

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 №3**  
з дисципліни «Алгоритмізація та програмування»  
на тему «"Реалізація алгоритмів з розгалуженням мовою C ++"»

XAI.301.175.318.13 ЛР

Виконав \_\_\_\_\_ студент гр. № 319а  
\_\_\_\_\_ Михайло ТЮТЮННИК  
(підпис, дата) (П.І.Б.)

Перевірів к.т.н., доцент  
(вчена ступінь, вчене звання)  
\_\_\_\_\_ Олена ГАВРИЛЕНКО  
(підпис, дата) (П.І.Б.)

## МЕТА РОБОТИ

Вивчити теоретичний матеріал щодо синтаксису у мові C++ і подання у вигляді UML діаграм активності алгоритмів з розгалуженням та реалізувати алгоритми з використанням інструкцій умовного переходу і вибору мовою C++ в середовищі Visual Studio. Також опанувати та відпрацювати навички структурування програми з функціями.

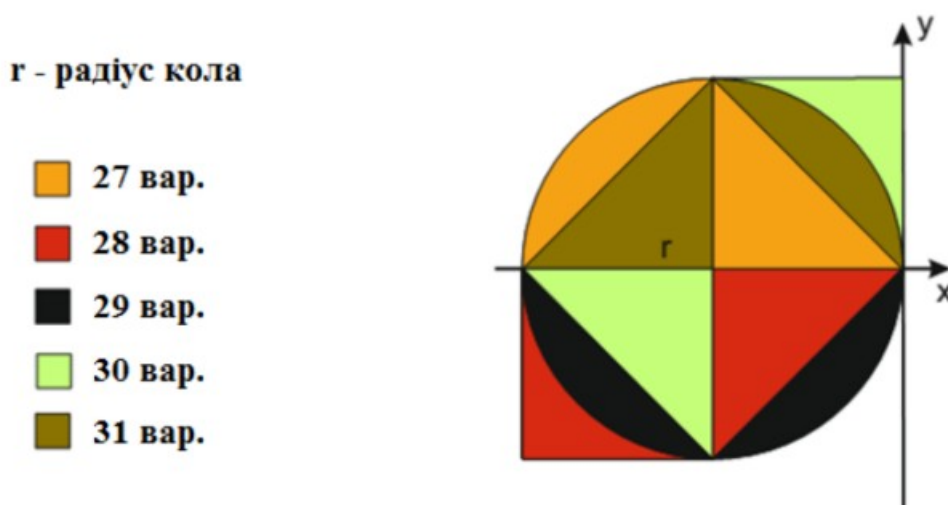
## ПОСТАНОВКА ЗАДАЧІ

Завдання 1. If19. Вирішити завдання на алгоритми з розгалуженням.

If19. Дано чотири цілих числа, одне з яких відмінно від трьох інших, рівних між собою. Визначити порядковий номер числа, відмінного від інших.

Завдання 2. Дано координати точки на площині (x, y). Визначити, чи потрапляє точка в фігуру заданого кольору (або групу фігур) і вивести відповідне повідомлення. Обчислити площу і периметр плоскої фігури.

Таблиця 2– Геометричні фігури №31



Завдання 4. Для вибору користувачем одного з трьох зазначених вище завдань розробити алгоритм організації меню в командному вікні з використанням інструкції вибору.

## ВИКОНАННЯ РОБОТИ

### Завдання 1

Вирішення задачі If19.

Вхідні дані:

Користувач вводить чотири цілих числа: a, b, c, d. Ці числа можуть бути будь-якими цілими числами.

Вихідні дані:

Програма визначає позицію числа, яке відрізняється від трьох інших. Якщо всі числа однакові, виводиться повідомлення про відсутність такого числа.

Алгоритм роботи програми:

- Крок 1: Програма запитує у користувача чотири цілих числа.  
Повідомлення для вводу: "Enter four integers (separated by spaces):".
- Крок 2: Перевіряється коректність вводу.  
Якщо користувач ввів нецілі числа, програма виводить повідомлення "Wrong input! Must be integers." і завершує роботу.
- Крок 3: Логіка визначення "особливого" числа:
  - Якщо перше число a відрізняється від трьох інших, виводиться: "The odd number is at position: 1".
  - Якщо друге число b відрізняється, виводиться: "The odd number is at position: 2".
  - Якщо третє число c відрізняється, виводиться: "The odd number is at position: 3".
  - Якщо четверте число d відрізняється, виводиться: "The odd number is at position: 4".
- Крок 4: Якщо всі числа однакові, програма виводить: "No odd number found!".

Лістинг коду до завдання 1 (If19.) наведено в дод. А (стор. 6).

Зображення вікна виконання роботи наведено в дод. Б(стор. 9, рис. 1 , 2)

### Завдання 2

Вирішення задачі Таблиця 2 – Геометричні фігури

Вхідні дані:

- Ім'я змінної: x
  - Опис: Координата точки по осі X.
  - Тип: Дійсне число (double).
  - Обмеження: Немає.
- Ім'я змінної: y
  - Опис: Координата точки по осі Y.

- Тип: Дійсне число (double).
- Обмеження: Немає.

Вихідні дані:

- Ім'я змінної: `isInFourthQuadrant`
  - Опис: Логічне значення, що визначає, чи точка з координатами (x, y) лежить у четвертій координатній чверті.
  - Тип: Логічний тип (bool).

Алгоритм роботи програми:

- Крок 1: Введення значення координати x.
  - Вивести запит для користувача: "Enter the x-coordinate: ".
- Крок 2: Введення значення координати y.
  - Вивести запит для користувача: "Enter the y-coordinate: ".
- Крок 3: Перевірка належності точки четвертій координатній чверті.
  - Установити значення логічної змінної: `isInFourthQuadrant = (x > 0 && y < 0)`.
- Крок 4: Якщо `isInFourthQuadrant == true`, вивести повідомлення: "The point (x, y) is in the fourth quadrant."
- Крок 5: Якщо `isInFourthQuadrant == false`, вивести повідомлення: "The point (x, y) is not in the fourth quadrant."

Лістинг коду до завдання 1 (Таблиця 2– Геометричні фігури) наведено в дод. А (стор. 6).

Зображення вікна виконання роботи наведено в дод. Б (стор. 9, рис. 3 , 4)

Завдання 3

Вирішення задачі Таблиця 2 – Геометричні фігури

Вхідні дані:

Користувач вводить три сторони трикутника: a, b та c. Усі вони повинні бути додатними дійсними числами.

Вихідні дані:

Програма обчислює та виводить площу трикутника (area) і його периметр (perimeter).

Алгоритм:

Користувач вводить послідовно три значення сторін трикутника: a, b, c.

Спершу перевіряється, чи всі сторони є додатними числами. Якщо хоча б одна зі сторін недопустима (менше або дорівнює нулю), програма завершує роботу із повідомленням: "All sides must be positive!"

Далі перевіряється умова існування трикутника: сума будь-яких двох сторін має бути більшою за третю сторону. Якщо ця умова не виконується,

виводиться повідомлення: "A triangle with such sides cannot exist!" і програма завершується.

Якщо сторони задовольняють умову існування трикутника, обчислюється його периметр як сума трьох сторін:

$$\text{perimeter} = a + b + c$$

Напівпериметр, необхідний для обчислення площі, визначається як половина периметра:

$$s = \text{perimeter} / 2$$

Для обчислення площі використовується формула Герона:

$$\text{area} = \sqrt{(s * (s - a) * (s - b) * (s - c))}$$

Після обчислення площі та периметра результати виводяться у вигляді:

"Perimeter of the triangle: <значення периметра>"

"Area of the triangle: <значення площі>"

Лістинг коду до завдання 1 (Таблиця 2– Геометричні фігури) наведено в дод. А (стор. 6).

Зображення вікна виконання роботи наведено в дод Б (стор. 9, рис. 5, 6)

## Висновки

Отже, було вивчено синтаксис мови C++ та реалізовано алгоритми з умовними переходами. Створено програму для обробки даних і перевірки умов за допомогою бібліотеки `cmath`. Також опрацьовано створення UML-діаграм активності для відображення алгоритмів з розгалуженням. Усі завдання були реалізовані в середовищі Visual Studio, що сприяло покращенню навичок програмування та структуризації програм.

## ДОДАТОК А

### Лістинг коду програми

```
#include <iostream>
#include <cmath> // Для математичних функцій, таких як sqrt (квадратний корінь)
using namespace std;

// Прототипи функцій
void task_if19(); // Завдання 1: знаходження "особливого" числа (If19)
void task_table2(); // Завдання 2: перевірка належності точки певній фігурі (Table 2 - Geometric figures №31)
void task_geometry(); // Завдання 3: обчислення площі та периметру плоскої фігури

int main() {
    int menu; // Змінна для вибору завдання

    cout << "Task number (1 - If19, 2 - Table 2 - Geometric figures №31, 3 - Geometry - Area and Perimeter): ";
    cin >> menu; // Введення номера завдання

    // Перемикання між завданнями за допомогою оператора switch
    switch (menu) {
        case 1:
            task_if19(); // Виклик функції для завдання 1
            break;
        case 2:
            task_table2(); // Виклик функції для завдання 2
            break;
        case 3:
            task_geometry(); // Виклик функції для завдання 3
            break;
        default:
            // Якщо введено неправильний номер завдання, вивести повідомлення
            cout << "Wrong task! (Only 1, 2, 3)" << endl;
    }

    system("pause"); // Затримка перед завершенням програми (потрібно для Windows)
    return 0; // Завершення програми
}

// Реалізація завдання 1 (If19)
void task_if19() {
    int a, b, c, d; // Чотири числа для перевірки
    cout << "***** Task 1: If19 - Odd Number *****" << endl;
    cout << "Enter four integers (separated by spaces): ";
    cin >> a >> b >> c >> d; // Введення чотирьох чисел

    // Перевірка, чи ввід був коректним (якщо ввели не числа)
    if (!cin) {
        cout << "Wrong input! Must be integers." << endl;
    }
    else {
        // Перевірка, яке з чисел є "особливим" (відрізняється від інших)
        if (a != b && a != c && a != d)
            cout << "The odd number is at position: 1" << endl; // Якщо "особливе" число в першій позиції
    }
}
```

```

else if (b != a && b != c && b != d)
    cout << "The odd number is at position: 2" << endl; // Якщо "особливе" число в другій позиції
else if (c != a && c != b && c != d)
    cout << "The odd number is at position: 3" << endl; // Якщо "особливе" число в третій позиції
else if (d != a && d != b && d != c)
    cout << "The odd number is at position: 4" << endl; // Якщо "особливе" число в четвертій
    позиції
else
    // Якщо всі числа однакові, "особливого" числа немає
    cout << "No odd number found!" << endl;
}
}

```

// Реалізація завдання 2 (Table 2 - Geometric figures №31)

```

void task_table2() {
    double x, y, r; // Оголошення змінних для координат точки та радіусу

    // Введення спільного радіусу для трикутника та кола
    cout << "Enter the radius r of the region: ";
    cin >> r; // Введення радіусу
    if (!cin || r <= 0) { // Перевірка на коректність введеного значення
        cout << "Radius must be numeric and positive!" << endl; // Якщо радіус некоректний, вивести
        помилку
        return; // Завершити функцію
    }

    // Введення координат точки
    cout << "Enter the coordinates of the point (x, y): ";
    cin >> x >> y; // Введення координат точки
    if (!cin) { // Якщо координати введено некоректно
        cout << "Coordinates must be numeric!" << endl;
        return; // Завершити функцію
    }

    // Перевірка, чи точка належить трикутнику (умови перевіряють належність трикутнику)
    if (x <= 0 && y >= 0 && (y <= -x) && (x * x + y * y <= r * r)) {
        cout << "A point in the brown region" << endl; // Якщо точка в трикутнику
        return; // Завершити функцію
    }

    // Якщо точка не в трикутнику, перевіряємо її належність частині кола
    if (x >= 0 && y >= 0 && (x * x + y * y <= r * r) && (x + y >= r)) {
        cout << "A point in the brown region" << endl; // Якщо точка в частині кола
    }
    else {
        cout << "The point is not in the brown region" << endl; // Якщо точка не належить жодній з фігур
    }
}

```

// Реалізація завдання 3 (Geometry - Area and Perimeter)

```

void task_geometry() {
    double side, hypotenuse, area, perimeter; // Змінні для введення сторони та обчислених
    результатів

    // Введення довжини сторони рівнобедреного трикутника

```

```
cout << "Enter the radius : ";
cin >> side; // Введення сторони

// Перевірка на коректність введеного значення
if (side <= 0) { // Якщо введено некоректне значення сторони
    cout << "The length of the side must be a positive number." << endl; // Вивести повідомлення про
помилку
    return; // Завершити функцію
}

// Обчислення гіпотенузи рівнобедреного трикутника (не використовується для виводу, тільки
для подальших обчислень)
hypotenuse = sqrt(2) * side; // Гіпотенуза рівнобедреного прямокутного трикутника:  $\sqrt{2}$  *
сторона

// Обчислення площі рівнобедреного трикутника
area = (side * side) / 2; // Площа трикутника:  $(1/2) * \text{сторона} * \text{сторона}$ 

// Обчислення периметра рівнобедреного трикутника
perimeter = 2 * side + hypotenuse; // Периметр трикутника: дві сторони + гіпотенуза

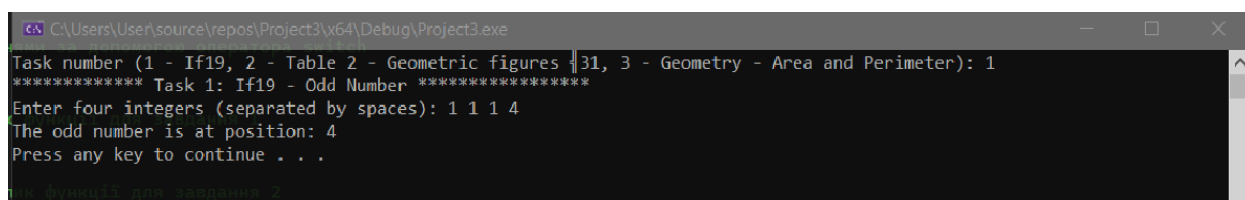
// Виведення результатів
cout << "Area: " << area << endl; // Виведення площі
cout << "Perimeter: " << perimeter << endl; // Виведення периметра
}
```



## ДОДАТОК Б

### Скрін-шоти вікна виконання програми

На рисунку 1 , 2 зображена робота програми до завдання If 19

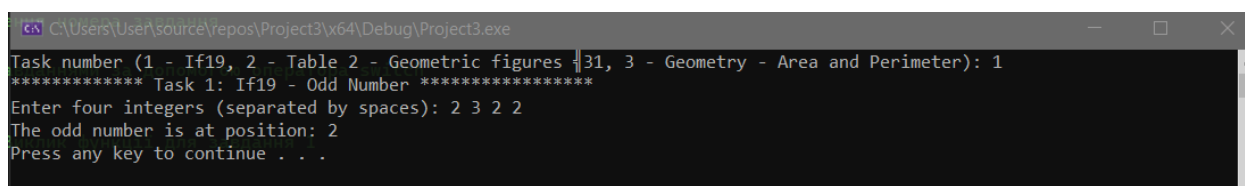


```

C:\Users\User\source\repos\Project3\Debug\Project3.exe
Task number (1 - If19, 2 - Table 2 - Geometric figures, 31, 3 - Geometry - Area and Perimeter): 1
***** Task 1: If19 - Odd Number *****
Enter four integers (separated by spaces): 1 1 1 4
The odd number is at position: 4
Press any key to continue . . .

```

Рисунок 1 — Результат роботи програми



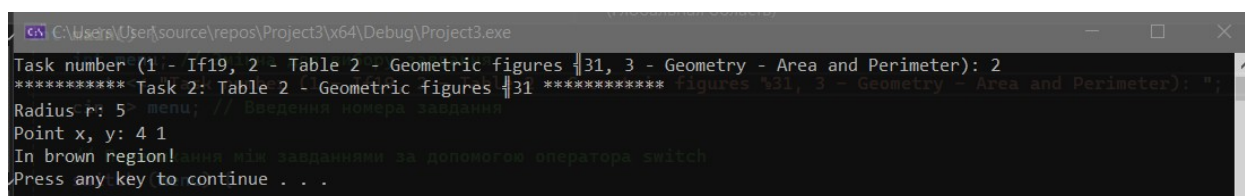
```

C:\Users\User\source\repos\Project3\Debug\Project3.exe
Task number (1 - If19, 2 - Table 2 - Geometric figures, 31, 3 - Geometry - Area and Perimeter): 1
***** Task 1: If19 - Odd Number *****
Enter four integers (separated by spaces): 2 3 2 2
The odd number is at position: 2
Press any key to continue . . .

```

Рисунок 2 — Результат роботи програми

На рисунку 3 , 4 зображена робота програми до завдання Таблиця – Геометричні фігури варіант № 31



```

C:\Users\User\source\repos\Project3\Debug\Project3.exe
Task number (1 - If19, 2 - Table 2 - Geometric figures, 31, 3 - Geometry - Area and Perimeter): 2
***** Task 2: Table 2 - Geometric figures *****
Radius r: 5
Point x, y: 4 1
In brown region!
Press any key to continue . . .

```

Рисунок 3 — Результат роботи програми

```

C:\Users\User\source\repos\Project3\Debug\Project3.exe, 3 - Geometry - Area and Perimeter): *;
Task number (1 - If19, 2 - Table 2 - Geometric figures #31, 3 - Geometry - Area and Perimeter): 2
***** Task 2: Table 2 - Geometric figures #31 *****
Radius r: 5
Point x, y: 6 6
Out of brown region!
Press any key to continue . . .

```

Рисунок 4 — Результат роботи програми

На рисунку 5 , 6 зображена робота програми до завдання Таблиця – Геометричні фігури варіант № 31 обчислення площі і перимета плоскої фігури.

```

C:\Users\User\source\repos\Project3\Debug\Project3.exe
Task number (1 - If19, 2 - Table 2 - Geometric figures #31, 3 - Geometry - Area and Perimeter): 3
***** Task 3: Geometry - Area and Perimeter *****
Enter lengths of three sides of a triangle (a, b, c): 5 5 5
Perimeter: 15
Area: 10.8253
Press any key to continue . . .

```

Рисунок 5 - Результат роботи програми

```

C:\Users\User\source\repos\Project3\Debug\Project3.exe
Task number (1 - If19, 2 - Table 2 - Geometric figures #31, 3 - Geometry - Area and Perimeter): 3
***** Task 3: Geometry - Area and Perimeter *****
Enter lengths of three sides of a triangle (a, b, c): 3 3 3
Perimeter: 9
Area: 3.89711
Press any key to continue . . .

```

Рисунок 6 - Результат роботи програми

На рисунку 7 зображено діаграму активності всієї програми

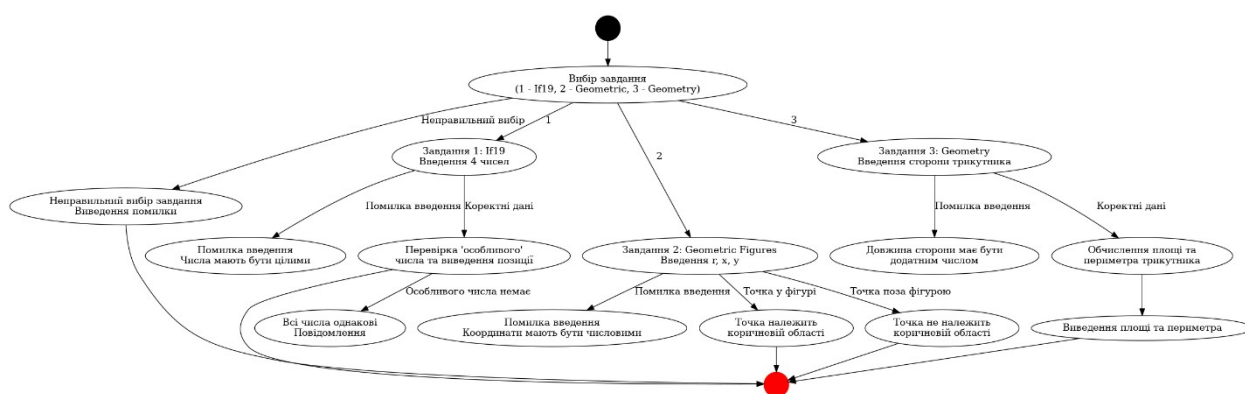


Рис.7 Діаграма активності програми

На рисунку 8 зображено діаграму активності програми до завдання If19

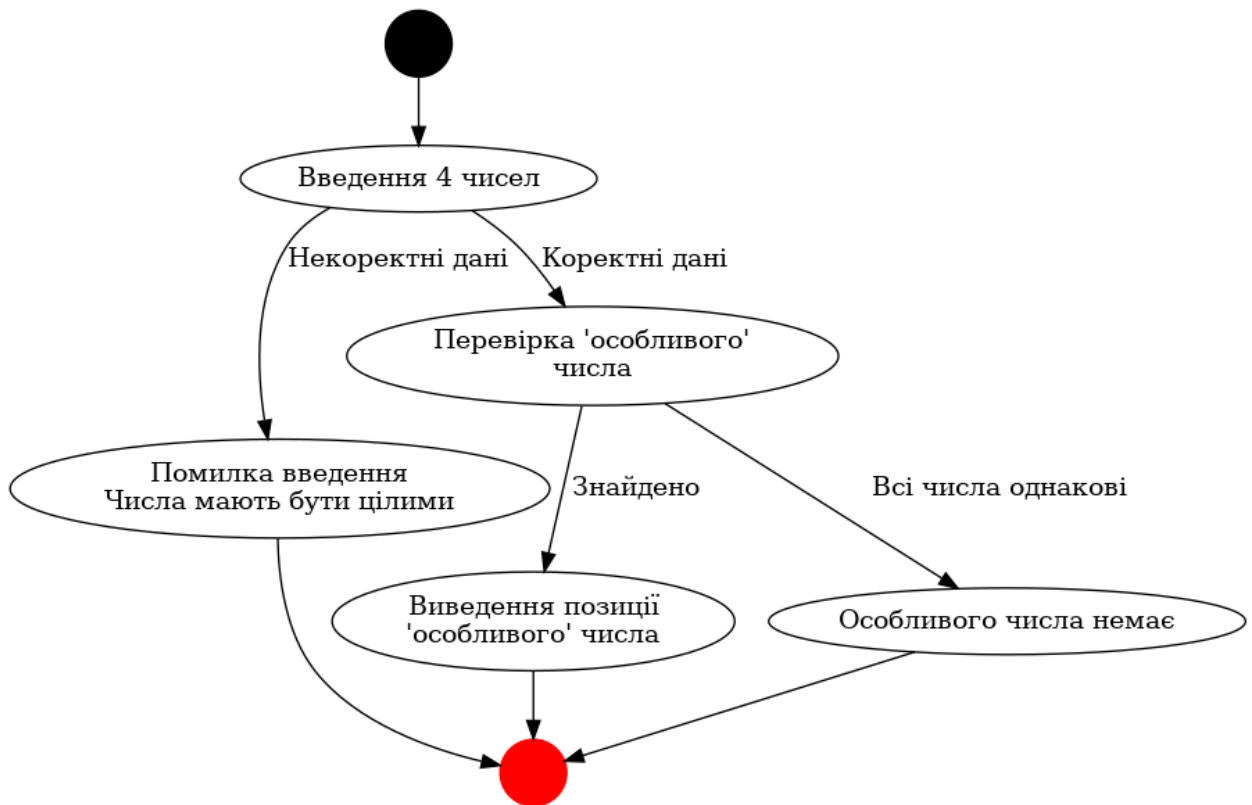


Рис.8 Діаграма активності програми

На рисунку 9 зображено діаграму активності програми до завдання 2

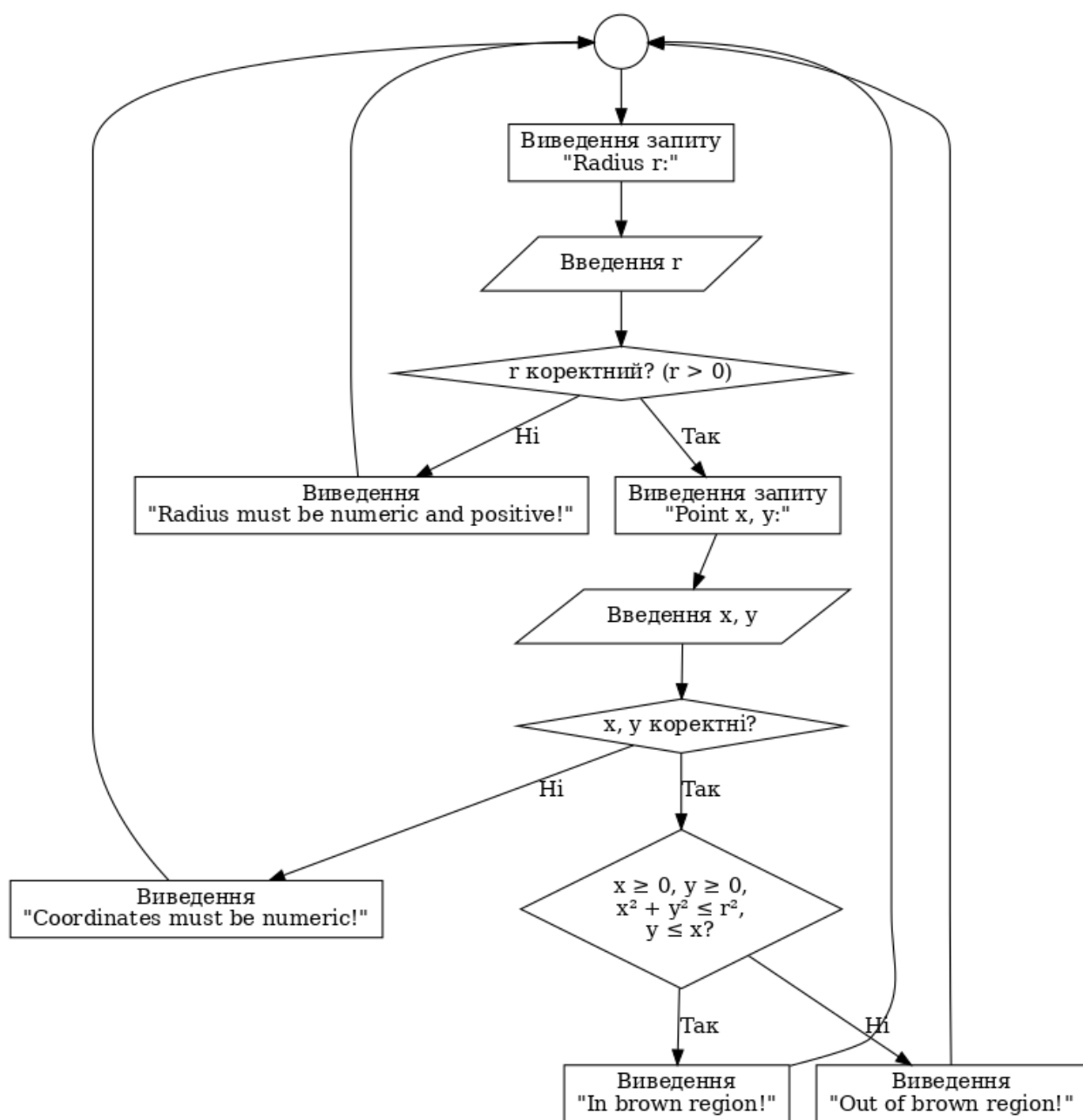


Рис.9 Діаграма активності програми