

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. 175. 318. 08 ЛР

Виконав студент гр. _____ 318

_____ Каріна ГЛІБОВА
(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 завдань представлено в табл.1.

Param 84. Використовуючи тип TTime (див. Param71), описати функцію SubHour (T, N) типу TTime з двома вхідними 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 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 структури.

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

E. Вивести значення полів вихідних даних.

Begin 14. Дана довжина L окружності. Знайти її радіус R і площу S круга, обмеженого цим колом, враховуючи, що $L = 2 \cdot \pi \cdot R$, $S = \pi \cdot R^2$. Як значення π вважати рівним 3.14.

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

Завдання 3.

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

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Param 84.

Функція SubHour.

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

T.Hour – число годин, ціле, дійсне значення.

T.Min – число хвилин, ціле, дійсне значення.

T.Sec – число секунд, ціле, дійсне значення.

N – задана кількість годин для віднімання, ціле, дійсне значення.

const int SIZE = 5 – кількість моментів часу, ціле, константа.

Вихідні дані (ім'я, опис, тип):

newTime – новий час після віднімання годин, ціле, дійсне значення.

Лістинг коду вирішення задачі Param 84 наведено в дод. А (стор. 6-10).

Екран роботи програми показаний на рис. Б.1. (дод. Б, стор. 11-13)

Приклад діаграми для завдання Param 84 наведено на рис. Б.4. (дод. Б, стор. 11-13)

Завдання 2.

Вирішення задачі Begin 14.

Виведення радіуса і площі.

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

L – довжина кола, число з плаваючою крапкою;

Вихідні дані (ім'я, опис, тип):

R – радіус, число з плаваючою крапкою;

S – площа круга, число з плаваючою крапкою;

Лістинг коду вирішення задачі Begin 14 наведено в дод. А (стор. 6-10).

Екран роботи програми показаний на рис. Б.2. (дод. Б, стор. 11-13)

Приклад діаграми для завдання Begin 14 наведено на рис. Б.5. (дод. Б, стор. 11-13)

Вирішення задачі Boolean 2.

Перевірка на непарність числа.

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

A – ціле число, дійсний тип.

Вихідні дані (ім'я, опис, тип):

isOdd – перевірка «парне» чи «непарне» число, дійсний тип із подвійною точністю.

Лістинг коду вирішення завдання Boolean 2 наведено в дод. А (стор. 6-10)

Екран роботи програми показаний на рис. Б.3. (дод. Б, стор. 11-13)

Приклад діаграми для завдання Boolean 2 наведено на рис. Б.6. (дод. Б, стор. 11-13)

ВИСНОВКИ

Було вивчено теоретичний матеріал з основ представлення структур (записів) мовою C ++ та опрацьована їх передача в функції. На практиці було реалізовано декларування і обробку структур мовою C ++ в середовищі Visual Studio.

ДОДАТОК А

Лістинг коду програми

```

#include <iostream> // підключення бібліотеки для роботи з введенням/виведенням
#include <Windows.h> // підключення бібліотеки для налаштування кодування в консолі
#include "task1.h" // заголовковий файл для першого завдання
#include "task2.h" // заголовковий файл для другого і третього завдань
using namespace std;

int main() {
    SetConsoleOutputCP(1251);
    int menu;

    do { // початок циклу
        // Виведення меню з варіантами завдань
        cout << "Оберіть завдання:\n\t1 - Функція SubHour.\n\t2 - Визначення радіуса
кола та площі круга.\n\t3 - Перевірити непарність введенного числа." << endl;
        cin >> menu; // зчитування вибору користувача

        switch (menu) { // обробка вибору
            case 1:
                cout << "\n=== 1: функція SubHour ===\n"; // вивід заголовку і запуск
функції з task1.h для обчислення радіуса і площі круга
                RunTimeTask(); // виклик функції для першого завдання
                break;
            case 2:
                cout << "\n=== 2: РАДІУС І ПЛОЩА ===\n"; // вивід заголовку і запуск
функції з task2.h
                Circle_function(); // виклик функції для другого завдання
                break;
            case 3:
                cout << "\n=== 3: ПЕРЕВІРКА неПАРНОСТІ ===\n"; // вивід заголовку і
запуск функції для перевірки непарності числа
                Numbercheck_function(); // виклик функції для третього завдання
                break;
            case -1: // вихід з програми
                cout << "Вихід..." << endl;
                break;
            default:
                cout << "\n!!!Обирати можна тільки з наведених вище чисел!!!\n"; //
повідомлення про помилковий вибір
        }

        cout << endl;
        cout << "+-----+" << endl; // строка задля полегшення
візуального сприймання тексту
        cout << endl;
    } while (menu != -1); // повторювати меню, поки користувач не введе -1
}

#include <iostream> // підключення стандартної бібліотеки для вводу/виводу
using namespace std;

struct TTime { // структура TTime
    int Hour; // години (0-23)
    int Min; // хвилини (0-59)
    int Sec; // секунди (0-59)
};

int CheckTime(TTime T); // перевірка правильності часу

```

```

TTime SubHour(TTime T, int N); // зменшення на N годин

void PrintTime(TTime T); // виведення часу

void RunTimeTask(); // вхід у завдання

#include <iostream> // підключення бібліотеки для введення/виведення
#include "task1.h" // підключення заголовкового файлу для першого завдання
using namespace std;

// функція перевірки правильності часу
int CheckTime(TTime T) {
    // перевірка, чи значення годин, хвилин і секунд у межах допустимого діапазону
    if (T.Hour >= 0 && T.Hour < 24 &&
        T.Min >= 0 && T.Min < 60 &&
        T.Sec >= 0 && T.Sec < 60)
    {
        return 1; // якщо всі значення коректні, повернення 1
    }
    return 0; // якщо ні, повернення 0
}

// функція віднімання годин
TTime SubHour(TTime T, int N) {
    T.Hour -= N; // віднімання заданої кількості годин
    while (T.Hour < 0) { // поки значення більше нуля
        // якщо результат менше 0 – додавання 24, щоб перейти в попередній день
        T.Hour += 24;
    }

    return T; // повернення нового моменту часу
}

// функція виводу часу
void PrintTime(TTime T) {
    // Виведення часу з доданими нулями, якщо потрібно
    cout << (T.Hour < 10 ? "0" : "") << T.Hour << ":"
        << (T.Min < 10 ? "0" : "") << T.Min << ":"
        << (T.Sec < 10 ? "0" : "") << T.Sec << endl;
}

bool InputInt(const string& prompt, int& result) { // функція введення цілого числа
    cout << prompt;
    cin >> result;
    // перевірка на помилку вводу (неціле число)
    if (cin.fail()) {
        cin.clear(); // скидання стану помилки
        cin.ignore(1000, '\n'); // очищення буфера вводу
        cout << "Помилка: введено не ціле число." << endl; // виведення сповіщення про
        // помилку
        return false;
    }
    return true;
}

TTime InputTime(int index) { // функція для введення одного моменту часу
    TTime t;
    cout << "Введення моменту часу #" << index + 1 << ":\n"; // введення часу

    while (!InputInt("  Години: ", t.Hour)); // поки
    while (!InputInt("  Хвилини: ", t.Min)); // вводиться

```

```

    while (!InputInt(" Секунди: ", t.Sec)); // час
    return t; // виводяться ці числа
}

void RunTimeTask() { // головна функція виконання завдання з обробки часу
    const int SIZE = 5; // кількість моментів часу
    TTime times[SIZE]; // масив з 5 моментів часу
    int choice;

    cout << "Виберіть спосіб введення часу:\n"; // запит користувача, як він хоче
    вводити час
    cout << "1 - Ввести час вручну\n";
    cout << "2 - Використати вбудовані значення\n";
    cout << "Ваш вибір: ";
    cin >> choice;

    if (choice == 1) { // введення часу вручну
        for (int i = 0; i < SIZE; i++) {
            times[i] = InputTime(i); // введення кожного моменту часу
        }
    }
    else {
        // Вбудовані значення
        times[0] = { 14, 30, 45 };
        times[1] = { 3, 5, 15 };
        times[2] = { 23, 59, 59 };
        times[3] = { 0, 0, 0 };
        times[4] = { 25, 10, 5 }; // неправильний
    }

    int N;
    cout << "Введіть кількість годин для віднімання: "; // введення кількості годин
    для віднімання
    cin >> N;

    for (int i = 0; i < SIZE; i++) { // опрацювання кожного моменту часу
        cout << "\nПочатковий час: \t\t"; // введення початкового часу
        PrintTime(times[i]); // виведення початкового часу

        bool check = true; // перевірка, чи час коректний
        if (!CheckTime(times[i])) { // якщо ні
            cout << "Неправильний час.\nПовертається без змін: \t\t",
            PrintTime(times[i]); // виведення відповідного повідомлення
            check = false;
        }

        TTime newTime = SubHour(times[i], N); // обчислення нового часу (із
        відніманням годин)

        if (check) { // якщо коректний
            cout << "Після зменшення на " << N << " годин: \t", PrintTime(newTime); //
            виведення зміненого результат
        }
    }
}

#include <iostream> // підключення стандартної бібліотеки для вводу/виводу
using namespace std;

const float PI = 3.14; // константа, число пі

```



```

struct Circle { // структура для кола
    float L, R, S; // L – довжина кола, R – радіус, S – площа круга
    void input(); // метод для зчитування довжини кола
    void calculate(); // метод для обчислення радіуса та площі на основі введеної
довжини
    void output(); // метод для виведення результатів обчислень
};

struct NumberCheck { // структура для перевірки числа на непарність
    int A;
    bool isOdd;
    void input(); // метод для зчитування числа
    void calculate(); // метод для перевірки, чи є число непарним
    void output(); // метод для виведення результату перевірки
};

void Circle_function(); // функція запускає обробку задачі з колом
void Numbercheck_function(); // функція запускає обробку задачі з перевіркою числа на
непарність

#include <iostream> // підключення бібліотеки для вводу/виводу
#include "task2.h" // підключення заголовкового файлу для другого і третього завдань
using namespace std;

// реалізація методів Circle
void Circle::input() { // функція введення
    cout << "Введіть довжину кола L: "; // введення довжини кола
    cin >> L; // зчитування довжини кола від користувача
    if (cin.fail() || L <= 0) { // перевірка, чи дійсне число і чи більше нуля
        cin.clear(); // скидає помилку
        cin.ignore(1000, '\n'); // очищає буфер
        // виведення сповіщення про помилку
        cout << "Недійсне введення. Довжина має бути більше за 0. За замовчуванням
встановлено L = 1.\n";
        L = 1; // значення за замовчуванням, якщо введення було некоректним
    }
}

void Circle::calculate() { // функція розрахунку
    R = L / (2 * PI); // обчислення радіуса за формулою:  $R = L / (2\pi)$ 
    S = PI * R * R; // обчислення площі круга:  $S = \pi R^2$ 
}

void Circle::output() { // функція виведення
    cout << "Радіус R = " << R << endl; // виведення радіуса
    cout << "Площа S = " << S << endl; // виведення площі
}

// реалізація методів NumberCheck
void NumberCheck::input() { // функція введення
    cout << "Введіть число A: "; // введення числа A
    cin >> A; // зчитування числа
    cin.ignore(1000, '\n'); // очищає буфер, якщо число не ціле
}

void NumberCheck::calculate() { // функція розрахунку
    isOdd = (A % 2 != 0); // перевірка на парність числа
}

void NumberCheck::output() { // функція виведення
    cout << "Число " << A << (isOdd ? " непарне." : " парне.") << endl; // виведення
результату перевірки
}

```

```
}

void Circle_function() { // функція для обробки задачі з колом
    Circle c;
    c.input(); // введення даних
    c.calculate(); // обчислення параметрів
    c.output(); // виведення результатів
}

void Numbercheck_function() { // функція для обробки задачі з перевіркою числа на
непарність
    NumberCheck num;
    num.input(); // введення даних
    num.calculate(); // обчислення параметрів
    num.output(); // виведення результатів
}
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

```

Оберіть завдання:
    1 - Функція SubHour.
    2 - Визначення радіуса кола та площі круга.
    3 - Перевірити непарність введенного числа.
1

=== 1: функція SubHour ===
Виберіть спосіб введення часу:
1 - Ввести час вручну
2 - Використати вбудовані значення
Ваш вибір: 2
Введіть кількість годин для віднімання: 8

Початковий час:          14:30:45
Після зменшення на 8 годин: 06:30:45

Початковий час:          03:05:15
Після зменшення на 8 годин: 19:05:15

Початковий час:          23:59:59
Після зменшення на 8 годин: 15:59:59

Початковий час:          00:00:00
Після зменшення на 8 годин: 16:00:00

Початковий час:          25:10:05
Неправильний час.
Повертається без змін:    25:10:05

```

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

```

Оберіть завдання:
    1 - Функція SubHour.
    2 - Визначення радіуса кола та площі круга.
    3 - Перевірити непарність введенного числа.
2

=== 2: РАДІУС І ПЛОЩА ===
Введіть довжину кола L: 56
Радіус R = 8.9172
Площа S = 249.682

```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Begin 14

Оберіть завдання:

- 1 – Функція SubHour.
- 2 – Визначення радіуса кола та площі круга.
- 3 – Перевірити непарність введенного числа.

3

=== 3: ПЕРЕВІРКА неПАРНОСТІ ===

Введіть число А: 7

Число 7 непарне.

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

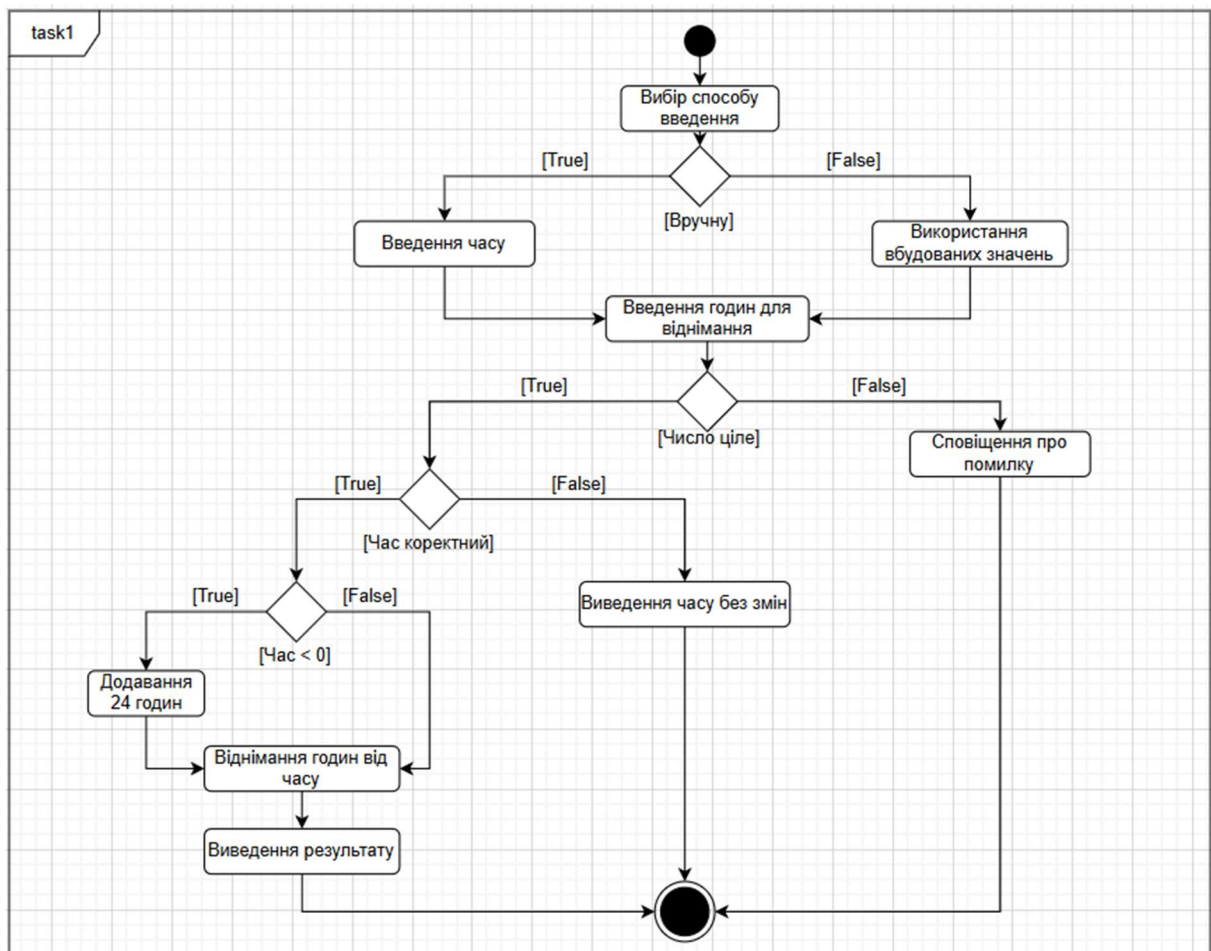


Рисунок Б.4 – Діаграма для завдання Param 84

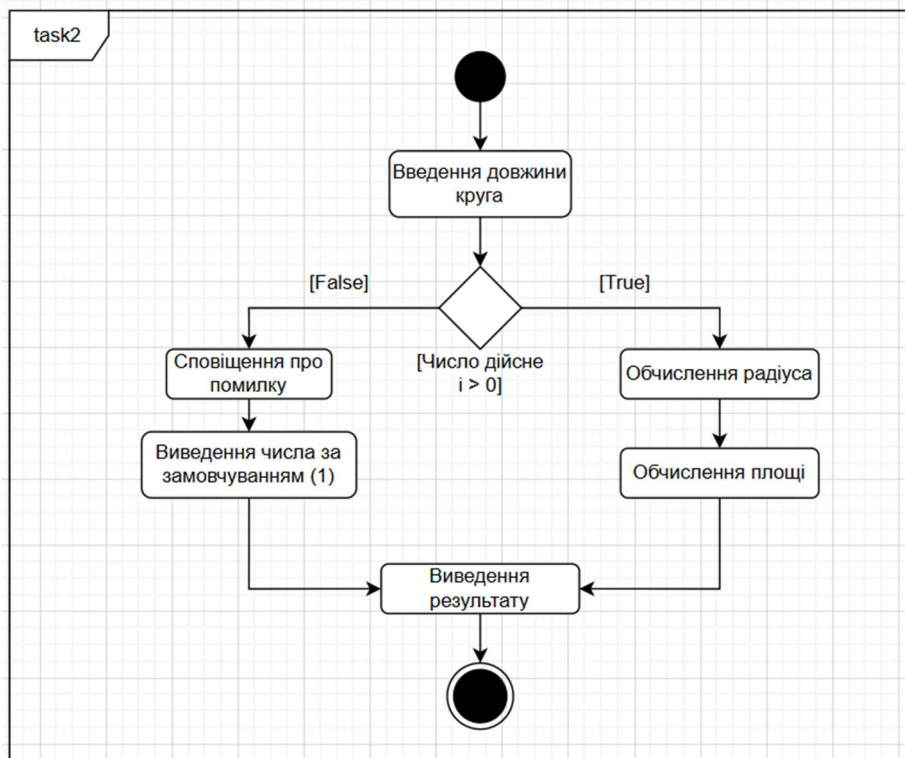


Рисунок Б.5 – Діаграма для завдання Begin 14

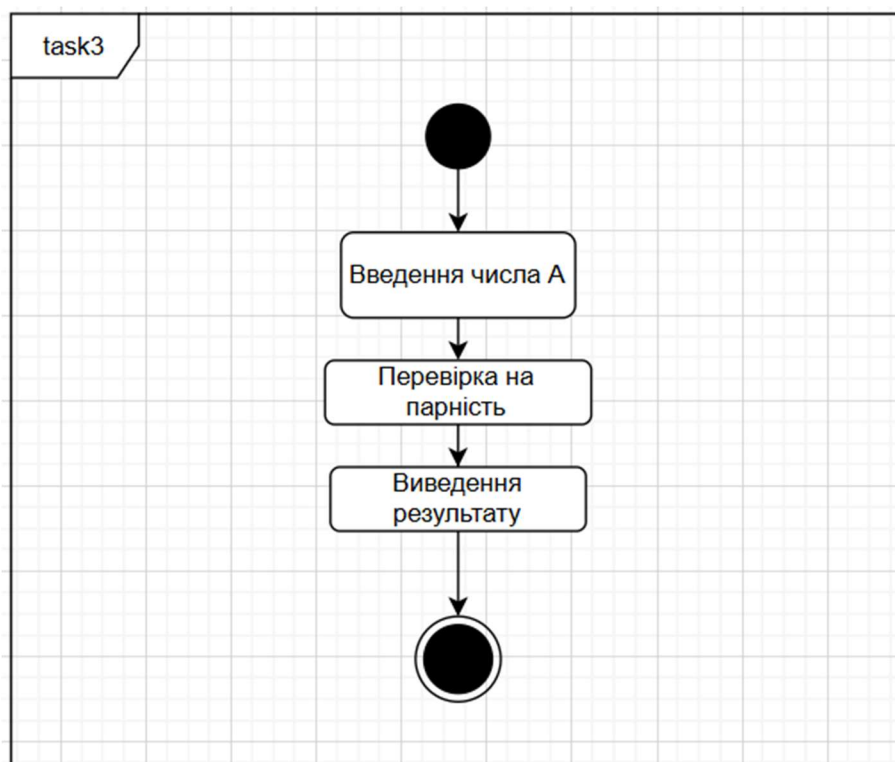


Рисунок Б.6 – Діаграма для завдання Boolean 2