

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.9).

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.10).

### 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.11).

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

Екран роботи програми для першого завдання показаний в додатку Б, рис. Б.4(стор.12).

Екран роботи програми для другого та третього завдання показаний в додатку Б, рис. Б.5(стор.13).

## 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)

    // Перевірка годин: мають бути від 0 до 23
    if (T.Hour < 0 || T.Hour > 23) {
        errors += 100; // Додаємо 100 до коду помилки (помилка в годинах)
    }

    // Перевірка хвилин: мають бути від 0 до 59
    if (T.Min < 0 || T.Min > 59) {
        errors += 10; // Додаємо 10 до коду помилки (помилка в хвилинах)
    }

    // Перевірка секунд: мають бути від 0 до 59
    if (T.Sec < 0 || T.Sec > 59) {
        errors += 1; // Додаємо 1 до коду помилки (помилка в секундах)
    }

    return errors; // Повертаємо код помилки (0 означає коректний час)
}

// Функція для перевірки, чи час вже у 24-годинному форматі
bool Is24HourFormat(TTime T) {
    // Час у 24-годинному форматі, якщо години від 0 до 23
    // і хвилини/секунди коректні
    return (T.Hour >= 0 && T.Hour <= 23) &&
        (T.Min >= 0 && T.Min <= 59) &&
        (T.Sec >= 0 && T.Sec <= 59);
}

// Функція для перевірки коректності часу в 12-годинному форматі
bool Is12HourFormatValid(TTime T) {
    // У 12-годинному форматі години мають бути від 1 до 12
    // і хвилини/секунди коректні
    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 (Param85)
TTime To24h(TTime T, bool PM) {
    TTime result = {0, 0, 0}; // Ініціалізуємо результат нулями (00:00:00)

    // Перевіряємо, чи час вже у 24-годинному форматі
    if (Is24HourFormat(T)) {
        return T; // Якщо так, повертаємо без змін
    }
}

```

```

// Перевіряємо коректність 12-годинного формату
if (!Is12HourFormatValid(T)) {
    return result; // Повертаємо 00:00:00 у випадку некоректного часу
}

// Перетворення в 24-годинний формат
if (PM) {
    // Для PM (після полудня)
    if (T.Hour != 12) {
        result.Hour = T.Hour + 12; // Додаємо 12 до годин (крім 12 PM)
    } else {
        result.Hour = 12; // 12 PM залишається 12 у 24-годинному форматі
    }
} else {
    // Для AM (до полудня)
    if (T.Hour == 12) {
        result.Hour = 0; // 12 AM стає 00 у 24-годинному форматі
    } else {
        result.Hour = T.Hour; // Інші години залишаються без змін
    }
}

// Хвилини і секунди залишаються незмінними
result.Min = T.Min;
result.Sec = T.Sec;

// Додаткова перевірка на випадок, якщо результат вийшов некоректним
if (!Is24HourFormat(result)) {
    result = {0, 0, 0}; // Повертаємо 00:00:00 у випадку некоректного
    результату
}

return result;
}

// Функція для виведення часу у зручному форматі (HH:MM:SS)
void PrintTime(TTime T) {
    // Виводимо години з ведучим нулем, якщо потрібно
    cout << (T.Hour < 10 ? "0" : "") << T.Hour << ":";
    // Виводимо хвилини з ведучим нулем, якщо потрібно
    cout << (T.Min < 10 ? "0" : "") << T.Min << ":";
    // Виводимо секунди з ведучим нулем, якщо потрібно
    cout << (T.Sec < 10 ? "0" : "") << T.Sec;
}

// Демонстрація роботи функцій для часу
void DemoTimeFunctions() {
    cout << "=== Завдання 1 (Param85) ===" << endl;

    // Тестові приклади
    TTime testCases[] = {
        {3, 45, 30}, // 3:45:30 AM - коректний 12-годинний формат
        {12, 0, 0}, // 12:00:00 PM або AM - особливий випадок
        {8, 15, 0}, // 8:15:00 AM - коректний 12-годинний формат
        {6, 30, 45}, // 6:30:45 PM - коректний 12-годинний формат
        {11, 59, 59}, // 11:59:59 PM - коректний 12-годинний формат
        {13, 0, 0}, // Некоректний час (години > 12)
        {5, 60, 0}, // Некоректний час (хвилини > 59)
        {7, 0, 60}, // Некоректний час (секунди > 59)
        {16, 30, 0}, // 16:30:00 - вже у 24-годинному форматі
        {0, 0, 0} // 00:00:00 - вже у 24-годинному форматі
    };

    bool pmFlags[] = {
        false, // 3:45:30 AM
        true, // 12:00:00 PM
    };
}

```

```

        false, // 8:15:00 AM
        true,  // 6:30:45 PM
        true,  // 11:59:59 PM
        false, // Некоректний час
        false, // Некоректний час
        false, // Некоректний час
        false, // Вже у 24-годинному форматі
        false  // Вже у 24-годинному форматі
    };

    for (int i = 0; i < 10; i++) {
        cout << "Тестовий випадок " << i+1 << ":" << endl;
        cout << "Початковий час: ";
        PrintTime(testCases[i]);
        cout << " (" << (pmFlags[i] ? "PM" : "AM") << ")" << endl;

        // Перевірка, чи час вже у 24-годинному форматі
        if (Is24HourFormat(testCases[i])) {
            cout << "Час вже у 24-годинному форматі." << endl;
        }

        // Перевірка коректності 12-годинного формату
        if (!Is12HourFormatValid(testCases[i])) {
            cout << "Увага: час не відповідає 12-годинному формату!" << endl;
        }

        // Перетворення часу
        TTime converted = To24h(testCases[i], pmFlags[i]);

        cout << "У 24-годинному форматі: ";
        PrintTime(converted);
        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 A, B, C 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.

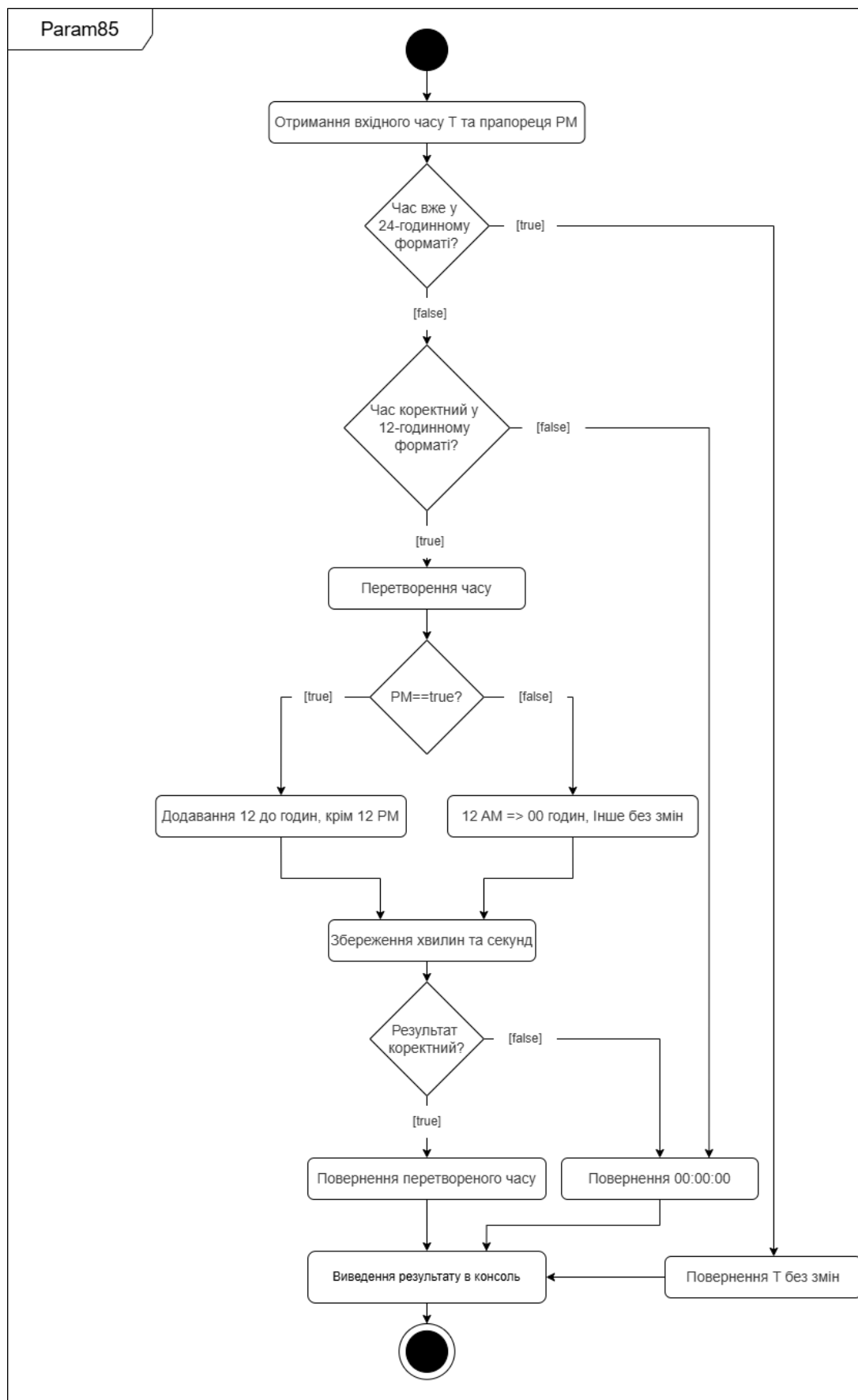


Рисунок Б.1 — Діаграма активності першого завдання.

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

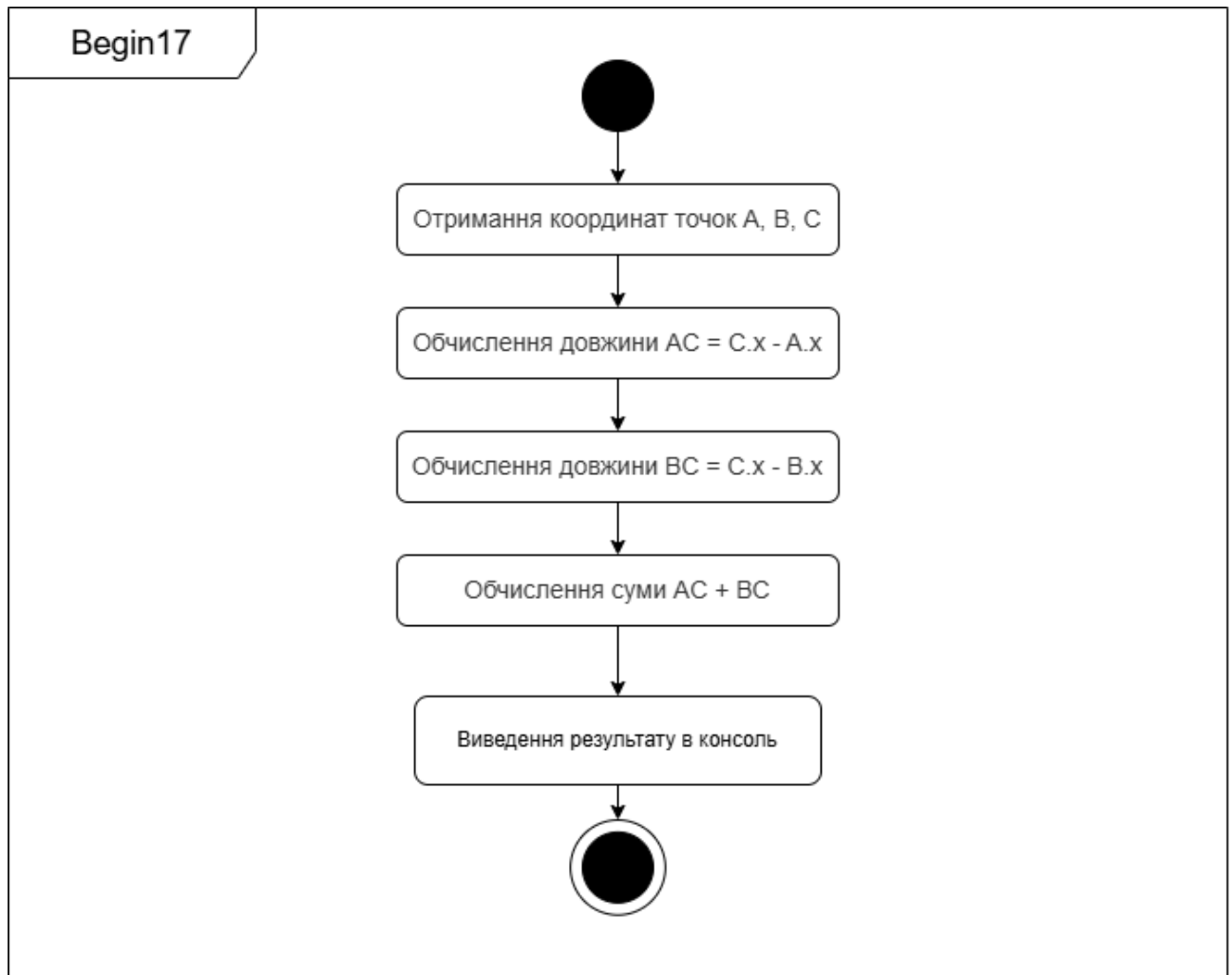


Рисунок Б.2 – Діаграма активності другого завдання.

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

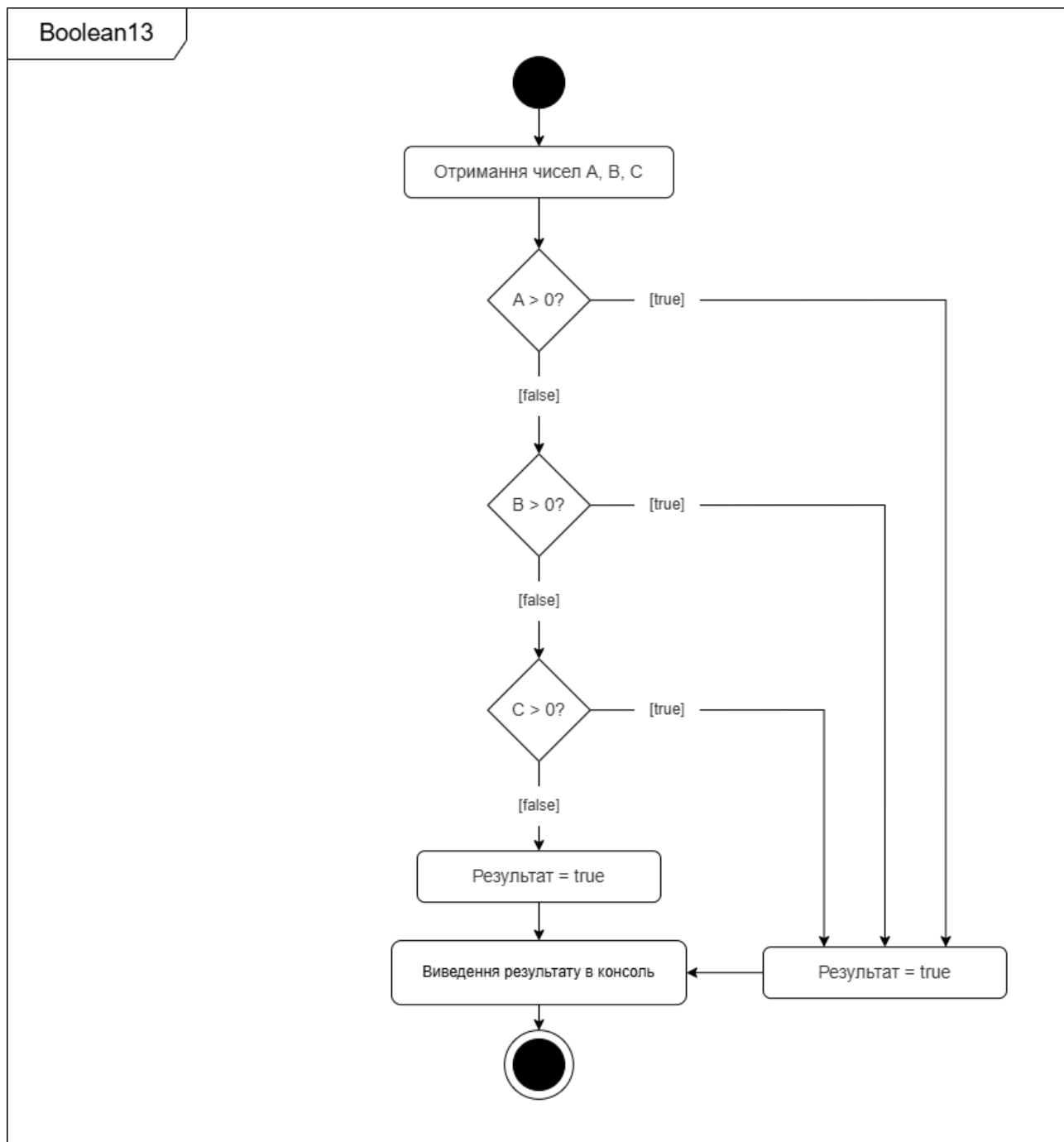


Рисунок Б.3 – Діаграма активності третього завдання.

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

```

=== Завдання 1 (Param85) ===
Тестовий випадок 1:
Початковий час: 03:45:30 (AM)
Час вже у 24-годинному форматі.
У 24-годинному форматі: 03:45:30

Тестовий випадок 2:
Початковий час: 12:00:00 (PM)
Час вже у 24-годинному форматі.
У 24-годинному форматі: 12:00:00

Тестовий випадок 3:
Початковий час: 08:15:00 (AM)
Час вже у 24-годинному форматі.
У 24-годинному форматі: 08:15:00

Тестовий випадок 4:
Початковий час: 06:30:45 (PM)
Час вже у 24-годинному форматі.
У 24-годинному форматі: 06:30:45

Тестовий випадок 5:
Початковий час: 11:59:59 (PM)
Час вже у 24-годинному форматі.
У 24-годинному форматі: 11:59:59

Тестовий випадок 6:
Початковий час: 13:00:00 (AM)
Час вже у 24-годинному форматі.
Увага: час не відповідає 12-годинному формату!
У 24-годинному форматі: 13:00:00

Тестовий випадок 7:
Початковий час: 05:60:00 (AM)
Увага: час не відповідає 12-годинному формату!
У 24-годинному форматі: 00:00:00

Тестовий випадок 8:
Початковий час: 07:00:60 (AM)
Увага: час не відповідає 12-годинному формату!
У 24-годинному форматі: 00:00:00

Тестовий випадок 9:
Початковий час: 16:30:00 (AM)
Час вже у 24-годинному форматі.
Увага: час не відповідає 12-годинному формату!
У 24-годинному форматі: 16:30:00

Тестовий випадок 10:
Початковий час: 00:00:00 (AM)
Час вже у 24-годинному форматі.
Увага: час не відповідає 12-годинному формату!
У 24-годинному форматі: 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: 4.3  
Довжина відрізка АС: 3.2  
Довжина відрізка ВС: 2.1  
Сума довжин АС і ВС: 5.3  
  
=== Завдання 3 (Boolean13) ===  
Введіть число А: -1  
Введіть число В: 3  
Введіть число С: -4  
Хоча б одне з чисел А, В, С позитивне: так
```

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