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 № 10

з дисципліни «Алгоритмізація та програмування»
Тема: "Створення і обробка структур даних мовою С ++"

ХАІ.301.141.319а. 27 ЛР

Виконав студент гр. 319а
Владислав РУДНЄВ
(Підпис,дата) (П.І.Б.)

Перевірив к.т.н., доц.
(вчена ступінь, вчене звання)
Олена ГАВРИЛЕНКО
(підпис, дата) (П.І.Б.)

2024

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення структур (записів) мовою С ++, а також їх передачі в функції, і реалізувати декларування і обробку структур мовою С ++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити задачу зі структурами даних.

Param59. Описати тип TDate - запис з полями цілого типу Day (день), Month (місяць) і Year (рік) - і функцію LeapYear (D) логічного типу з параметром типу TDate, яка повертає True, якщо рік у даті D є високосним, і False в іншому випадку. Вивести значення функції LeapYear для п'яти даних дат (передбачається, що всі дати є правильними) .Вісокосним вважається рік, ділиться на 4, за винятком тих років, які діляться на 100 і не діляться на 400.

Завдання 2. Для задач з табл.2-3:

A. Описати структуру, яка містить всі вхідні і всі вихідні дані задачі.

B. Визначити функцію (*метод), що реалізує обробку структури відповідно до задачі.

C. Визначити функцію (*метод), що перевіряє на коректність і заповнює відповідні поля вхідних даних стуктури

D. Викликати функції (*методи) з пунктів С, В після оголошення змінної (об’єкту) структури.

E. Вивести значення полів вихідних даних.

Begin18. Дано три точки A, B, C на числової осі. Точка C розташована між точками A і B.Знайти твір довжин відрізків AC і BC.

Boolean27. Дано числа x, y. Перевірити істинність висловлювання: «Точка з координатами (x, y) лежить в другій або третій координатній чверті».

Завдання 3. Рішення всіх трьох задач реалізувати в одному консольному додатку, *структурувати на модулі.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Param59.

Вхідні дані:

- day — день місяця.
  - Тип: int.
  - Обмеження: від 1 до 31.

- month — номер місяця.
  - Тип: int.
  - Обмеження: від 1 до 12.

- year — рік.
  - Тип: int.
  - Обмеження: позитивне ціле число (наприклад, від 1 до 9999).

Вихідні дані:

Результат типу bool (логічне значення):

- true — якщо рік у введеній даті є високосним;

- false — якщо рік не є високосним.

Алгоритм:

1. Створити структуру TDate з полями day, month, year.
2. Реалізувати функцію LeapYear, яка приймає об'єкт TDate як параметр.
3. Усередині функції реалізувати перевірку високосного року за наступними правилами:
   - Якщо рік ділиться на 400 — це високосний рік.
   - Інакше, якщо рік ділиться на 100 — це не високосний рік.
   - Інакше, якщо рік ділиться на 4 — це високосний рік.
     В усіх інших випадках — не високосний.
4. Створити 5 прикладів дат і викликати для кожної з них функцію LeapYear.
5. Вивести на екран рік і результат перевірки (Yes/No).

Завдання 2.

Вирішення задачі Begin18.

Вхідні дані:

- A — координата точки A на числовій осі.
  - Тип: double.
  - Обмеження: довільне дійсне число.
- B — координата точки B.
  - Тип: double.
  - Обмеження: довільне дійсне число.
- C — координата точки C, яка обов’язково розташована між A і B.
  - Тип: double.

Вихідні дані (ім’я, опис, тип):

- product — добуток довжин відрізків AC та BC. Тип: double.

Алгоритм:

1. Створити структуру SegmentPoints, яка містить поля: A, B, C, product.
2. Реалізувати функцію CalculateSegmentProduct, яка приймає структуру як параметр за посиланням.
3. Усередині функції обчислити:
   - довжину відрізка AC як |C - A|
   - довжину відрізка BC як |C - B|
   - добуток цих довжин зберегти у product
4. У функції RunSegmentTest запросити в користувача значення A, B, C.
5. Викликати CalculateSegmentProduct і вивести результат.

Завдання 3.

Виконання Boolean27.

Вхідні дані:

- x — абсциса точки (горизонтальна координата).
    - Тип: double.
- y — ордината точки (вертикальна координата).
    - Тип: double.

Вихідні дані:

- isInSecondOrThird — логічна змінна, яка визначає, чи точка знаходиться у 2-й або 3-й координатній чверті.
    - Тип: bool.

Алгоритм:

1. Створити структуру Point, що містить x, y та логічне поле isInSecondOrThird.
2. Реалізувати функцію CheckQuadrant, яка приймає структуру Point за посиланням.
3. У тілі функції встановити isInSecondOrThird = true, якщо:
    - $x < 0$ і $y \neq 0$ (тобто точка лівіше осі Y та не лежить на ній).
4. Створити функцію RunPointTest, яка:
    - зчитує координати x і y від користувача,
    - викликає CheckQuadrant,
    - виводить результат.

ВИСНОВОК

Під час виконання роботи було засвоєно основи використання структур у C++ та їх передавання у функції. У середовищі Visual Studio реалізовано створення, обробку та тестування структур, що підтвердило практичне розуміння теми та вміння застосовувати її для розв’язання задач.

ДОДАТОК А
Лістинінг коду

№main.cpp

```
#include <iostream>
#include "Param59.h"
#include "Begin18.h"
#include "Begin27.h"
using namespace std;

/// Вивід текстового меню для користувача
void ShowMenu() {
    cout << "\n========== MENU ==========\n";
    cout << "1. Leap year check (Task 1)\n";
    cout << "2. Calculate product of segments AC and BC (Task 2A)\n";
    cout << "3. Coordinate quadrant check (II or III) (Task 2B)\n";
    cout << "0. Exit\n";
    cout << "==========================\n";
    cout << "Your choice: ";
}

int main() {
    int choice;

    // Main menu loop: runs until user selects 0
    do {
        ShowMenu();
        std::cin >> choice;

        switch (choice) {
        case 1:
            std::cout << "\n--- Task 1: Leap Year ---\n";
            RunLeapYearTest();
            break;

        case 2:
            std::cout << "\n--- Task 2A: Segments AC and BC ---\n";
            RunSegmentTest();
            break;

        case 3:
            std::cout << "\n--- Task 2B: Coordinate Quadrant ---\n";
            RunPointTest();
            break;

        case 0:
            std::cout << "Program terminated.\n";
```

```
            break;

        default:
            std::cout << "Invalid choice! Please try again.\n";
        }

    } while (choice != 0);

    return 0;
}
```

№Param59.cpp

```
#include <iostream>
#include "Param59.h"
using namespace std;

/// Функція перевіряє, чи є рік високосним.
/// Високосний рік:
///   - кратний 4
///   - але не кратний 100, за винятком кратних 400
bool LeapYear(const TDate& d) {
    return (d.year % 4 == 0 && d.year % 100 != 0) || (d.year % 400 == 0);
}

/// Тестова функція для перевірки LeapYear на 5 різних датах.
/// Дати вводяться в масив вручну, як приклади.
/// Для кожної дати виводиться: рік — високосний чи ні.
void RunLeapYearTest() {
    // Ініціалізація масиву з 5 прикладами дат
    TDate dates[5] = {
        {1, 1, 2000}, // високосний
        {1, 1, 1900}, // не високосний (кратний 100, не кратний 400)
        {1, 1, 2024}, // високосний
        {1, 1, 2023}, // не високосний
        {1, 1, 2100}  // не високосний
    };

    cout << "Leap Year Check:\n";

    // Перевірка кожної дати з масиву
    for (int i = 0; i < 5; ++i) {
        cout << dates[i].year << " -> "
            << (LeapYear(dates[i]) ? "Yes (intercalary)" : "No (normal)") << endl;
    }
}
```

№Param59.h

```
#pragma once  // Гарантує, що цей заголовочний файл буде включено лише один раз при компіляції

// Структура TDate представляє дату з трьома цілими полями:
// day — день місяця
// month — номер місяця (1–12)
// year — повний рік (наприклад, 2024)
struct TDate {
    int day;    // День місяця (1–31)
    int month;  // Місяць (1–12)
    int year;   // Рік (наприклад, 2024)
};

// Функція LeapYear приймає константне посилання на об'єкт типу TDate,
// і повертає true, якщо рік у цій даті є високосним, інакше — false.
//
// Правило високосного року:
//   - рік кратний 4, але не кратний 100,
//   - або кратний 400
bool LeapYear(const TDate& d);

// Допоміжна функція RunLeapYearTest виконує тестування функції LeapYear:
// створює кілька прикладів дат, перевіряє їх на високосність
// і виводить результати на екран.
void RunLeapYearTest();
```

№Begin18.cpp

```
#include <iostream>
#include "Begin18.h"
using namespace std;

/// Перевіряє, чи точка знаходиться у 2-й або 3-й координатній чверті.
/// II чверть: x < 0, y > 0
/// III чверть: x < 0, y < 0
/// Всі точки з x < 0, y ? 0 — задовольняють умові.
void CheckQuadrant(Point& p) {
    p.isInSecondOrThird = (p.x < 0 && (p.y != 0));
}

/// Функція вводить координати точки, перевіряє її положення
/// і виводить True/False відповідно до умови задачі.
void RunPointTest() {
    Point p;

    cout << "\nEnter x coordinate: ";
    cin >> p.x;
```

```
    cout << "Enter y coordinate: ";
    cin >> p.y;

    // Перевірка координатної чверті
    CheckQuadrant(p);

    // Вивід результату
    cout << "Point belongs to the II or III quadrant: "
        << (p.isInSecondOrThird ? "True (True)" : "False (False)") << endl;
}
```

№Begin18.h

```
#pragma once  // Директива, що запобігає багаторазовому включенню заголовка при компіляції

// Структура Point описує точку на декартовій площині (x, y)
// та результат перевірки: чи належить вона другій або третій координатній чверті.
struct Point {
    double x;  // Абсциса точки (горизонтальна координата)
    double y;  // Ордината точки (вертикальна координата)

    // Логічний результат перевірки:
    // true — якщо точка знаходиться у другій або третій чверті,
    // false — в іншому випадку
    bool isInSecondOrThird;
};

// Функція CheckQuadrant приймає посилання на структуру Point,
// перевіряє координати точки та встановлює значення isInSecondOrThird
// відповідно до її розташування в координатній площині.
void CheckQuadrant(Point& p);

// Функція RunPointTest створює приклад точки, викликає функцію перевірки чверті,
// і виводить результат (чи знаходиться точка в II або III чверті).
void RunPointTest();
```

№Boolean27.cpp

```
#include <iostream>
#include <cmath>
#include "Boolean27.h"
using namespace std;

/// Обчислення добутку довжин відрізків AC і BC.
/// Формула: |C - A| * |C - B|
/// Вхід: структура SegmentPoints з координатами A, B, C.
/// Результат зберігається в полі product.
void CalculateSegmentProduct(SegmentPoints& sp) {
    sp.product = abs(sp.C - sp.A) * abs(sp.C - sp.B);
```

```
}

/// Функція, яка запитує користувача ввести точки A, B, C.
/// Перевірку, що C між A і B, ми не робимо — за умовою довіряємо користувачу.
/// Потім викликається обчислення і виводиться результат.
void RunSegmentTest() {
   SegmentPoints sp;

   cout << "\nEnter the coordinate of point A: ";
   cin >> sp.A;

   cout << "Enter the coordinate of point B: ";
   cin >> sp.B;

   cout << "Enter the coordinate of point C (between A and B): ";
   cin >> sp.C;

   // Calculating the product of segment lengths
   CalculateSegmentProduct(sp);

   // Outputting the result
   cout << "Product of segment lengths AC and BC = " << sp.product << endl;
}
```

№Boolean27.h

```
#pragma once  // Гарантує, що файл буде підключено лише один раз при компіляції

// Структура SegmentPoints призначена для зберігання координат трьох точок A, B, C на числовій осі,
// а також результату обчислення — добутку довжин відрізків AC і BC.
struct SegmentPoints {
   double A;       // Координата точки A
   double B;       // Координата точки B
   double C;       // Координата точки C (умовою задачі передбачено, що вона розташована між A і B)
   double product;  // Результат: добуток довжин відрізків AC і BC
};

// Функція CalculateSegmentProduct приймає посилання на структуру SegmentPoints,
// обчислює довжини відрізків AC та BC, множить їх і записує результат у поле product структури.
void CalculateSegmentProduct(SegmentPoints& sp);

// Функція RunSegmentTest використовується для тестування роботи CalculateSegmentProduct.
// Вона створює приклад об'єкта SegmentPoints, заповнює його даними,
// викликає функцію обчислення, а потім виводить результат.
void RunSegmentTest();
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми та діаграми

На рис.1 показано код виконання програми лабораторна робота№9

```
========== MENU ==========
1. Leap year check (Task 1)
2. Calculate product of segments AC and BC (Task 2A)
3. Coordinate quadrant check (II or III) (Task 2B)
0. Exit
==========================
Your choice: 1

--- Task 1: Leap Year ---
Leap Year Check:
2000 -> Yes (intercalary)
1900 -> No (normal)
2024 -> Yes (intercalary)
2023 -> No (normal)
2100 -> No (normal)

========== MENU ==========
1. Leap year check (Task 1)
2. Calculate product of segments AC and BC (Task 2A)
3. Coordinate quadrant check (II or III) (Task 2B)
0. Exit
==========================
Your choice: 2

--- Task 2A: Segments AC and BC ---

Enter the coordinate of point A: 4
Enter the coordinate of point B: 2
Enter the coordinate of point C (between A and B): -8
Product of segment lengths AC and BC = 120

========== MENU ==========
1. Leap year check (Task 1)
2. Calculate product of segments AC and BC (Task 2A)
3. Coordinate quadrant check (II or III) (Task 2B)
0. Exit
==========================
Your choice: 3

--- Task 2B: Coordinate Quadrant ---

Enter x coordinate: 4
Enter y coordinate: -5
Point belongs to the II or III quadrant: False (False)

========== MENU ==========
1. Leap year check (Task 1)
2. Calculate product of segments AC and BC (Task 2A)
3. Coordinate quadrant check (II or III) (Task 2B)
0. Exit
==========================
Your choice: 3

--- Task 2B: Coordinate Quadrant ---

Enter x coordinate: -4
Enter y coordinate: -6
Point belongs to the II or III quadrant: True (True)

========== MENU ==========
1. Leap year check (Task 1)
2. Calculate product of segments AC and BC (Task 2A)
3. Coordinate quadrant check (II or III) (Task 2B)
0. Exit
==========================
Your choice: 0
Program terminated.
```

Рисунок 1 — функціонування коду лабораторної роботи№9

На рис.2 показано диаграму коду програми для завдання Param59

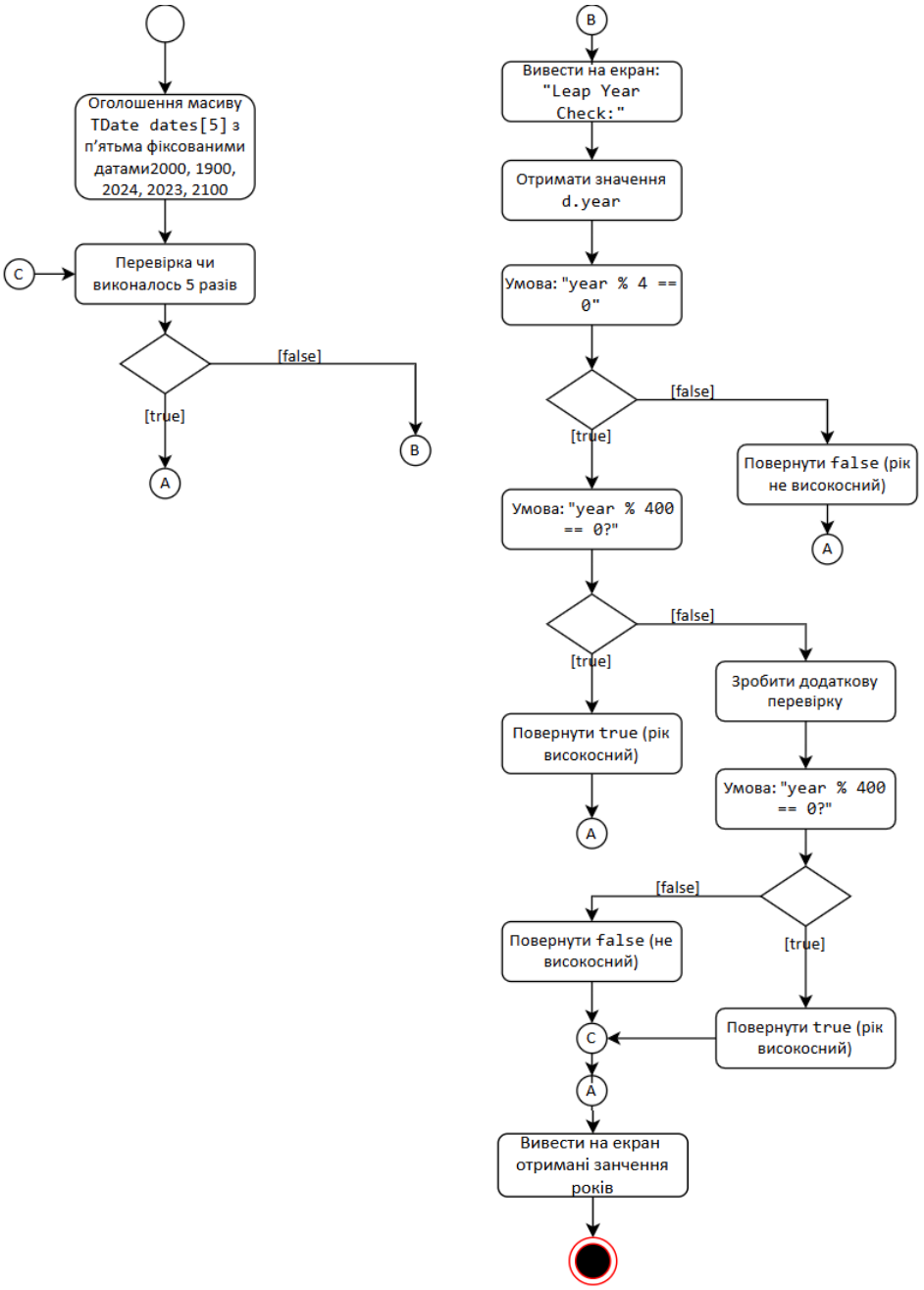

Рисунок 2 — Діаграма активності перевірки високосності року .

На рис.3 показано диаграму коду програми для завдання Begin18

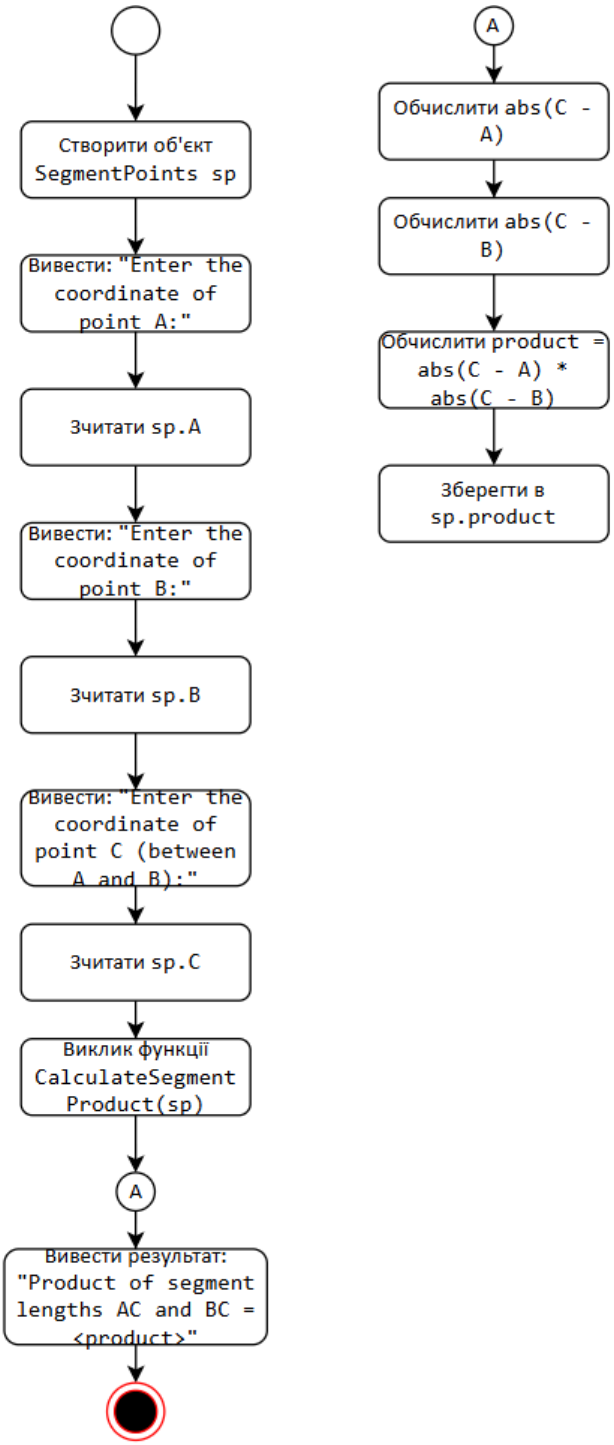

Рисунок 3 — Діаграма активності обчислення координат

На рис.3 показано диаграму коду програми для завдання Boolean27

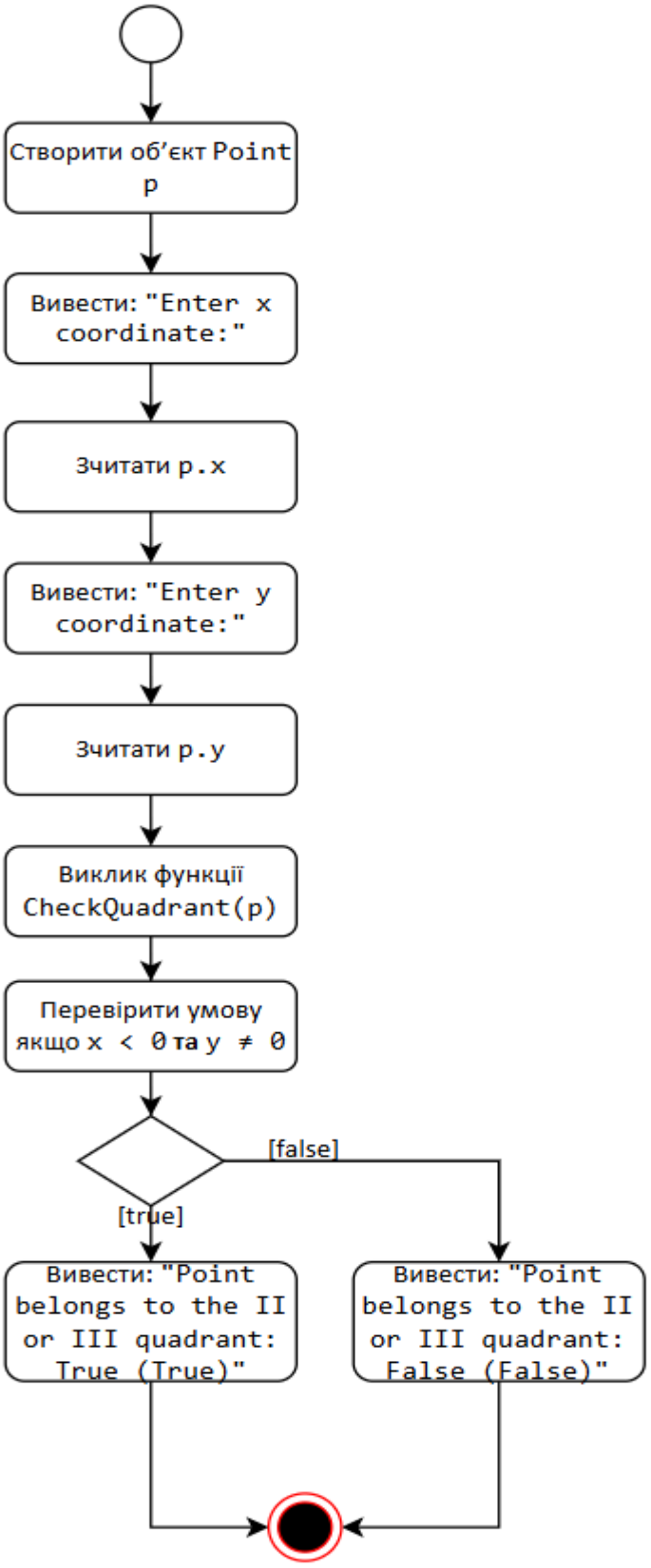

Рисунок 4 — Діаграма активності перевірки належності координат чвертей