Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №4 з дисципліни
« Основи програмування 2. Модульне програмування»

«Перевантаження операторів»
Варіант 3

Виконав студент ІП-15, Борисик Владислав Тарасович
(шифр, прізвище, ім'я, по батькові)

Перевірила Вєчерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №3
Перевантаження операторів
Варіант 3
Задача

3. Визначити клас "Час" для роботи із часом в межах доби. Членами класу є години, хвилини та секунди. Реалізувати для нього декілька конструкторів, геттери, метод обчислення часу, що залишився до завершення доби. Перевантажити оператори: префіксний "+=" – для збільшення кількості хвилин на вказану величину, "-" – для знаходження тривалості часу між двома моментами часу. Створити три об'єкта часу (T1, T2, T3), використовуючи різні конструктори. Збільшити час T1 на 17 хвилин, а час T2 – на 34 хвилини. Визначити тривалість часу між моментами часу T1 і T2. Для часу T3 визначити час, що залишився до завершення доби.

Код

C++

main.cpp:

```cpp
#include "classes.h"
#include "functions.h"

int main() {
    Time T1(12);
    T1.ShowInfo();

    Time T2(13,40);
    T2.ShowInfo();

    Time T3(21, 43,14);
    T3.ShowInfo();

    test_getters(T3);

    test_t1_increment(T1);

    test_t2_increment(T2);

    test_time_difference(T1, T2);

    test_time_left(T3);
}
```

functions.cpp:

```
#include "functions.h"

void test_getters(Time T3){
    cout << "\nTesting getters (T3):\n";
    printf("%d:%d:%d", T3.GetHours(), T3.GetMinutes(),
T3.GetSeconds());
}

void test_t1_increment(Time &T1){
    printf("\n\nT1 before incrementing:\n");
    T1.ShowInfo();
    printf("T1 after incrementing:\n");
    T1 += 17;
    T1.ShowInfo();
}

void test_t2_increment(Time &T2){
    printf("\nT2 before incrementing:\n");
    T2.ShowInfo();
    printf("T2 after incrementing:\n");
    T2 += 34;
    T2.ShowInfo();
}

void test_time_difference(Time T1, Time T2){
    printf("\nTime differece between T2 and T1:\n");
    Time T4 = T2 - T1;
    T4.ShowInfo();
}

void test_time_left(Time T3){
    printf("\nTime left until the end of the day (T3):\n");
    T3.TimeLeft();
}
```

functions.h:

```
#ifndef INC_2LABWORK_4_FUNCTIONS_H
#define INC_2LABWORK_4_FUNCTIONS_H
#endif

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

void test_getters(Time T3);
void test_t1_increment(Time &T1);
void test_t2_increment(Time &T2);
void test_time_difference(Time T1, Time T2);
void test_time_left(Time T3);
```

classes.cpp:

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

Time::Time(){
    hours = 0;
    minutes = 0;
    seconds = 0;
}

Time::Time(int hours){
    // якщо кількість годин більша за 24
    if(hours >= 24){
        // переводимо години у дні
        int days = hours / 24;
        // віднімаємо від годин кількість днів * 24 (віднімаємо
зайві години)
        hours -= days * 24;

    }

    this->hours = hours;
    minutes = 0;
    seconds = 0;
}

Time::Time(int hours, int minutes){
    // якщо кількість хвилин більша за 60
    if(minutes >= 60){
        // переводимо хвилини у години
        int exceeded_hours = minutes / 60;
        // віднімаємо від хвилин кількість годин * 60 (віднімаємо
зайві хвилини)
        minutes -= exceeded_hours * 60;

        // додаємо до основних годин "зайві" години
        hours += exceeded_hours;
    }

    // якщо кількість годин більша за 24
    if(hours >= 24){
        // переводимо години у дні
        int days = hours / 24;
```

```
        // віднімаємо від годин кількість днів * 24 (віднімаємо
зайві години)
        hours -= days * 24;
    }


    this->hours = hours;
    this->minutes = minutes;
    seconds = 0;
}

Time::Time(int hours, int minutes, int seconds){
    // якщо кількість секунд більша за 60
    if(seconds >= 60){
        // переводимо секунди у хвилини
        int exceeded_minutes = seconds / 60;
        // віднімаємо від секунд кількість хвилин * 60 (віднімаємо
зайві секунди)
        seconds -= exceeded_minutes * 60;

        // додаємо до основних хвилин "зайві" хвилини
        minutes += exceeded_minutes;
    }

    // якщо кількість хвилин більша за 60
    if(minutes >= 60){
        // переводимо хвилини у години
        int exceeded_hours = minutes / 60;
        // віднімаємо від хвилин кількість годин * 60 (віднімаємо
зайві хвилини)
        minutes -= exceeded_hours * 60;

        // додаємо до основних годин "зайві" години
        hours += exceeded_hours;
    }

    // якщо кількість годин більша за 24
    if(hours >= 24){
        // переводимо години у дні
        int days = hours / 24;
        // віднімаємо від годин кількість днів * 24 (віднімаємо
зайві години)
        hours -= days * 24;
    }
```

```
    this->hours = hours;
    this->minutes = minutes;
    this->seconds = seconds;
}

int Time::GetHours(){
    return this->hours;
}

int Time::GetMinutes(){
    return this->minutes;
}

int Time::GetSeconds(){
    return this->seconds;
}

void Time::TimeLeft(){
    // секунди, які залишились
    int seconds_left = 0;
    // хвилини, які залишились
    int minutes_left = 0;
    // години, які залишились
    int hours_left = 0;

    // якщо кількість секунд більше 0
    if(this->GetSeconds() > 0){
        // секунди, які залишились (60 - кількість секунд)
        seconds_left = 60 - this->seconds;
        // хвилини, які залишились (60 - кількість хвилин - 1)
        minutes_left = 60 - this->minutes - 1;
        // години, які залишились (24 - кількість годин - 1)
        hours_left = 24 - this->hours - 1;

    // якщо кількість хвилин більше 0
    } else if(this->GetMinutes() > 0){
        // секунди, які залишились (60 - кількість секунд)
        seconds_left = 60 - this->seconds;
        // хвилини, які залишились (60 - кількість хвилин)
        minutes_left = 60 - this->minutes;
        // години, які залишились (24 - кількість годин - 1)
        hours_left = 24 - this->hours - 1;

    // якщо кількість годин більше 0
```

```
    } else{
        // секунди, які залишились (60 - кількість секунд)
        seconds_left = 60 - this->seconds;
        // хвилини, які залишились (60 - кількість хвилин)
        minutes_left = 60 - this->minutes;
        // години, які залишились (24 - кількість хвилин)
        hours_left = 24 - this->hours;
    }

    printf("Time left until the end of the day: %d hours %d minutes
%d seconds\n", hours_left, minutes_left, seconds_left);
}

void Time::ShowInfo(){
    printf("Hours: %d, minutes: %d, seconds: %d\n", this-
>GetHours(), this->GetMinutes(), this->GetSeconds());
}

void Time::operator+=(int minutes_to_add) {
    // об'єкт, до якого потрібно додати хвилини
    Time *main_object = this;
    // нове значення хвилин (хвилини об'єкту + кількість хвилин,
яку потрібно додати)
    int new_minutes_value = main_object->GetMinutes() +
minutes_to_add;

    // години з об'єкту
    int main_hours = main_object->GetHours();
    // нове значення хвилин
    int main_minutes = new_minutes_value;
    // секунди з об'єкту
    int main_seconds = main_object->GetSeconds();

    // створюємо тимчасовий об'єкт, на випкадок, якщо потрібно
перевести "зайві хвилини у години"
    // наприклад, якщо нове значення хвилин буде 74, то в
temp_object воно конвертується в 1 годину 14 хвилин
    Time temp_object(main_hours, main_minutes, main_seconds);

    // присвоюємо основному об'єкту години з тимчасового об'єкту
    main_object->hours = temp_object.hours;
    // присвоюємо основному об'єкту хвилини з тимчасового об'єкту
    main_object->minutes = temp_object.minutes;
}
```

```
Time Time::operator-(Time right_value) {
    // операнд зліва від оператору -
    Time *left_value = this;

    // конвертуємо весь час лівого операнду в секунди (години *
3600 + хвилини * 60 + секунди)
    int left_value_seconds = left_value->GetHours() * 3600 +
left_value->GetMinutes() * 60 + left_value->GetSeconds();
    // конвертуємо весь час правого операнду в секунди (години *
3600 + хвилини * 60 + секунди)
    int right_value_seconds = right_value.GetHours() * 3600 +
right_value.GetMinutes() * 60 + right_value.GetSeconds();

    // якщо кількість секунд правого операнду більша за кількість
секунд лівого операнду
    if(right_value_seconds > left_value_seconds){
        // виводимо в консоль повідомлення з помилкою
        cout << "Error: right-hand side operand is greater that
left-hand side operand. Returning an empty object\n";

        // створюємо "пустий" об'єкт часу
        Time empty_time(0);
        // повертаємо його
        return empty_time;
    }
    // якщо кількість секунд лівого операнду більша за кількість
секунд правого операнду
    else{
        // рахуемо різницю в секундах
        int time_difference = left_value_seconds -
right_value_seconds;

        // створюємо об'єкт часу (заповнюємо тільки секунди)
        Time difference_time(0, 0, time_difference);
        // повертаємо його
        return difference_time;
    }
}
```

functions.h:

```
#ifndef INC_2LABWORK_4_CLASSES_H
#define INC_2LABWORK_4_CLASSES_H

#endif //INC_2LABWORK_4_CLASSES_H
#pragma once

class Time{
private:
    int hours;
    int minutes;
    int seconds;
public:
    Time();
    Time(int hours);
    Time(int hours, int minutes);
    Time(int hours, int minutes, int seconds);

    int GetHours();
    int GetMinutes();
    int GetSeconds();
    void TimeLeft();
    void ShowInfo();

    void operator+=(int minutes_to_add);
    Time operator-(Time right_value);
};
```

Результат виконання програми

```
Hours: 12, minutes: 0, seconds: 0
Hours: 13, minutes: 40, seconds: 0
Hours: 21, minutes: 43, seconds: 14

Testing getters (T3):
21:43:14

T1 before incrementing:
Hours: 12, minutes: 0, seconds: 0
T1 after incrementing:
Hours: 12, minutes: 17, seconds: 0

T2 before incrementing:
Hours: 13, minutes: 40, seconds: 0
T2 after incrementing:
Hours: 14, minutes: 14, seconds: 0

Time differece between T2 and T1:
Hours: 1, minutes: 57, seconds: 0

Time left until the end of the day (T3):
Time left until the end of the day: 2 hours 16 minutes 46 seconds

Process finished with exit code 0
```