

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет  
«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 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 з основ представлення структур (записів) мовою C ++, а також їх передачі в функції, і реалізувати декларування і обробку структур мовою C ++ в середовищі Visual Studio.

## ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити задачу зі структурами даних. Варіанти завдання Param85.

Завдання 2-3. Для задач 1, 2:

- А. Описати структуру, яка містить всі вхідні і всі вихідні дані задачі.
- В. Визначити функцію, що реалізує обробку структури відповідно до задачі.
- С. Визначити функцію, що перевіряє на коректність і заповнює відповідні поля вхідних даних структури
- Д. Викликати функції з пунктів С, В після оголошення змінної (об'єкту) структури.
- Е. Вивести значення полів вихідних даних.

Завдання 4. Рішення всіх трьох задач реалізувати в одному консольному Додатку.

## ВИКОНАННЯ РОБОТИ

Завдання 1 (Робота з часом - Param85)

Опис: Перетворення часу з 12-годинного формату в 24-годинний та перевірка коректності часу.

Вирішення задачі: DemoTimeFunctions().

Вхідні дані:

- TTime - структура з полями Hour, Min, Sec
- PM - bool (чи є час післяполудневим)

Вихідні дані:

- converted - TTime (перетворений час у 24-годинному форматі)
- Коди помилок від CheckTime()

Алгоритм вирішення представлено в додатку Б, рис. Б.1(стор.8).

Завдання 2 (Точки на числовій осі - Begin17)

Опис: Обчислення довжин відрізків AC, BC та їх суми для трьох точок на числовій осі.

Вирішення задачі: DemoLineSegments().

Вхідні дані:

- Point - структура з координатою x
- Три точки A, B, C

Вихідні дані:

- AC - double (довжина відрізка AC)

- BC - double (довжина відрізка BC)
- Sum - double (сума довжин AC і BC)

Алгоритм вирішення представлено в додатку Б, рис. Б.2(стор.9).

### Завдання 3 (Перевірка чисел - Boolean13)

Опис: Перевірка істинності висловлювання "Хоча б одне з чисел А, В, С є позитивним".

Вирішення задачі: DemoNumbersCheck().

Вхідні дані:

- Три цілі числа А, В, С

Вихідні дані:

- result - bool (результат перевірки висловлювання)

Алгоритм вирішення представлено в додатку Б, рис. Б.3(стор.10).

Лістинг коду вирішення задач наведено в додатку А (стор. 4).

Екран роботи програми для першого завдання показаний в додатку Б, рис. Б.4(стор.11).

Екран роботи програми для другого та третього завдання показаний в додатку Б, рис. Б.5(стор.12).

## ВИСНОВКИ

У ході виконання лабораторної роботи була розроблена програма на C++, яка складається з трьох 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> // Для функції abs() у задачі Begin17
using namespace std;

// ===== Завдання 1 (Param85) =====

// Структура для представлення часу (з Param71)
struct TTime {
    int Hour;    // Години (0-23 або 1-12 в залежності від формату)
    int Min;     // Хвилини (0-59)
    int Sec;     // Секунди (0-59)
};

// Функція для перевірки коректності часу в 24-годинному форматі (з Param71)
int CheckTime(TTime T) {
    int errors = 0; // Лічильник помилок (0 - немає помилок)

    if (T.Hour < 0 || T.Hour > 23) errors += 100;
    if (T.Min < 0 || T.Min > 59) errors += 10;
    if (T.Sec < 0 || T.Sec > 59) errors += 1;

    return errors;
}

// Функція для перевірки, чи час вже у 24-годинному форматі
bool Is24HourFormat(TTime T) {
    return CheckTime(T) == 0; // Час коректний у 24-годинному форматі
}

// Функція для перевірки коректності часу в 12-годинному форматі
bool Is12HourFormatValid(TTime T, bool PM) {
    // Для 12-годинного формату години мають бути 1-12, хвилини/секунди 0-59
    return (T.Hour >= 1 && T.Hour <= 12) &&
           (T.Min >= 0 && T.Min <= 59) &&
           (T.Sec >= 0 && T.Sec <= 59);
}

// Функція для перетворення 12-годинного формату в 24-годинний
TTime To24h(TTime T, bool PM) {
    // Якщо час вже у 24-годинному форматі
    if (Is24HourFormat(T) && !PM) {
        return T;
    }

    // Якщо час некоректний для 12-годинного формату
    if (!Is12HourFormatValid(T, PM)) {
        return {0, 0, 0}; // Повертаємо 00:00:00 при помилці
    }

    // Виконуємо перетворення
    int hour24 = T.Hour;
    if (PM) {
        hour24 = (T.Hour == 12) ? 12 : T.Hour + 12;
    } else {
        hour24 = (T.Hour == 12) ? 0 : T.Hour;
    }

    return {hour24, T.Min, T.Sec};
}

void PrintTime(TTime T) {
    printf("%02d:%02d:%02d", T.Hour, T.Min, T.Sec);
}
```

```

}

void DemoTimeFunctions() {
    cout << "=== Завдання 1 (Param85) ===" << endl;

    struct TestCase {
        TTime time;
        bool PM;
        string desc;
    };

    TestCase tests[] = {
        {{3,45,30}, false, "3:45:30 AM → 03:45:30"},
        {{12,0,0}, true, "12:00:00 PM → 12:00:00"},
        {{8,15,0}, false, "8:15:00 AM → 08:15:00"},
        {{6,30,45}, true, "6:30:45 PM → 18:30:45"},
        {{11,59,59}, true, "11:59:59 PM → 23:59:59"},
        {{33,0,0}, false, "33:00:00 AM → 00:00:00 (некоректний)"},
        {{5,60,0}, false, "5:60:00 AM → 00:00:00 (некоректний)"},
        {{7,0,60}, false, "7:00:60 AM → 00:00:00 (некоректний)"},
        {{16,30,0}, false, "16:30:00 → 16:30:00 (вже 24-годинний)"},
        {{0,0,0}, false, "00:00:00 → 00:00:00 (вже 24-годинний)"}
    };

    for (int i = 0; i < 10; i++) {
        auto test = tests[i];
        cout << "Тест " << i+1 << ": " << test.desc << endl;
        cout << "Вхідний час: ";
        PrintTime(test.time);
        cout << " (" << (test.PM ? "PM" : "AM") << ")" << endl;

        TTime result = To24h(test.time, test.PM);
        cout << "Результат: ";
        PrintTime(result);
        cout << endl << endl;
    }
}

// ===== Завдання 2 (Begin17) =====
// Структура для представлення точки на числовій осі
struct Point {
    double x; // Координата точки
};

// Структура для вхідних та вихідних даних задачі
struct LineSegments {
    Point A, B, C; // Вхідні дані - три точки
    double AC, BC, Sum; // Вихідні дані - довжини відрізків та їх сума
};

// Функція для обчислення довжин відрізків
void CalculateSegments(LineSegments &data) {
    // Обчислюємо довжину відрізка AC (модуль різниці координат)
    data.AC = abs(data.C.x - data.A.x);

    // Обчислюємо довжину відрізка BC (модуль різниці координат)
    data.BC = abs(data.C.x - data.B.x);

    // Обчислюємо суму довжин
    data.Sum = data.AC + data.BC;
}

// Функція для введення даних та перевірки їх коректності
void InputAndValidate(LineSegments &data) {
    cout << "Введіть координату точки A: ";
}

```

```

    cin >> data.A.x;

    cout << "Введіть координату точки В: ";
    cin >> data.B.x;

    cout << "Введіть координату точки С: ";
    cin >> data.C.x;

    // У цій задачі всі вхідні дані коректні, оскільки вони просто числа
}

// Демонстрація роботи для задачі з точками
void DemoLineSegments() {
    cout << "\n=== Завдання 2 (Begin17) ===" << endl;

    LineSegments data;

    // Введення даних
    InputAndValidate(data);

    // Обчислення
    CalculateSegments(data);

    // Виведення результатів
    cout << "Довжина відрізка AC: " << data.AC << endl;
    cout << "Довжина відрізка BC: " << data.BC << endl;
    cout << "Сума довжин AC і BC: " << data.Sum << endl;
}

// ===== Завдання 3 (Boolean13) =====
// Структура для вхідних та вихідних даних задачі
struct NumbersCheck {
    int A, B, C; // Вхідні дані - три цілих числа
    bool result; // Вихідні дані - результат перевірки
};

// Функція для перевірки умови
void CheckCondition(NumbersCheck &data) {
    // Умова: хоча б одне з чисел A, B, C позитивне
    data.result = (data.A > 0) || (data.B > 0) || (data.C > 0);
}

// Функція для введення даних
void InputNumbers(NumbersCheck &data) {
    cout << "Введіть число A: ";
    cin >> data.A;

    cout << "Введіть число B: ";
    cin >> data.B;

    cout << "Введіть число C: ";
    cin >> data.C;
}

// Демонстрація роботи для задачі з числами
void DemoNumbersCheck() {
    cout << "\n=== Завдання 3 (Boolean13) ===" << endl;

    NumbersCheck data;

    // Введення даних
    InputNumbers(data);

    // Перевірка умови
    CheckCondition(data);
}

```

```
// Виведення результату
cout << "Хоча б одне з чисел А, В, С позитивне: "
      << (data.result ? "так" : "ні") << endl;
}

// ===== Головна функція =====
int main() {
    // Демонстрація роботи всіх трьох завдань
    DemoTimeFunctions();
    DemoLineSegments();
    DemoNumbersCheck();

    return 0;
}
```

## ДОДАТОК Б

На рис.Б.1 показано діаграму активності для першого завдання.

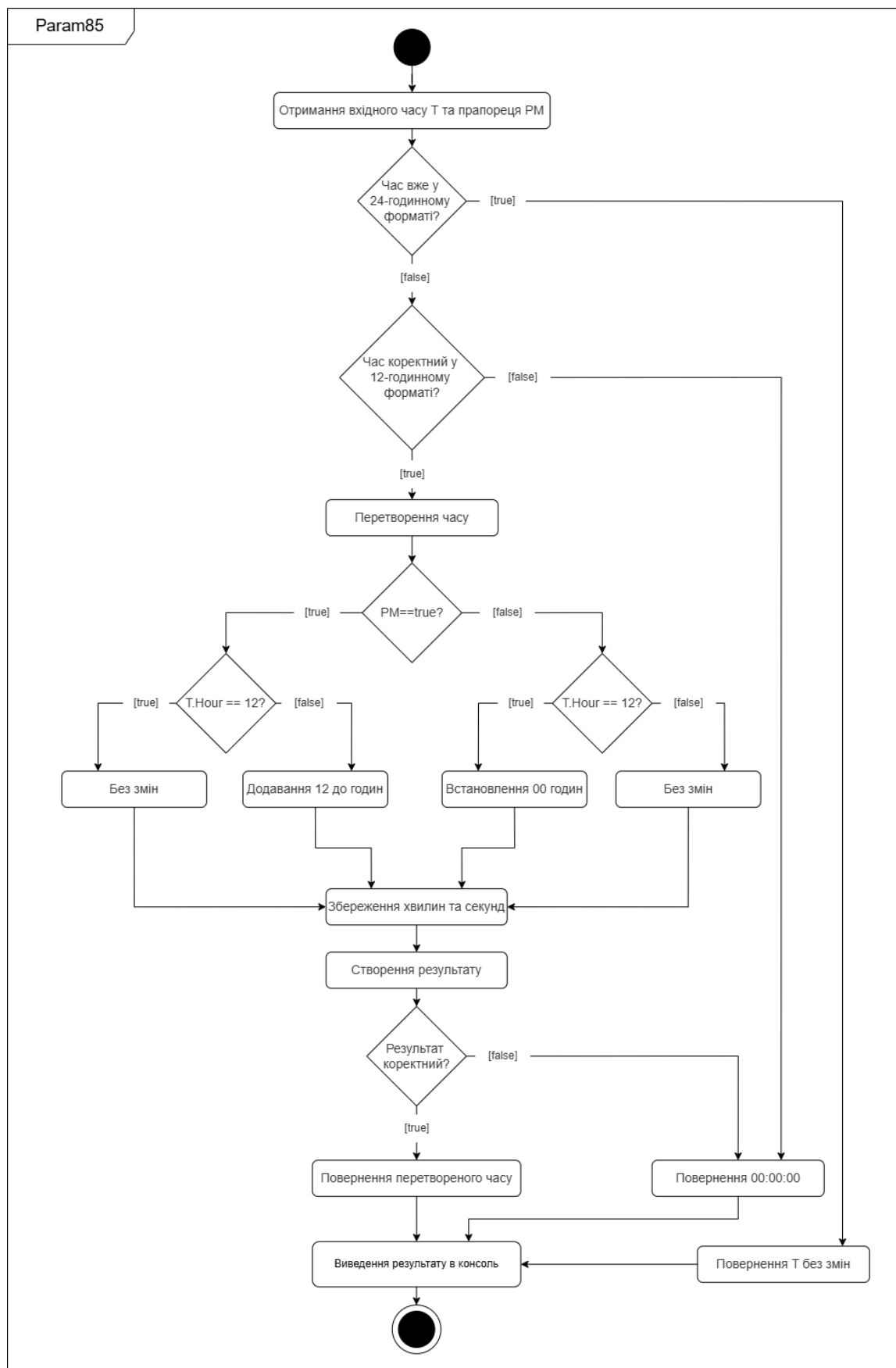


Рисунок Б.1 — Діаграма активності першого завдання.



На рис.Б.2 показано діаграму активності для другого завдання.

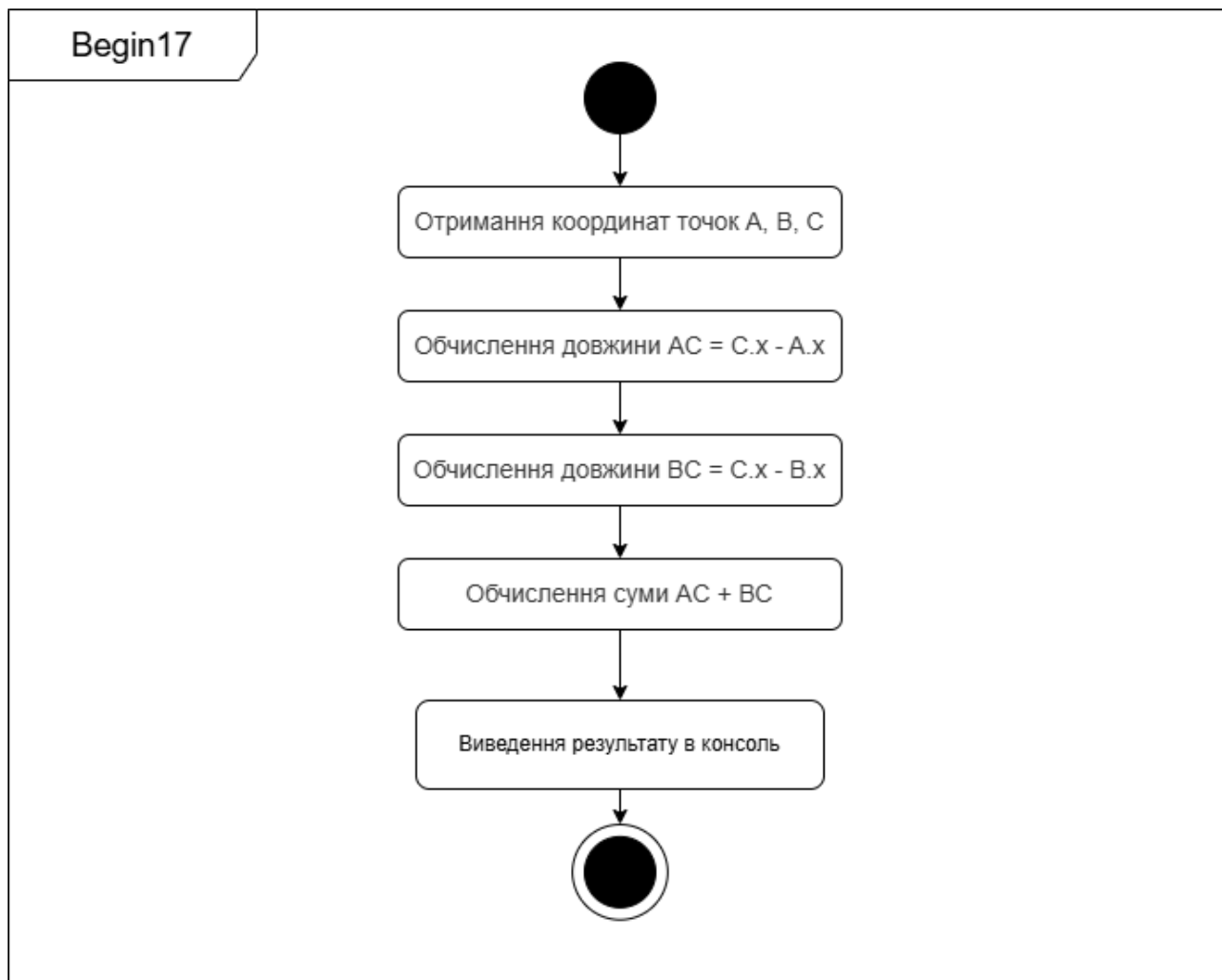


Рисунок Б.2 – Діаграма активності другого завдання.

На рис.Б.3 показано діаграму активності для третього завдання.

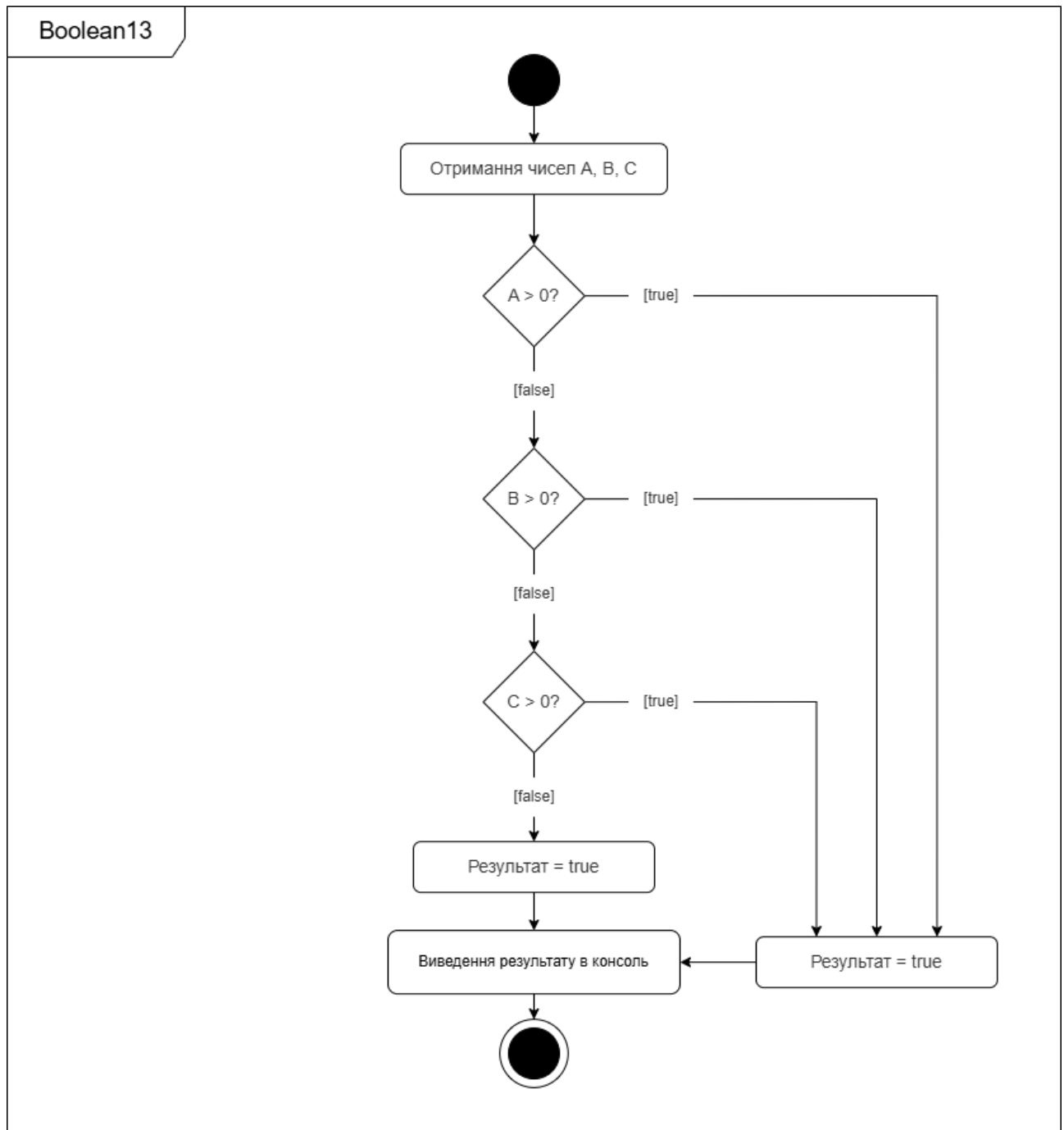


Рисунок Б.3 – Діаграма активності третього завдання.

На рис.Б.4 показано екран роботи програми для першого завдання.

```
=== Завдання 1 (Param85) ===  
Тест 1: 3:45:30 AM → 03:45:30  
Вхідний час: 03:45:30 (AM)  
Результат: 03:45:30  
  
Тест 2: 12:00:00 PM → 12:00:00  
Вхідний час: 12:00:00 (PM)  
Результат: 12:00:00  
  
Тест 3: 8:15:00 AM → 08:15:00  
Вхідний час: 08:15:00 (AM)  
Результат: 08:15:00  
  
Тест 4: 6:30:45 PM → 18:30:45  
Вхідний час: 06:30:45 (PM)  
Результат: 18:30:45  
  
Тест 5: 11:59:59 PM → 23:59:59  
Вхідний час: 11:59:59 (PM)  
Результат: 23:59:59  
  
Тест 6: 33:00:00 AM → 00:00:00 (некоректний)  
Вхідний час: 33:00:00 (AM)  
Результат: 00:00:00  
  
Тест 7: 5:60:00 AM → 00:00:00 (некоректний)  
Вхідний час: 05:60:00 (AM)  
Результат: 00:00:00  
  
Тест 8: 7:00:60 AM → 00:00:00 (некоректний)  
Вхідний час: 07:00:60 (AM)  
Результат: 00:00:00  
  
Тест 9: 16:30:00 → 16:30:00 (вже 24-годинний)  
Вхідний час: 16:30:00 (AM)  
Результат: 16:30:00  
  
Тест 10: 00:00:00 → 00:00:00 (вже 24-годинний)  
Вхідний час: 00:00:00 (AM)  
Результат: 00:00:00
```

Рисунок Б.4 – Екран роботи програми для першого завдання.

На рис.Б.5 показано екран роботи програми для другого та третього завдання.

```
=== Завдання 2 (Begin17) ===  
Введіть координату точки А: 1.1  
Введіть координату точки В: 2.2  
Введіть координату точки С: 8.5  
Довжина відрізка АС: 7.4  
Довжина відрізка ВС: 6.3  
Сума довжин АС і ВС: 13.7  
  
=== Завдання 3 (Boolean13) ===  
Введіть число А: -2  
Введіть число В: 4  
Введіть число С: -6  
Хоча б одне з чисел А, В, С позитивне: так
```

Рисунок Б.5 – Екран роботи програми для другого та третього завдання.