ММИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСТИТЕТ)

**ЛАБОРАТОРНАЯ РАБОТА №1**

по курсу “Объектно-ориентированное программирование”

I семестр, 2021/22 учебный год

Студент: *Морозов Артем Борисович, группа М8О-208Б-20*

Преподаватель: *Дорохов Евгений Павлович, каф. 806*

**Задание:** Разработать программу на языке C++ согласно варианту задания. Программа должна получать данные из стандартного ввода и выводить данные в стандартный вывод.

**Вариант №14:**

Создать класс **TimePoint** для работы с моментами времени в формате «час:минута:секунда». Обязательными операциями являются: вычисление разницы между двумя моментами времени, сумма моментов времени, сложение момента времени и заданного количества секунд, вычитание из момента времени заданного количества секунд, вычисление во раз сколько один момент времени больше (меньше) другого, сравнение моментов времени, перевод в секунды и обратно, перевод в минуты (с округлением до минуты) и обратно.

**Описание программы:**

Исходный код разделён на 3 файла:

* TimePoint.h – описание основных функций класса TimePoint
* TimePoint.cpp – реализация функционала класса TimePoint
* main.cpp – основная программа

**Дневник отладки:**

Программа в отладке не нуждалась, весь необходимый функционал был реализован без всяких заминок.

**Вывод:**  
 В процессе выполнения данной лабораторной работы я, можно сказать, познакомился с самим понятием ООП – прочувствовал, что такое классы, осознал отличие класса от структуры, познакомился с понятием “метод класса”, успешно реализовал необходимый функционал для работы. На примере данной лабораторной работы я столкнулся с одним из трех китов ООП – инкапсуляцией. Действительно, в public-зоне у меня лежат все необходимые методы, а в private-зоне, по правилу хорошего тона, лежат 3 переменные по заданию: часы, минуты и секунды.

**Исходный код:**

**TimePoint.h:**

#ifndef TIMEPOINT\_H

#define TIMEPOINT\_H

#include <iostream>

class TimePoint {

public:

TimePoint();

TimePoint(int h, int m, int s);

TimePoint(std::istream &is);

TimePoint(const TimePoint &other);

void Difference(const TimePoint &other);

void Sum(const TimePoint& other);

void AddSeconds(int s);

void RemoveSeconds(int s);

int IsBigger(const TimePoint &other);

void Compare(const TimePoint &other);

int ToSeconds();

int ToMinutes();

void Print(std::ostream &os);

~TimePoint();

private:

int hours;

int minutes;

int seconds;

};

#endif **TimePoint.cpp:**

#include "TimePoint.h"

TimePoint::TimePoint() {

hours = 0;

minutes = 0;

seconds = 0;

std:: cout << "The default time-ojbect has been created" << std:: endl;

}

TimePoint::TimePoint(int h, int m, int s) {

if (h >= 0 && m >= 0 && s >= 0) {

hours = h;

minutes = m;

seconds = s;

}

else {

std:: cout << "Please enter positive numbers!" << std:: endl;

}

std:: cout << "The time-object according to your parameters has been created" << std:: endl;

}

TimePoint::TimePoint(std::istream &is) {

std:: cout << "Please enter your time-object data: " << std:: endl;

is >> hours >> minutes >> seconds;

if ((hours < 0 || hours > 23) || (minutes < 0 || minutes > 59) || (seconds < 0 || seconds > 59)) {

std:: cout << "Invalind input. Enter again!" << std:: endl;

is >> hours >> minutes >> seconds;

}

std:: cout << "The time-object has been created via istream" << std:: endl;

}

TimePoint::TimePoint(const TimePoint& other) {

hours = other.hours;

minutes = other.minutes;

seconds = other.seconds;

std:: cout << "The copy of your time-object has been created" << std:: endl;

}

void TimePoint::Difference(const TimePoint &other) {

int x = hours \* 3600 + minutes \* 60 + seconds;

int y = other.hours \* 3600 + other.minutes \* 60 + other.seconds;

int dhours, dminutes, dseconds;

if ((hours > other.hours) || (hours == other.hours && minutes > other.minutes) || (hours == other.hours && minutes == other.minutes && seconds > other.seconds)) {

int z = x - y;

dhours = z / 3600;

dminutes = (z % 3600) / 60;

dseconds = (z % 3600) - (dminutes \* 60);

}

else {

int z = y - x;

dhours = z / 3600;

dminutes = (z % 3600) / 60;

dseconds = (z % 3600) - (dminutes \* 60);

}

std:: cout << "The difference between your time-objects is: " << dhours << ":" << dminutes << ":" << dseconds << std:: endl;

}

void TimePoint::Sum(const TimePoint& other) {

int x = hours \* 3600 + minutes \* 60 + seconds;

int y = other.hours \* 3600 + other.minutes \* 60 + other.seconds;

int z = x + y;

int dhours = z / 3600;

int dminutes = (z % 3600) / 60;

int dseconds = (z % 3600) - (dminutes \* 60);

std:: cout << "The sum of your time-objects is: " << dhours << ":" << dminutes << ":" << dseconds << std:: endl;

}

void TimePoint::AddSeconds(int s) {

if (s < 0) {

std:: cout << "Please enter positive number!" << std:: endl;

}

else {

int x = hours \* 3600 + minutes \* 60 + seconds + s;

hours = x / 3600;

minutes = ((x % 3600) / 60);

seconds = (x % 3600) - (((x % 3600) / 60) \* 60);

std:: cout << "After adding seconds your time is: " << hours << ":" << minutes << ":" << seconds << std:: endl;

}

}

void TimePoint::RemoveSeconds(int s) {

if (s < 0) {

std:: cout << "Please enter positive number!" << std:: endl;

}

else {

int x = hours \* 3600 + minutes \* 60 + seconds - s;

hours = x / 3600;

minutes = ((x % 3600) / 60);

seconds = (x % 3600) - (((x % 3600) / 60) \* 60);

std:: cout << "After removing seconds your time is: " << hours << ":" << minutes << ":" << seconds << std:: endl;

}

}

int TimePoint::IsBigger(const TimePoint &other ) {

int x = hours \* 3600 + minutes \* 60 + seconds;

int y = other.hours \* 3600 + other.minutes \* 60 + other.seconds;

if ((hours > other.hours) || (hours == other.hours && minutes > other.minutes) || (hours == other.hours && minutes == other.minutes && seconds > other.seconds)) {

return x / y;

}

return y / x;

}

void TimePoint::Compare(const TimePoint &other) {

if ((hours > other.hours) || (hours == other.hours && minutes > other.minutes) || (hours == other.hours && minutes == other.minutes && seconds > other.seconds)) {

std:: cout << "The first time is more that second time!" << std:: endl;

}

else if (hours == other.hours && minutes == other.minutes && seconds == other.seconds) {

std:: cout << "Times are equal!" << std:: endl;

}

else {

std:: cout << "The second time is more that first time!" << std:: endl;

}

}

int TimePoint::ToSeconds() {

return hours \* 3600 + minutes \* 60 + seconds;

}

int TimePoint::ToMinutes() {

int z = hours \* 3600 + minutes \* 60 + seconds;

int m = z / 60;

if (z % 60 == 0) {

return m;

}

else {

if (z % 60 >= 30) {

return m + 1;

}

}

return m;

}

void TimePoint::Print(std::ostream& os) {

os << "Your current time is: " << hours << ":" << minutes << ":" << seconds << std:: endl;

}

TimePoint::~TimePoint() {

std:: cout << "FROM DESTRUCTOR: Your time-object has been deleted" << std:: endl;

}

**main.cpp:**

#include "TimePoint.h"

int main () {

TimePoint a(std:: cin);

TimePoint b(12, 38, 40);

TimePoint c(20, 20, 41);

TimePoint d(c);

c.Difference(d);

b.Sum(c);

d.AddSeconds(3600);

c.Print(std:: cout);

d.Print(std:: cout);

b.RemoveSeconds(3240);

b.Print(std:: cout);

a.Compare(c);

TimePoint e(06, 00, 00);

TimePoint f(18, 00, 00);

std:: cout << "The diffrence between times in their division is: " << e.IsBigger(f) << std:: endl;

std:: cout << "Your time in minutes is: " << a.ToMinutes() << std:: endl;

std:: cout << "Your time is seconds is: " << a.ToSeconds() << std:: endl;

return 0;

}

**Пример работы:**

