

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»
на тему "Створення і обробка структур даних мовою C ++"

XAI.301. 174. 319. 21 ЛР

Виконав студент гр. 319

_____ Сисоєв Володимир Сергійович
(підпис, дата) (П.І.Б.)

Перевірів

_____ к.т.н., доц. Олена ГАВРИЛЕНКО
(підпис, дата) (П.І.Б.)

2023

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.

Param 84. Використовуючи тип TTime, описати функцію SubHour (T, N) типу TTime з двома входними параметрами типу TTime і цілого, яка змінює час T на -N годин (якщо час T є неправильним, то воно повертається без змін) . За допомогою функції SubHour вивести новий час для п'яти заданих моментів часу.

Завдання 2.

A. Описати структуру, яка містить всі входні 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 входних даних структури

D. Викликати функції (*методи) з пунктів C, B після оголошення змінної (об'єкту) структури.

E. Вивести значення полів вихідних даних.

Begin 16. Знайти відстань між двома точками із заданими координатами x_1 і x_2 на числової осі: $|x_2 - x_1|$.

Boolean 2. Дано ціле число A. Перевірити істинність висловлювання: «Число A є непарних».

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Param 84

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

- 1) **Hour** – години , int , від 0 - 23
- 2) **Min** – хвилини , int , від 0 - 59
- 3) **Sec** – секунди , int , від 0 – 59
- 4) **N** – час який віднімається , int

Вихідні дані (ім'я, опис, тип):

- 1) **newTime.Hour** – години після віднімання , int
- 2) **newTime.Min** – хвилини після віднімання , int
- 3) **newTime.Sec** – секунди після віднімання , int

Алгоритм вирішення показано на рис. 1

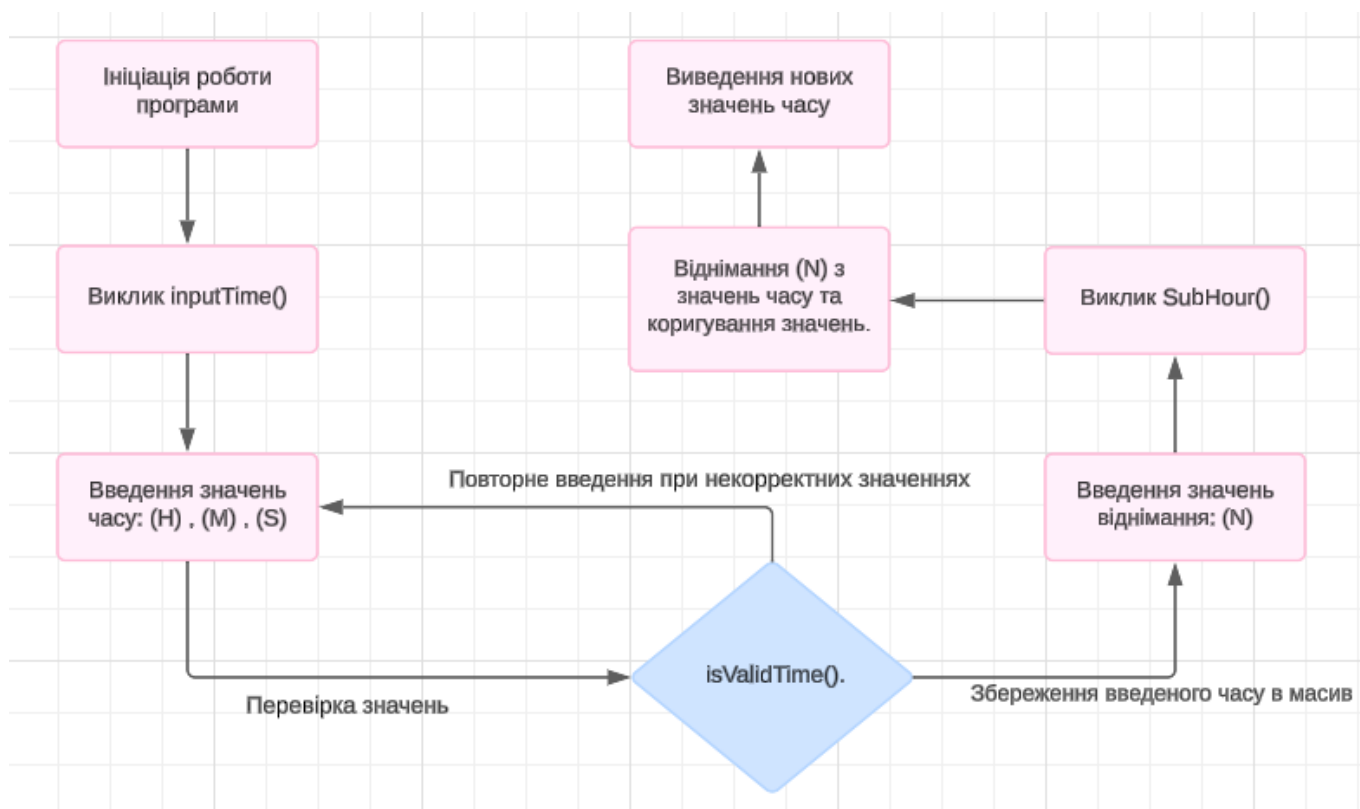


Рисунок 1 – Param 84

Лістинг коду вирішення задачі Param 84 наведено в дод. А (стор. 7).

Екран роботи програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі Begin 16

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

1) **X1** – координати точки у одновимірному просторі , int

2) **X2** – координати точки у одновимірному просторі , int

Вихідні дані (ім'я, опис, тип):

1) **distance** – відстань між двома точками , int

Алгоритм вирішення показано на рис. 2

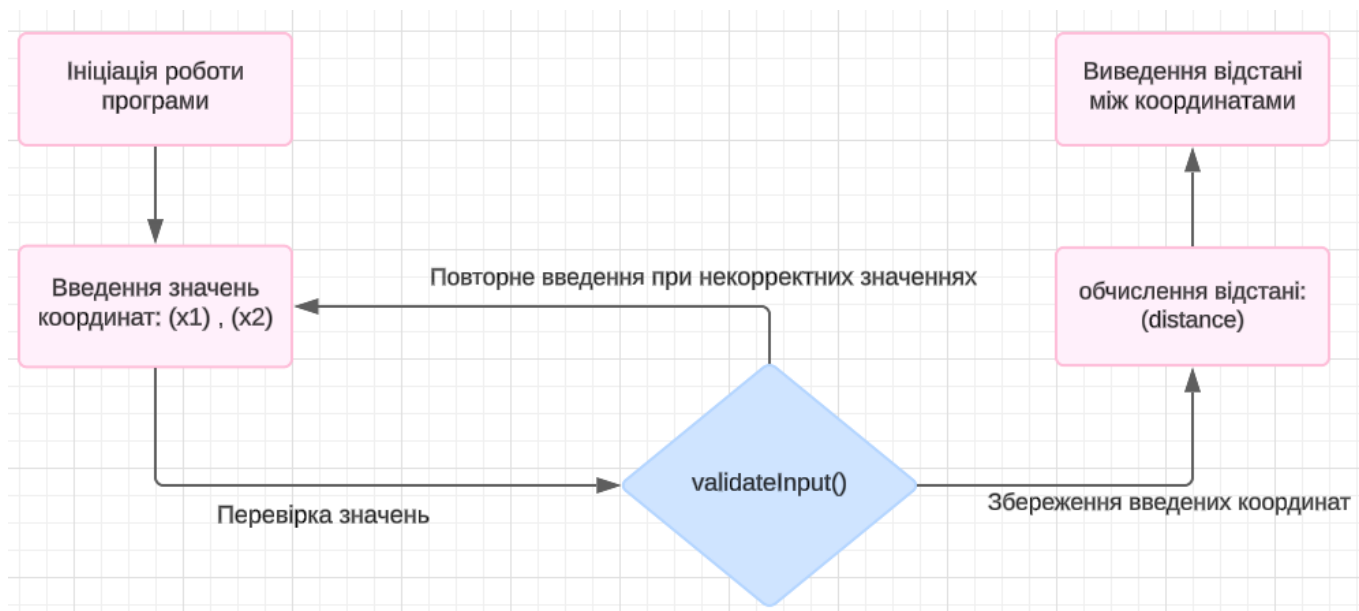


Рисунок 2 – Begin 16

Лістинг коду вирішення задачі Begin 16 наведено в дод. А (стор. 8).

Екран роботи програми показаний на рис. Б.2.

Завдання 3.

Вирішення задачі Boolean 2

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

1) **input** – вхідне число яке користувач вводить з клавіатури , int

Вихідні дані (ім'я, опис, тип):

1) **isOdd** – значення яке вказує чи є введене число непарним , bool

Алгоритм вирішення показано на рис. 3

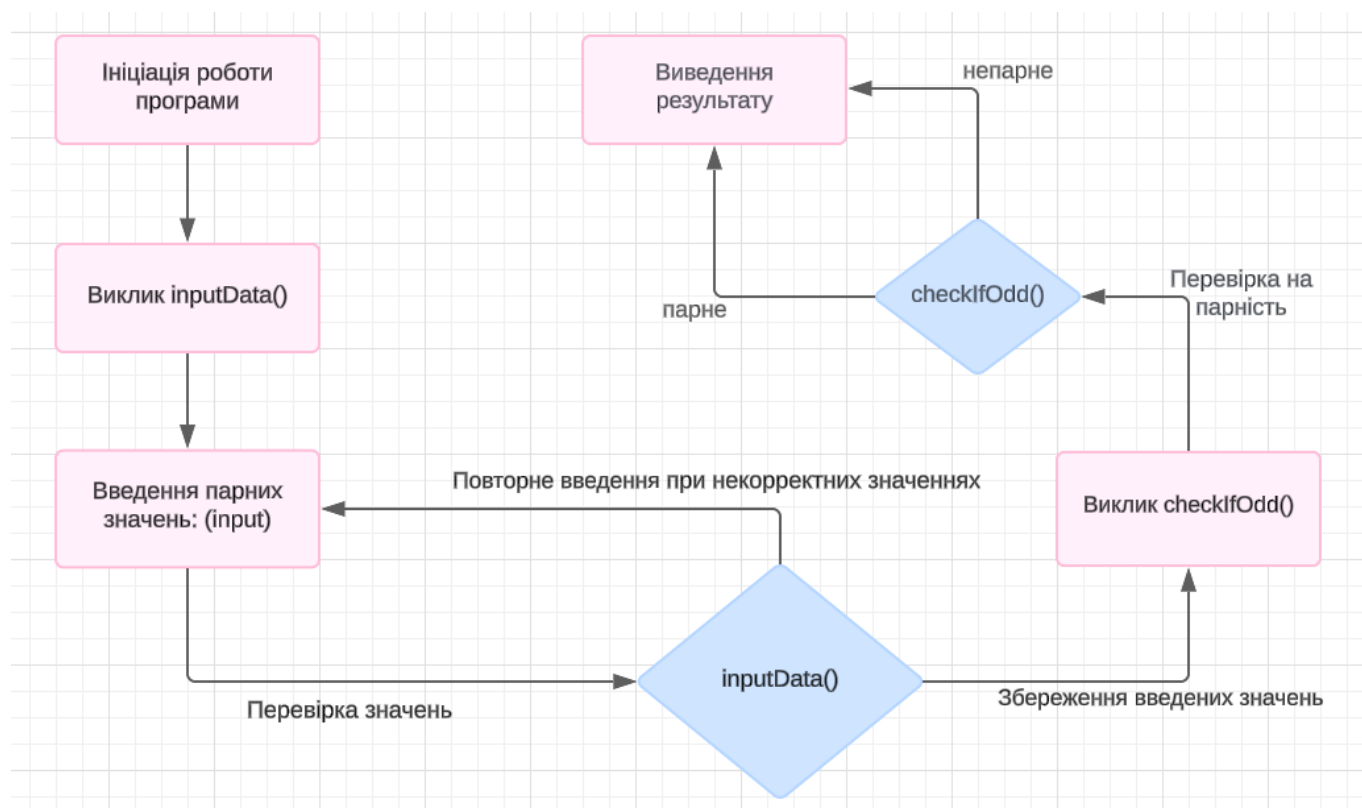


Рисунок 3 – Boolean 2

Лістинг коду вирішення задачі Boolean 2 наведено в дод. А (стор. 9).

Екран роботи програми показаний на рис. Б.3.

ВИСНОВКИ

Було вивчено теоретичний матеріал з основ представлення структур (записів) мовою C ++, а також їх передачі в функції, реалізація декларування і обробки структур мовою C ++ в середовищі Visual Studio

ДОДАТОК А

Лістинг коду програми

Param 84

```

struct TTime {
    int Hour;
    int Min;
    int Sec;
};
// Function for value check
bool isValidTime(TTime t) {
    return (t.Hour >= 0 && t.Hour < 24 && t.Min >= 0 && t.Min < 60 && t.Sec >= 0 &&
t.Sec < 60); // value settings
}
// Function for entering values
TTime inputTime() {
    TTime t;
    cout << "Enter value (hour minutes seconds): " << endl;
    cin >> t.Hour >> t.Min >> t.Sec;

    // value check
    while (!isValidTime(t)) {
        cout << "Incorrect value: " << endl;
        cin >> t.Hour >> t.Min >> t.Sec;
    }
    return t;
}
// function to change time
TTime SubHour(TTime t, int n) {
    if (!isValidTime(t)) {
        return t; // return without changes if value incorrect
    }
    t.Hour -= n;
    // configure value of time if it goes out of 0 - 23
    while (t.Hour < 0) {
        t.Hour += 24;
    }
    while (t.Hour >= 24) {
        t.Hour -= 24;
    }

    return t;
}
//output function
void param_84() {
    TTime times[5];
    int n;
    for (int i = 0; i < 5; ++i) {
        cout << "Enter " << i + 1 << " moment of time:" << endl;
        times[i] = inputTime();
    }
    cout << "Enter the time value for subtracting: " << endl;
    cin >> n;
    cout << "New moment of time:" << endl;
    for (int i = 0; i < 5; ++i) {
        TTime newTime = SubHour(times[i], n);
        cout << "New time " << i + 1 << ": " << newTime.Hour << ":" << newTime.Min <<
":" << newTime.Sec << endl;
    }
}

```

Begin 16

```

struct Point {
    int x1;
    int x2;
    int distance;
};

// Function for check correct values
bool validateInput(int& x) {
    cin >> x;
    if (cin.fail()) {
        cin.clear(); // restore flow state
        cin.ignore(numeric_limits<streamsize>::max(), '\n'); // ignore incorrect
entrence
        return false; // entering values incorrect
    }
    return true;
}

// Function for calculate distance between coordinates
void calculateDistance(Point& p) {
    p.distance = abs(p.x2 - p.x1);
}

// function for enter coordinate and correct check
void inputCoordinates(Point& p) {
    cout << "Enter coordinate x1: ";
    while (!validateInput(p.x1)) {
        cout << "Invalid input. Enter an integer for coordinate x1: ";
    }

    cout << "Enter coordinate x2: ";
    while (!validateInput(p.x2)) {
        cout << "Invalid input. Enter an integer for coordinate x2: ";
    }
}

void begin_16() {
    Point p;

    inputCoordinates(p);
    calculateDistance(p);

    // output results
    cout << "Distance between points: " << p.distance << endl;
}

```


Boolean 2

```

struct Data {
    int input;
    bool isOdd;
};

void checkIfOdd(Data& data) {
    data.isOdd = (data.input % 2 != 0);
}

bool inputData(Data& data) {
    cout << "enter the whole number : ";
    if (cin >> data.input) {
        return true; // if entering data is correct
    }
    else {
        cin.clear(); // restore flow state
        cin.ignore(numeric_limits<streamsize>::max(), '\n'); // ignore incorrect
entrance
        return false; // entering values incorrect
    }
}

void boolean_2() {
    Data data;
    if (inputData(data)) {
        checkIfOdd(data);
        cout << "value " << data.input << (data.isOdd ? " paired." : " unpaired.") <<
endl;
    }
    else {
        cout << "entered incorrect value." << endl;
    }
}

```

ДОДАТОК Б
Скрін-шоти вікна виконання програми

```
Choose task:
1 - Param84
2 - Begin16
3 - Boolean2
4 - Exit

1
Enter 1 moment of time:
Enter value (hour minutes seconds):
09 24 16
Enter 2 moment of time:
Enter value (hour minutes seconds):
12 36 28
Enter 3 moment of time:
Enter value (hour minutes seconds):
17 01 31
Enter 4 moment of time:
Enter value (hour minutes seconds):
22 05 16
Enter 5 moment of time:
Enter value (hour minutes seconds):
03 52 01
Enter the time value for subtracting:
3
New moment of time:
New time 1: 6:24:16
New time 2: 9:36:28
New time 3: 14:1:31
New time 4: 19:5:16
New time 5: 0:52:1

Choose task:
1 - Param84
2 - Begin16
3 - Boolean2
4 - Exit
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання
Param 84

```
Choose task:
1 - Param84
2 - Begin16
3 - Boolean2
4 - Exit

2
Enter coordinate x1: 12
Enter coordinate x2: 6
Distance between points: 6

Choose task:
1 - Param84
2 - Begin16
3 - Boolean2
4 - Exit
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Begin 16

```
Choose task:
1 - Param84
2 - Begin16
3 - Boolean2
4 - Exit

3
enter the whole number : 10
value 10 unpaired.

Choose task:
1 - Param84
2 - Begin16
3 - Boolean2
4 - Exit
```

Рисунок Б.3 – Екран виконання програми для вирішення завдання
Boolean 2