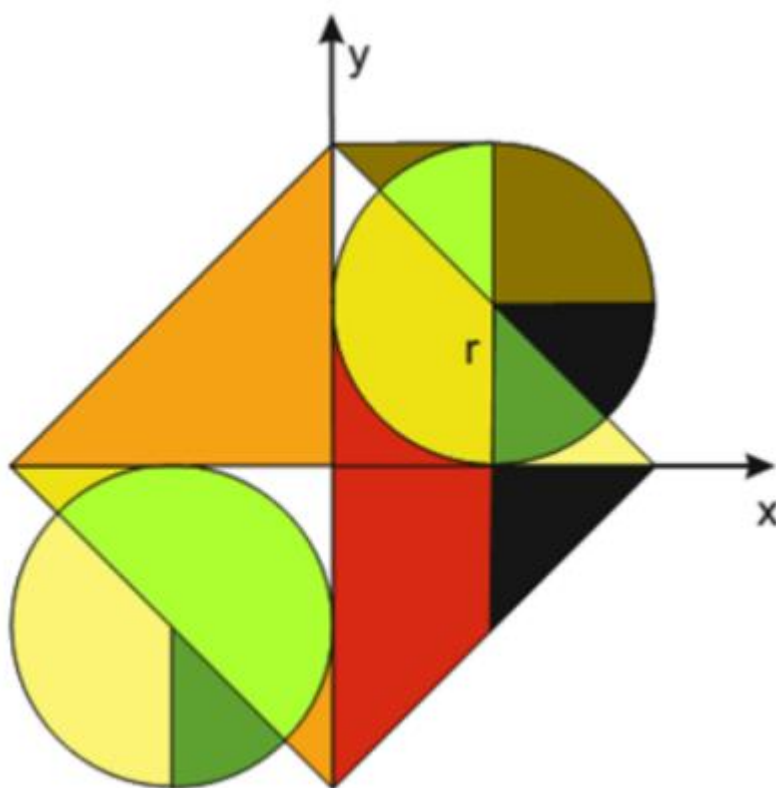


Рис.1 – Фігура для виконання завдання 1.

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  $\epsilon$  – мала величина для переривання циклу обчислення суми збіжного ряду ( $\epsilon = 10^{-5} \dots 10^{-20}$ );  $g$  – величина для переривання циклу обчислення суми розбіжного ряду ( $g = 10^2 \dots 10^5$ ). 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.

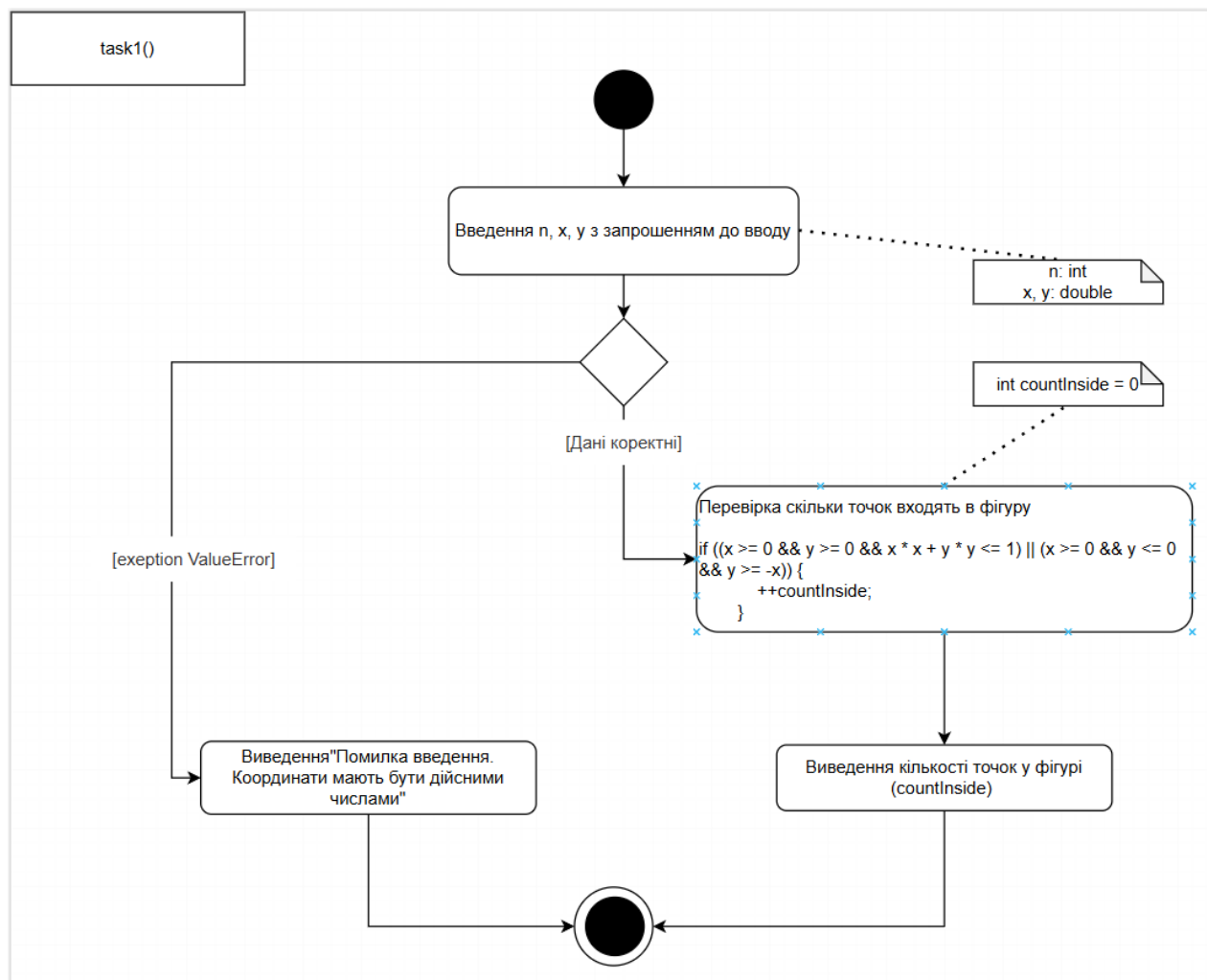


рис.4 – діаграма активності функції task1().

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.

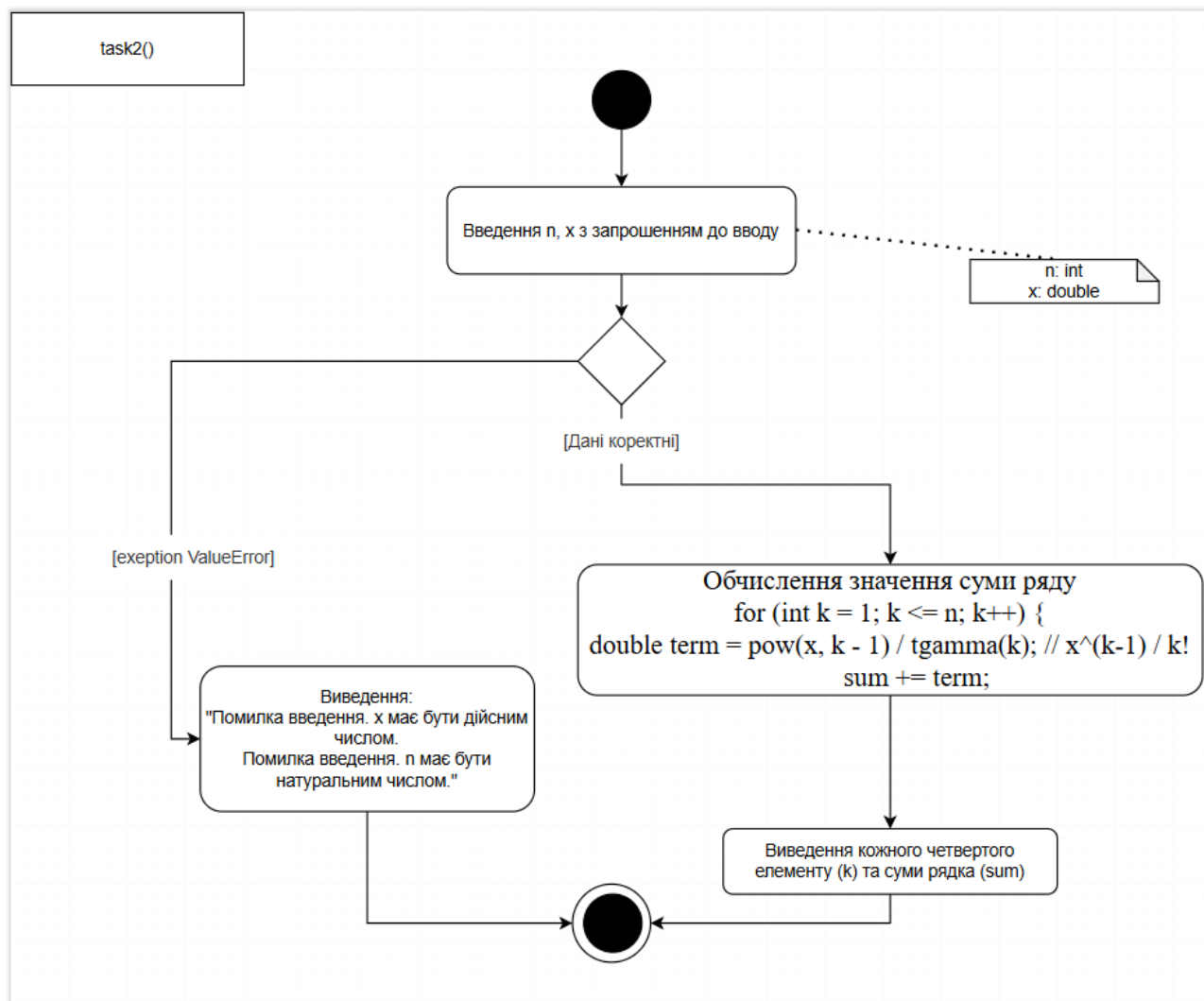


рис.5 – діаграма активності функції `task2 ()`.

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.

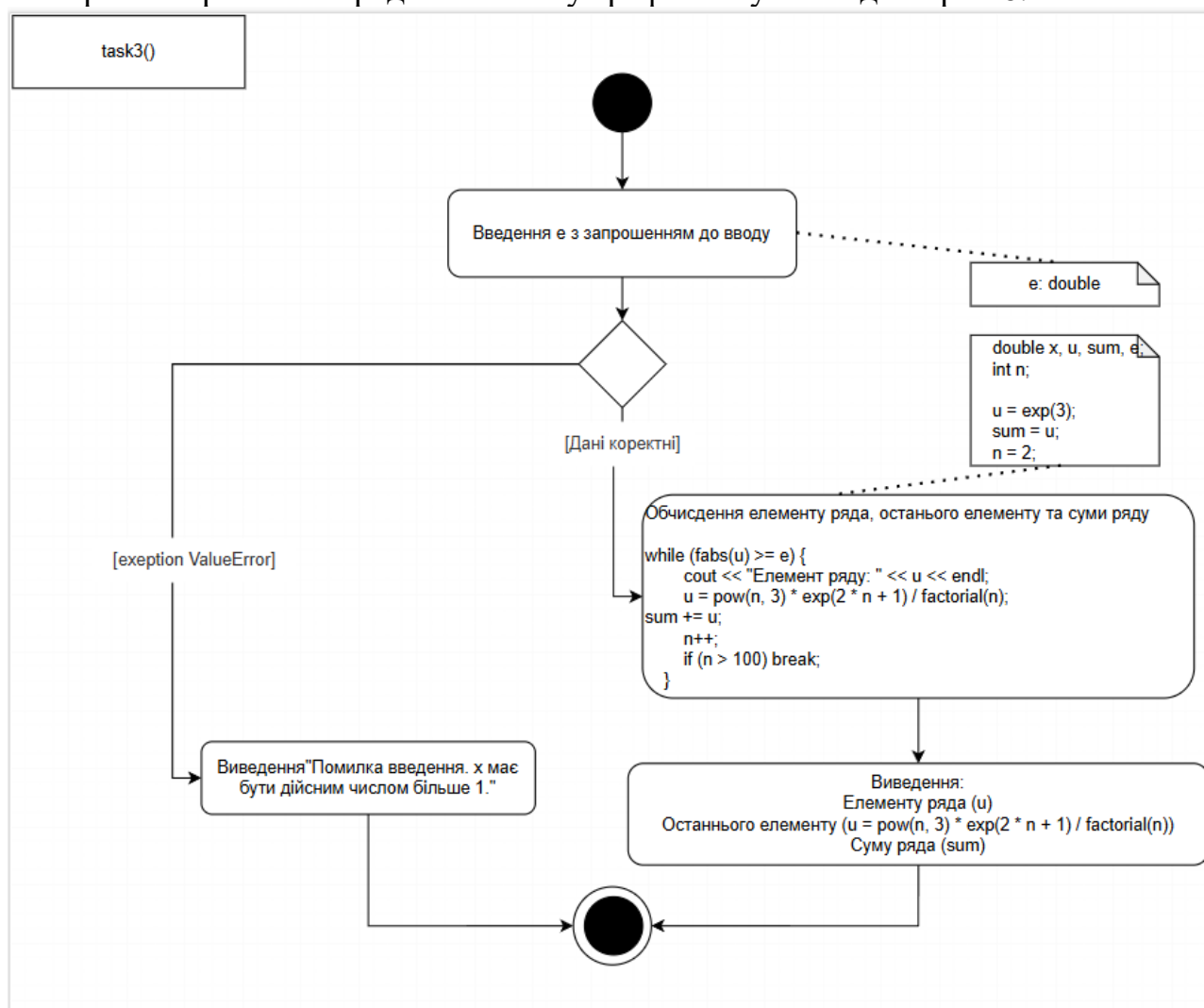


рис.6 – діаграма активності функції task2 ().

#### 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.

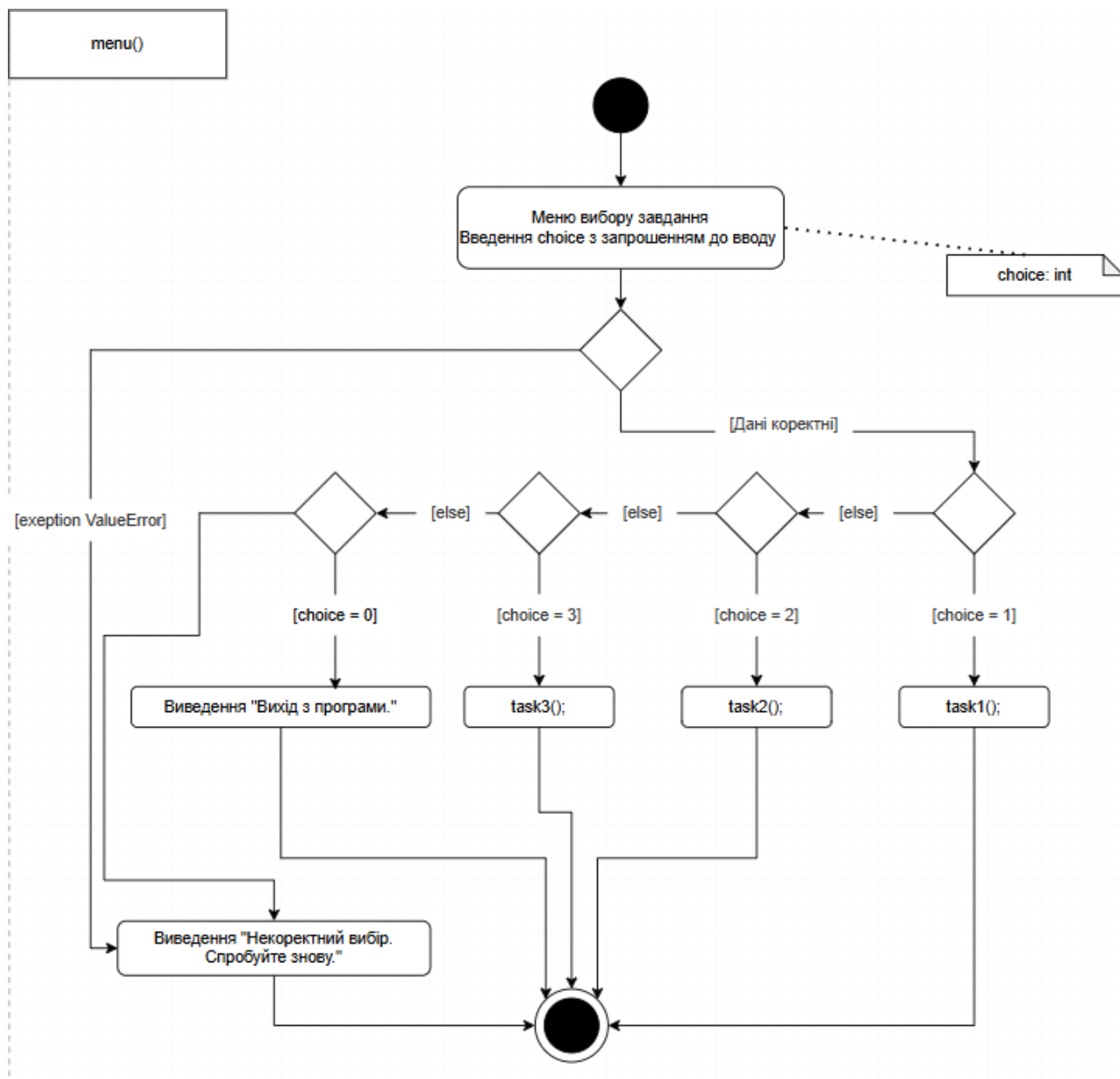


рис.7 – діаграма активності функції menu().

Лістинг коду вирішення задач наведено в додатку А (стор. 8).

Екран роботи програми показаний в додатку Б, рис. Б.1(стор.11),  
рис. Б.2(стор.12).

## 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;

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

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

    double sum = 0.0;

    // Обчислення значення суми ряду
    for (int k = 1; k <= n; k++) {
        double term = pow(x, k - 1) / tgamma(k); // x^(k-1) / k!
        sum += term;

        // Виведення кожного четвертого елемента
        if (k % 4 == 0) {
            cout << "Кожен четвертий елемент (" << k << "): " << term << endl;
        }
    }

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

// Функція обчислення факторіала
double factorial(int x) {
    if (x <= 1) return 1;
    double res = 1;
    for (int i = 2; i <= x; i++)
        res *= i;
    return res;
}

// Завдання 3: Продемонструвати збіжність ряду

```

```

void task3() {
    double x, u, sum, e;
    int n;

    u = exp(3);
    sum = u;
    n = 2;
    e = 1E-05;

    cout << "\nВведіть e (малу величину, наприклад, 1E-5): ";
    if (!isValidDouble(e)) {
        cout << "Помилка введення. n має бути дійсним числом.\n";
        return;
    }

    while (fabs(u) >= e) {
        cout << "Елемент ряду: " << u << endl;
        u = pow(n, 3) * exp(2 * n + 1) / factorial(n);
        sum += u;
        n++;
        if (n > 100) break;
    }

    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 першої та другої задачі представлено на рисунках Б.1.

Скріншот вікна виконання третьої задачі представлено на рисунках Б.2.

```
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 3
Кількість точок у фігурі: 0

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

Введіть x (дійсне число): 1
Введіть n (натуральне число): 2
Сума ряду: 2
```

Рисунок Б.1 – Екран виконання першої та другої задачі

```

Ваш вибір: 3
Введіть е (малу величину, наприклад, 1E-5): 1E-5
Елемент ряду: 20.0855
Елемент ряду: 593.653
Елемент ряду: 4934.85
Елемент ряду: 21608.2
Елемент ряду: 62368.9
Елемент ряду: 132724
Елемент ряду: 222475
Елемент ряду: 306730
Елемент ряду: 358558
Елемент ряду: 363430
Елемент ряду: 324934
Елемент ряду: 259758
Елемент ряду: 187716
Елемент ряду: 123742
Елемент ряду: 74972.7
Елемент ряду: 42020.2
Елемент ряду: 21907.1
Елемент ряду: 10675.1
Елемент ряду: 4882.6
Елемент ряду: 2103.97
Елемент ряду: 856.992
Елемент ряду: 330.943
Елемент ряду: 121.487
Елемент ряду: 42.4969
Елемент ряду: 14.1969
Елемент ряду: 4.53846
Елемент ряду: 1.39093
Елемент ряду: 0.409373
Елемент ряду: 0.115886
Елемент ряду: 0.0315985
Елемент ряду: 0.00831028
Елемент ряду: 0.00211067
Елемент ряду: 0.000518306
Елемент ряду: 0.000123195
Елемент ряду: 2.83713e-05
Останній елемент: 6.33679e-06
Сума ряду: 2.52753e+06

```

Рисунок Б.2 – Екран виконання третьої задачі