

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. 175. 318. 02 ЛР

Виконав студент гр. _____ 318

_____ Сергій БОНДАРЕНКО

(підпис, дата)

(П.І.Б.)

Перевірив

_____ 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).

Geom 10. Дано дійсні числа (x_i, y_i) , $i = 1, 2, \dots, n$, – координати точок на площині. Визначити кількість точок, що потрапляють в фігуру коричневого кольору.

На рис. 1 показана фігура із завдання 1.

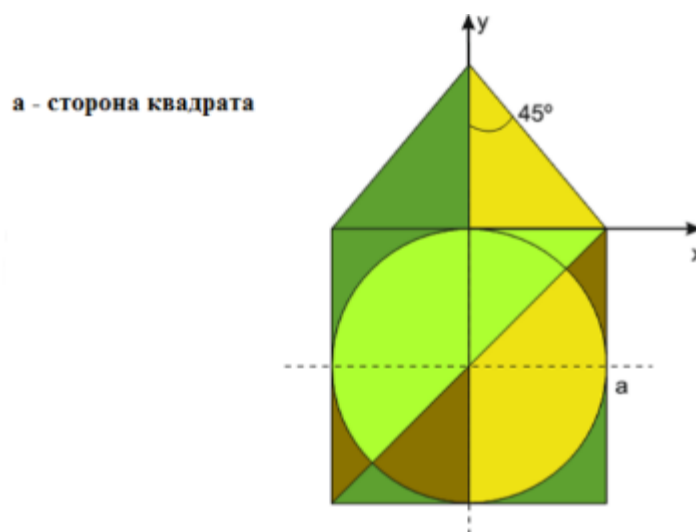


Рисунок 1 – Фігура із завдання 1

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

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

Таб. 2, №2.

На рис. 2 показаний вираз для завдання 2.

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

Рисунок 2 – Вираз для 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 ($g = 10^2 \dots 10^5$).

Таб. 3, №12.

На рис. 3 показаний вираз для завдання 3.

$$\sum_{n=1}^{\infty} \frac{n! * e^n}{n^{\sqrt{n}}}$$

Рисунок 3 – Вираз для 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.

Вирішення задачі Geom 10.

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

a – сторона квадрата, double, > 0 , числове значення.

n – кількість точок, int, > 0 , ціле число.

x – координата точки за віссю X, double, числове значення.

y – координата точки за віссю Y, double, числове значення.

Вихідні дані (ім'я, опис, тип):

countOfNumbersInBrownArea – кількість точок, що належать до коричневої області, int.

На рис. 3 показаний приклад діаграми для завдання Geom 10.

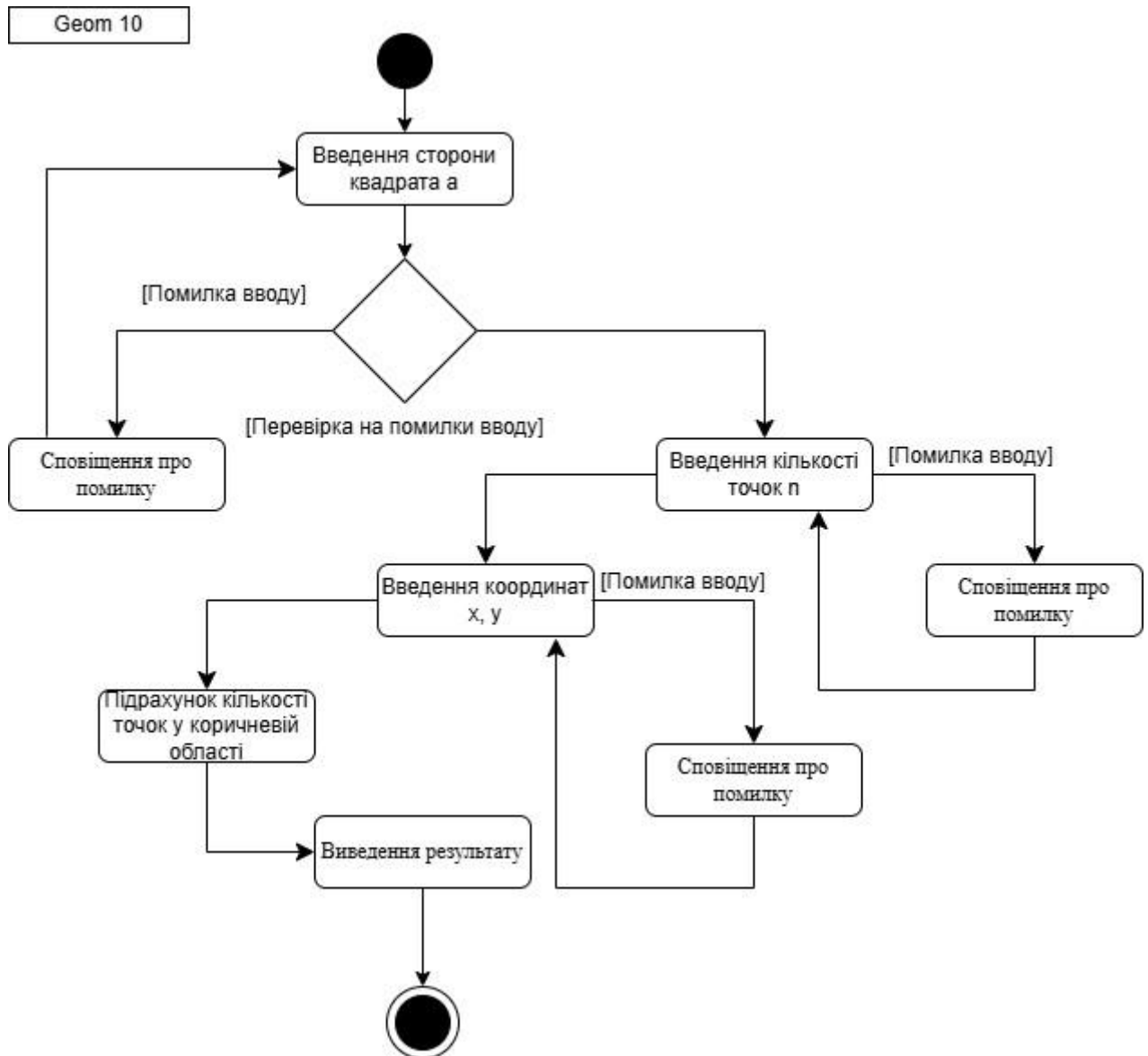


Рисунок 3 – Діаграма для завдання Geom 10

Лістинг коду вирішення завдання 1 наведено в дод. А (стор. 11-15) Екран роботи програми показаний на рис. Б.1 (додат. Б, стор. 16)

Завдання 2.

Вирішення задачі 2 із таблиці 2.

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

x – аргумент виразу, double, числове значення.

n – кількість елементів для обчислення добутку, int, > 0 , ціле число.

Вихідні дані (ім'я, опис, тип, обмеження):

result – результат обчислення добутку функції або повідомлення про помилку, double.

На рис. 4 показаний приклад діаграми для завдання 2.

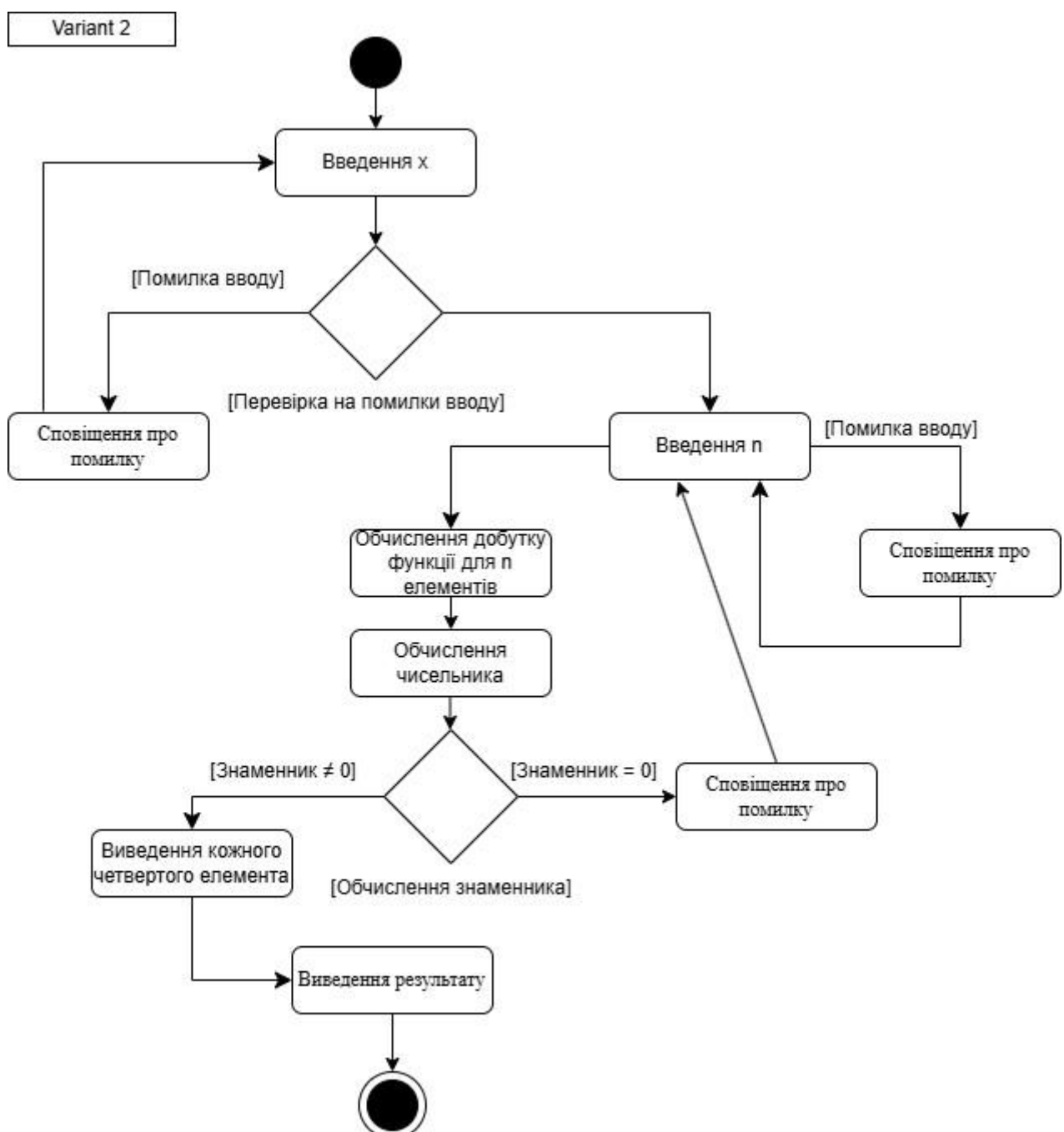


Рисунок 4 – Діаграма для завдання 2

Лістинг коду вирішення завдання 2 наведено в дод. А (стор. 11-15). Екран роботи програми показаний на рис. Б.2 (додат. Б, стор. 16)

Завдання 3.

Вирішення завдання 12 із таблиці 3.

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

eFromUser – нижня умова закінчення циклу (точність перевірки збіжності), double.

gFromUser – верхня умова закінчення циклу (грубість перевірки збіжності), double.

Вихідні дані (ім'я, опис, тип, обмеження):

result – логічний результат перевірки збіжності ряду, bool.

sum – сума елементів ряду, розрахована до зупинки обчислення, double.

На рисунку 5 показаний приклад діаграми для завдання 3

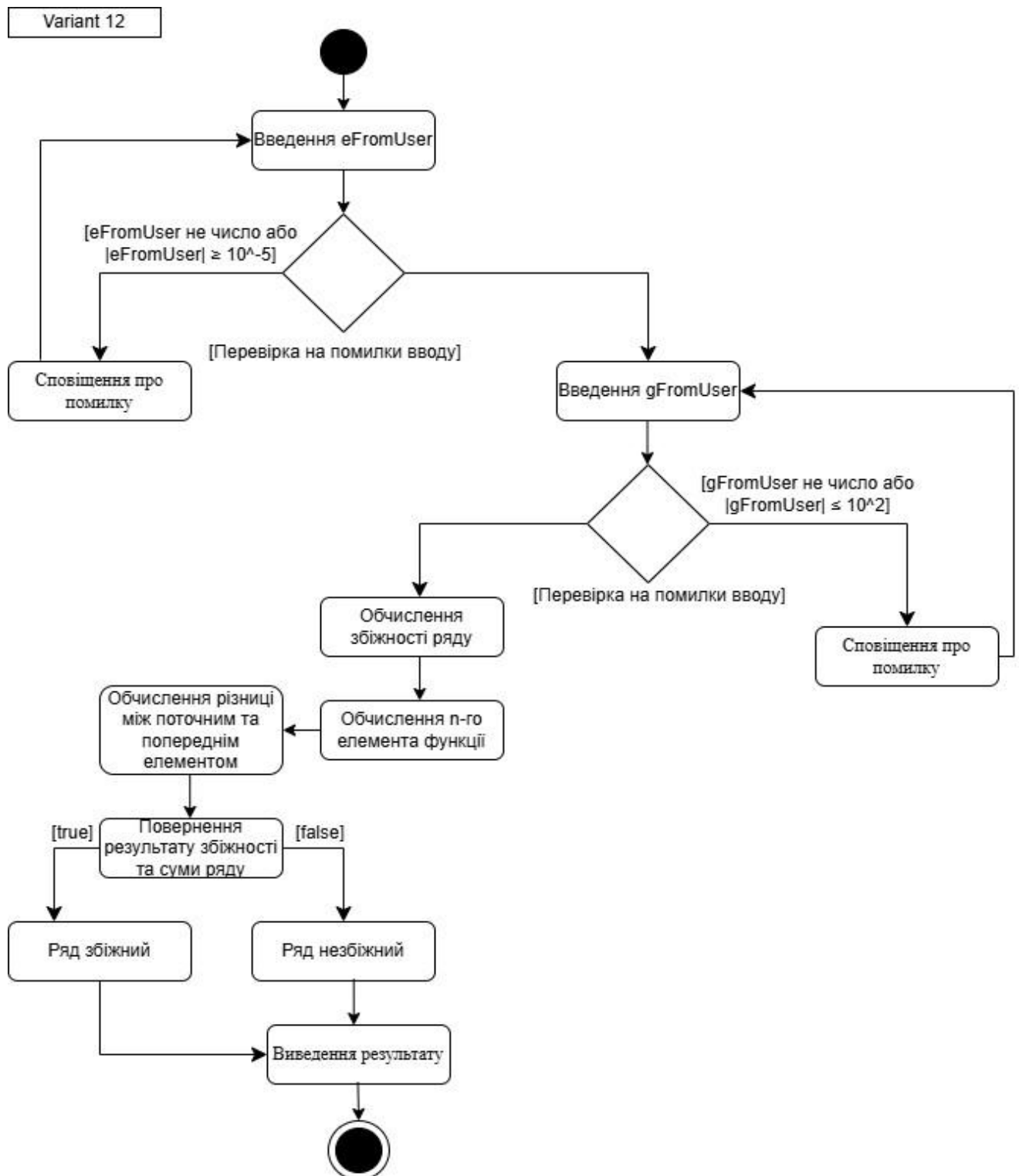


Рисунок 5 – Діаграма для завдання 3

Лістинг коду вирішення задачі 12 із таблиці 3 наведено в дод. А (стор. 11-15). Екран роботи програми показаний на рис. Б.3 (додат. Б, стор. 17)

Завдання 4.

Організація меню.

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

choice – вибір задачі для виконання, int.

Вихідні дані (ім'я, опис, тип):

Повідомлення – текстове повідомлення, що інформує про виконання програми, string.

На рис. 6 показаний приклад діаграми для завдання 4

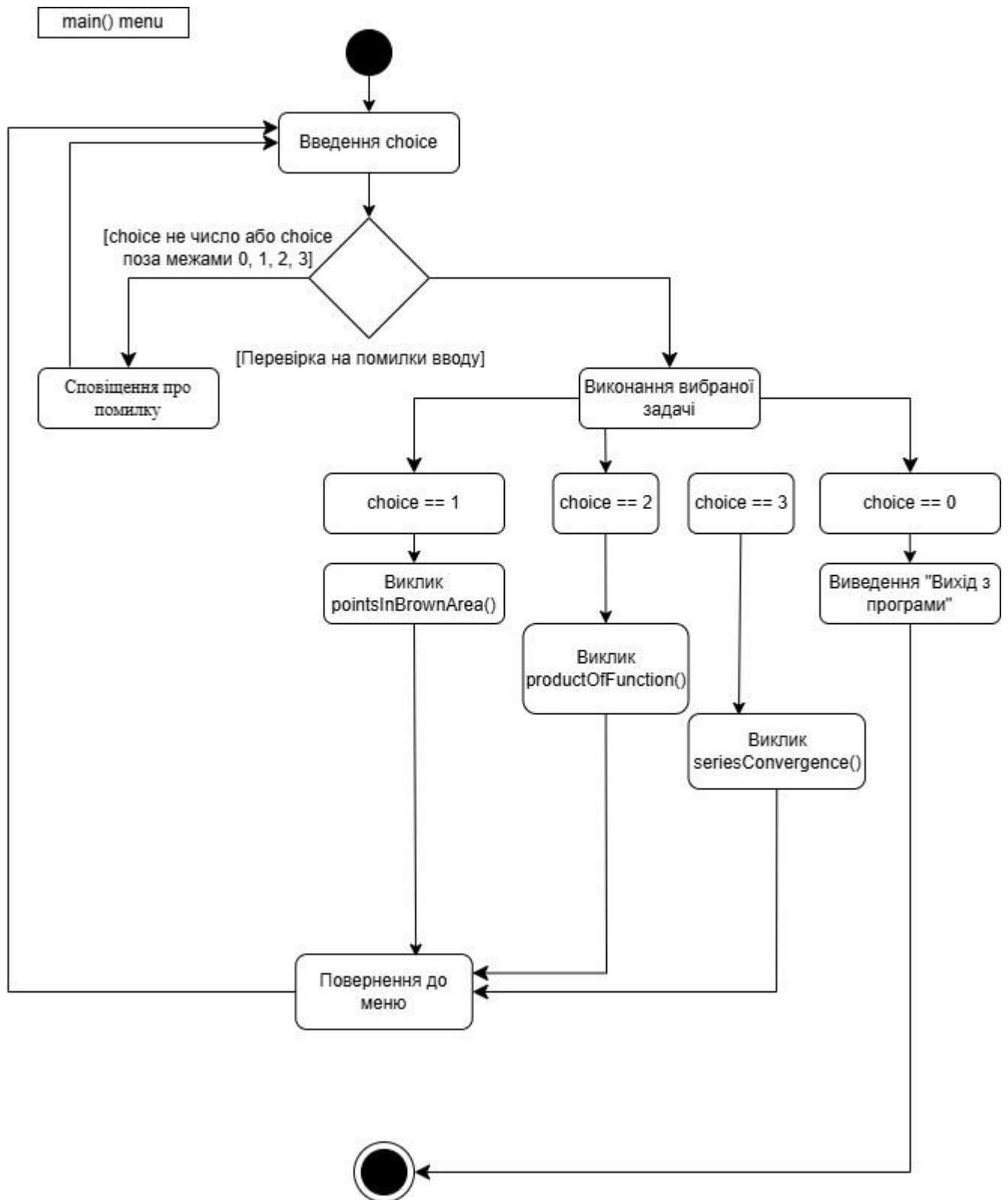


Рисунок 6 – Діаграма для завдання 4

Лістинг коду вирішення завдання 4 наведено в дод. А (стор. 11-15).

ВИСНОВКИ

Було вивчено теоретичний матеріал із синтаксису мови C++ та опрацьовано подання у вигляді UML діаграм алгоритмів. Було реалізовано алгоритми з використанням інструкцій циклу з передумовою, циклу з післяумовою і параметризованих операторів циклу мовою C++ у середовищі Visual Studio.

ДОДАТОК А

Лістинг коду програми

```

#include <iostream>
#include <cmath>
#include <stdexcept>
#include <vector>
#include <algorithm>
#include <string>
#include <utility>
using namespace std;

double inputSideOfSquare() {
    Step1:
        cout << "Введіть сторону квадрата a: ";
        double a;
        cin >> a;

        if (cin.fail()) { // перевірка на відсутність помилок при отриманні введеного
            від користувача числа
                cout << "\n\"a\" повинно бути числом\n";
                cin.clear();
                cin.ignore();
                goto Step1;
            }

        if (a <= 0) {
            cout << "\n\"a\" має бути більше нуля\n";
            cin.clear();
            cin.ignore();
            goto Step1;
        }

        return a;
    }

double inputCoordinate(char nameOfAxis, int numberOfDot) {
    Step1:
        cout << "Введіть " << numberOfDot << "координату " << nameOfAxis << ": ";
        double coordinate;
        cin >> coordinate;

        if (cin.fail()) { // перевірка на відсутність помилок при отриманні введеного
            від користувача числа
                cout << "\n\"\" << nameOfAxis << "\" повинно бути числом\n";
                cin.clear();
                cin.ignore();
                goto Step1;
            }

        return coordinate;
    }

// Функція перевірки точки в коричневій області
bool isPointInBrownArea(double x, double y, double a) {
    double r = a / 2.0; // Радіус кола

    // Перевірка для for all sectors
    bool inFirstBrownSector = (x * x + y * y <= r * r && x <= 0 && y <= x); //
    перевірка на наявність точки в секторі кола коричневої області
    bool inSecondBrownSector = (x * x + y * y >= r * r && x >= -r && y >= -r && y
    <= 0 && y >= x); // перевірка на наявність точки у маленькому шматочку коричневої
    області зліва внизу

```

```

    bool inThirdBrownSector = (x * x + y * y >= r * r && x <= r && y <= r && y >=
0 && y <= x); // перевірка на наявність точки у маленькому шматочку коричневої
області праворуч угорі
    return inFirstBrownSector || inSecondBrownSector || inThirdBrownSector; //
Повертає true, якщо точка буде в одному з коричневих секторів
}

int inputN(string textToConsole) {
Step1:
    cout << textToConsole;
    int n;
    cin >> n;

    if (cin.fail()) { // перевірка на відсутність помилок при отриманні введеного
від користувача числа
        cout << "\n\n" повинно бути числом\n";
        cin.clear();
        cin.ignore();
        goto Step1;
    }

    if (n <= 0) {
        cout << "\n\n" має бути більше нуля\n";
        cin.clear();
        cin.ignore();
        goto Step1;
    }

    return n;
}

int pointsInBrownAreaCheck() {
    double a = inputSideOfSquare();
    int n = inputN("Введіть кількість координат n: ");
    vector<bool> resultsArray(n);

    for (int i = 0; i < n; i++) { // виконання коду для n точок
        double x, y;
        x = inputCoordinate('x', i + 1); // запит у користувача x
        y = inputCoordinate('y', i + 1); // запит у користувача y

        resultsArray[i] = isPointInBrownArea(x, y, a); // перевірка приналежності
нової точки до коричневої області
    }

    int countOfNumbersInBrownArea = count(resultsArray.begin(),
resultsArray.end(), true); // підрахунок всіх true значень

    return countOfNumbersInBrownArea;
}

void pointsInBrownArea() {
    int countOfNumbersInBrownArea = pointsInBrownAreaCheck();

    cout << "Кількість точок в коричневій зоні: " << countOfNumbersInBrownArea;
}

// Variant 2: Обчислення виразу
double calculateProductOfFunction(double x, int n) {
    double numerator = 1.0;
    double denominator = 1.0;

    for (int i = 1; i <= n; i++) {
        numerator *= (x - pow(2, i)); // добуток наступного значення чисельника на
всі попередні
    }
}

```

```

        denominator *= (x - (pow(2, i) - 1)); // добуток наступного значення
знаменника на всі попередні
        if ((i + 1) % 4 == 0) { // виведення в консоль кожного четвертого елемента
            if (denominator != 0) {
                cout << "Елемент номер"<< i + 1 << ": " << numerator / denominator
<< "\n";
            }
            else { // у випадку, якщо знаменник дорівнює нулю - викидається
помилка, що обриває цикл
                throw invalid_argument("Знаменник дорівнює 0, обчислення
неможливе.");
            }
        }
    }

    if (denominator != 0) { // повернення значення виразу, якщо знаменник не
дорівнює нулю
        return numerator / denominator;
    } else { // у випадку, якщо знаменник дорівнює нулю - викидається помилка, що
обриває цикл
        throw invalid_argument("Знаменник дорівнює 0, обчислення неможливе.");
    }
}

double inputX() {
Step1:
    cout << "Введіть значення x: ";
    double x;
    cin >> x;

    if (cin.fail()) { // перевірка на відсутність помилок при отриманні введеного
від користувача числа
        cout << "\n\"x\" повинно бути числом\n";
        cin.clear();
        cin.ignore();
        goto Step1;
    }

    return x;
}

void productOfFunction() {
    double x = inputX();
    int n = inputN("Введіть значення n (ціле число): ");

    try { // виклик функції з уловлювання можливої помилки
        double result = calculateProductOfFunction(x, n);
        cout << "Результат обчислення: " << result << "\n";
    } catch (const exception &e) {
        cout << "Помилка: " << e.what() << "\n";
    }
}

// Variant 12: Обчислення часткової суми ряду
unsigned long long factorial(int n) {
    unsigned long long result = 1;
    for (int i = 1; i <= n; i++) {
        result *= i;
    }
    return result;
}

double functionInSeries(int n) {
    const double E = exp(1.0);
    double result = (factorial(n) * pow(E, n)) / pow(n, pow(n, 0.5));

    return result;
}

```

```

}

pair<bool, double> isSeriesConvergent(double e, double g) {
    double sum = 0.0;
    double difference, previousResult, currentResult;
    previousResult = 0;

    int n = 1;
    do {
        currentResult = functionInSeries(n);
        difference = currentResult - previousResult;
        previousResult = currentResult;
        sum += currentResult;
    } while (difference >= g || difference <= e);

    if (difference <= e) return {true, sum};
    return {false, sum};
}

void seriesConvergencePrintToConsole(bool result, double sum) {
    if (result) cout << "\nРяд збіжний\n";
    else cout << "\nРяд незбіжний\n";

    cout << "Сума чисел ряду: " << sum << "\n";
}

void seriesConvergence() {
    double e, eFromUser;
    int g, gFromUser;

    e = pow(10, -5);
    g = pow(10, 2);
    step1:

    cout << "введіть число, яке буде нижньою умовою закінчення циклу: ";
    cin >> eFromUser;

    if (cin.fail()) { // перевірка на відсутність помилок при отриманні введеного
від користувача числа
        cout << "\nумова закінчення циклу повинна бути числом\n";
        cin.clear();
        cin.ignore();
        goto step1;
    }

    if (!(abs(eFromUser) < e)) {
        cout << "\nмодуль умови закінчення циклу має бути менше 10-5\n";
        cin.clear();
        cin.ignore();
        goto step1;
    }

    step2:

    cout << "введіть число, яке буде верхньою умовою закінчення циклу: ";
    cin >> gFromUser;

    if (cin.fail()) { // перевірка на відсутність помилок при отриманні введеного
від користувача числа
        cout << "\нумова закінчення циклу повинна бути числом\n";
        cin.clear();
        cin.ignore();
        goto step2;
    }

    if (!(abs(gFromUser) > g)) {
        cout << "\нмодуль умови закінчення циклу має бути більше 102\n";
    }
}

```

```

        cin.clear();
        cin.ignore();
        goto step2;
    }

    auto [result, sum] = isSeriesConvergent(eFromUser, gFromUser);

    seriesConvergencePrintToConsole(result, sum);
}

// Головна функція
int main() {
    Step1:
    int choice;

    cout << "\nОберіть задачу для виконання:\n";
    cout << "1. Розрахунок параметрів коричневої області\n";
    cout << "2. Обчислення виразу\n";
    cout << "3. Обчислення часткової суми ряду\n";
    cout << "0. Вийти\n";
    cout << "Ваш вибір: ";
    cin >> choice;

    switch (choice) {
        case 1:
            pointsInBrownArea();
            goto Step1;
            break;
        case 2:
            productOfFunction();
            goto Step1;
            break;
        case 3:
            seriesConvergence();
            goto Step1;
            break;
        case 0:
            cout << "Вихід з програми.\n";
            break;
        default:
            cout << "Некоректний вибір. Спробуйте ще раз.\n";
            goto Step1;
    }

    return 0;
}

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

На рис. Б.1 показаний приклад виконання завдання 1.

```

Оберіть задачу для виконання:
1. Розрахунок параметрів коричневої області
2. Обчислення виразу
3. Обчислення часткової суми ряду
0. Вийти
Ваш вибір: 1
Введіть сторону квадрата a: 8
Введіть кількість координат n: 2
Введіть 1 координату x: -1
Введіть 1 координату y: -3
Введіть 2 координату x: 1
Введіть 2 координату y: 3
Кількість точок в коричневій зоні: 1

```

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

На рис. Б.2 показаний приклад виконання завдання 2.

```

Оберіть задачу для виконання:
1. Розрахунок параметрів коричневої області
2. Обчислення виразу
3. Обчислення часткової суми ряду
0. Вийти
Ваш вибір: 3
Введіть число, яке буде нижньою умовою закінчення циклу: 7

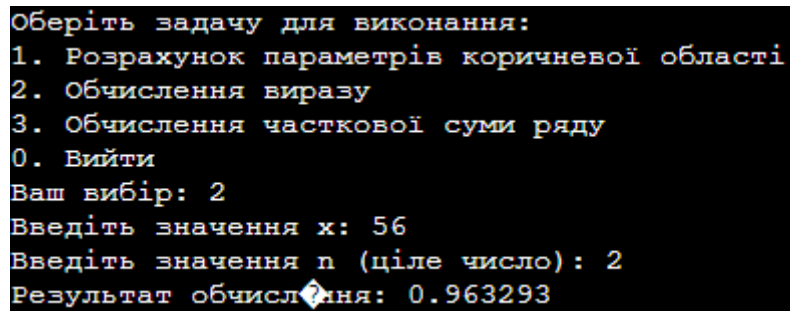
модуль умови закінчення циклу має бути менше 10^-5
Введіть число, яке буде нижньою умовою закінчення циклу: 0,000001
Введіть число, яке буде верхньою умовою закінчення циклу:
умова закінчення циклу повинна бути числом
Введіть число, яке буде верхньою умовою закінчення циклу:
модуль умови закінчення циклу має бути більше 10^2
Введіть число, яке буде верхньою умовою закінчення циклу: 101

Ряд незбіжний
Сума чисел ряду: 2.71828

```

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

На рис. Б.3 показаний приклад виконання завдання 3.



Оберіть задачу для виконання:
1. Розрахунок параметрів коричневої області
2. Обчислення виразу
3. Обчислення часткової суми ряду
0. Вийти
Ваш вибір: 2
Введіть значення x: 56
Введіть значення n (ціле число): 2
Результат обчислення: 0.963293

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