**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 1**

Тема: Простые классы на языке С++

Студент: Лагуткина Мария Сергеевна

Группа: 80-206

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

Постановка задачи

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

На ввод программы подается 2 момента времени в формате «час:минута:секунда», затем вводится количество секунд и количество минут. Если вводятся некорректные данные, программа выводит «ERROR INPUT» и завершается с кодом 1. Далее для 2-х моментов времени создаются 2 объекта TimePoint, время в которых хранится в виде секунд. Далее вызываются функции, выполняющие действия, описанные в варианте программы и на консоль выводятся результаты вычисления указанных действий.

Набор тестов

1. момент времени 1
2. момент времени 2
3. количество секунд
4. количество минут

Test 1:

00:00:00

00:00:00

0

0

Test2:

12:01:14

01:05:20

2456

1345

Test3:

12:21:14

20:59:20

1235355

1345

Test4:

12:21:14

20:59:20

1235

1345

Результаты выполнения тестов

Output1:

00:00:00 - Difference moments times

00:00:00 - Moments times sum

00:00:00 - Difference first times and seconds

00:00:00 - Sum first times and seconds

Can't difference

Times are equal

0 - Conversion of first time to seconds

00:00:00 - Conversion of seconds to time

0 - Conversion of first time to minutes

00:00:00 - Conversion of minutes to time

Output2:

10:55:54 - Difference moments times

13:06:34 - Moments times sum

11:20:18 - Difference first times and seconds

12:42:10 - Sum first times and seconds

First moment time more in 11

First time more than second time

43274 - Conversion of first time to seconds

00:40:56 - Conversion of seconds to time

721 - Conversion of first time to minutes

22:25:00 - Conversion of minutes to time

Output3:

ERROR INPUT

Output4:

08:38:06 - Difference moments times

time overflow - Moments times sum

12:00:39 - Difference first times and seconds

12:41:49 - Sum first times and seconds

Second moment time more in 1

Second time more than first time

44474 - Conversion of first time to seconds

00:20:35 - Conversion of seconds to time

741 - Conversion of first time to minutes

22:25:00 - Conversion of minutes to time

Листинг программы

//Лагуткина Мария, M8O-206Б-19, 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.

//тесты находятся в файле data.txt в таком порядке: 1-вый момент времении, 2-ой момент времени, количество секунд, количество минут

//#include "stdafx.h"

#include <conio.h>

#include <string>

#include <iostream>

#include <fstream>

#define MAX\_SECONDS 86399 //для 24-ого формата времени

#define MAX\_MINUTES 1439 //для 24-ого формата времени

using namespace std;

class TimePoint {

public:

void SetSeconds(const int new\_seconds) { seconds = new\_seconds; }

TimePoint() { seconds = 0; }

TimePoint(const int new\_hours, const int new\_minutes, const int new\_seconds) {

TranslateInSeconds(new\_hours, new\_minutes, new\_seconds);

}

~TimePoint() {}

int GetInSeconds() const { return seconds; }

string Get() const { //получение времени в формате «час:минута:секунда»

return TranslateFromSeconds();

}

private:

void TranslateInSeconds(int hour, int minute, int second) {

seconds = hour \* 3600 + minute \* 60 + second;

}

string TranslateFromSeconds() const {

int hour = seconds / 3600;

int minute = (seconds % 3600) / 60;

int second = seconds - hour \* 3600 - minute \* 60;

string h = to\_string(hour);

string m = to\_string(minute);

string s = to\_string(second);

string time = "";

if (hour < 10) { time += '0'; }

time += h;

time += ':';

if (minute < 10) { time += '0'; }

time += m;

time += ':';

if (second < 10) { time += '0'; }

time += s;

return time;

}

int seconds;

};

//вычисление разницы между двумя моментами времени

string DifferenceMomentsTimes(const TimePoint& time1, const TimePoint& time2) {

TimePoint times;

if (time1.GetInSeconds() >= time2.GetInSeconds()) {

times.SetSeconds(time1.GetInSeconds() - time2.GetInSeconds());

}

else {

times.SetSeconds(time2.GetInSeconds() - time1.GetInSeconds());

}

string time\_str = times.Get();

return time\_str;

}

//сумма двух моментов времени

string SumMomentsTimes(const TimePoint& time1, const TimePoint& time2) {

TimePoint times;

if (time1.GetInSeconds() + time2.GetInSeconds() >= MAX\_SECONDS) {

return "time overflow";

}

times.SetSeconds(time1.GetInSeconds() + time2.GetInSeconds());

string time\_str = times.Get();

return time\_str;

}

//вычитание из момента времени заданного количества секунд

string DifferenceTimesAndSeconds(const TimePoint& time1, int new\_seconds) {

TimePoint times;

if (time1.GetInSeconds() < new\_seconds) {

return "can't be difference";

}

times.SetSeconds(time1.GetInSeconds() - new\_seconds);

string time\_str = times.Get();

return time\_str;

}

//сложение момента времени и заданного количества секунд

string SumTimesAndSeconds(const TimePoint& time1, int new\_seconds) {

TimePoint times;

if (time1.GetInSeconds() + new\_seconds >= MAX\_SECONDS) {

return "Time overflow";

}

times.SetSeconds(time1.GetInSeconds() + new\_seconds);

string time\_str = times.Get();

return time\_str;

}

//вычисление во раз сколько один момент времени больше (меньше) другого

string HowManyTimesMore(const TimePoint& time1, const TimePoint& time2) {

TimePoint times;

if (time1.GetInSeconds() >= time2.GetInSeconds()) {

if (time2.GetInSeconds() == 0) {

return "Can't difference";

}

times.SetSeconds(time1.GetInSeconds() / time2.GetInSeconds());

string ans = "First moment time more in ";

ans += to\_string(times.GetInSeconds());

return ans;

}

if (time1.GetInSeconds() == 0) {

return "Can't difference";

}

times.SetSeconds(time2.GetInSeconds() / time1.GetInSeconds());

string ans = "Second moment time more in ";

ans += to\_string(times.GetInSeconds());

return ans;

}

// сравнение моментов времени

string ComporisionMomentsTimes(const TimePoint& time1, const TimePoint& time2) {

TimePoint times;

if (time1.GetInSeconds() > time2.GetInSeconds()) {

return "First time more than second time";

}

if (time2.GetInSeconds() > time1.GetInSeconds()) {

return "Second time more than first time";

}

return "Times are equal";

}

//перевод из формата момента времени в секунды

int MomentTimeToSeconds(const TimePoint& time1) {

return time1.GetInSeconds();

}

//перервод из секунд в формат момента времени

string SecondsToMomentTime(int new\_seconds) {

if (new\_seconds > MAX\_SECONDS) {

return "time overflow";

}

TimePoint times;

times.SetSeconds(new\_seconds);

string time\_str = times.Get();

return time\_str;

}

//перевод из формата момента времени в минуты(с округлением до минуты)

int MomentTimeToMinutes(const TimePoint& time1) {

int sec = time1.GetInSeconds();

int minutes = sec / 60;

sec -= minutes \* 60;

if (sec >= 30) {

++minutes;

}

return minutes;

}

//перевод из минут в формат момента времени

string MinutesToMomentTime(int new\_minutes) {

if (new\_minutes \* 60 > MAX\_SECONDS) {

return "time overflow";

}

TimePoint times;

times.SetSeconds(new\_minutes \* 60);

return times.Get();

}

bool IsHour(const int num) {

if (num < 0 || num > 24) {

return false;

}

return true;

}

bool IsMinOrSec(const int num) {

if (num < 0 || num > 59) {

return false;

}

return true;

}

bool IsAllSec(const int num) {

if (num < 0 || num > MAX\_SECONDS) {

return false;

}

return true;

}

bool IsAllMin(const int num) {

if (num < 0 || num > MAX\_MINUTES) {

return false;

}

return true;

}

int main() {

//setvbuf(stdout, NULL, \_IONBF, 0);

//setvbuf(stderr, NULL, \_IONBF, 0);

const string path = "data.txt";

ifstream input(path);

int hours1, minutes1, seconds1, hours2, minutes2, seconds2;

int sec, min;

/\*if (cin >> hours1 >> minutes1 >> seconds1

>> hours2 >> minutes2 >> seconds2 >> sec >> min)

{}

else {

cout << "ERROR INPUT" << endl;

return 1;

}

\*/

if (input.is\_open()) { //ввод двух моментов времени в фоормате «час:минута:секунда»

input >> hours1;

if (!IsHour(hours1)) { cout << "ERROR INPUT"; return 1; }

input.ignore(1);

input >> minutes1;

if (!IsMinOrSec(minutes1)) { cout << "ERROR INPUT"; return 1; }

input.ignore(1);

input >> seconds1;

if (!IsMinOrSec(seconds1)) { cout << "ERROR INPUT"; return 1; }

input >> hours2;

if (!IsHour(hours2)) { cout << "ERROR INPUT"; return 1; }

input.ignore(1);

input >> minutes2;

if (!IsMinOrSec(minutes2)) { cout << "ERROR INPUT"; return 1; }

input.ignore(1);

input >> seconds2;

if (!IsMinOrSec(seconds2)) { cout << "ERROR INPUT"; return 1; }

input >> sec; //ввод вромени в секундах

if (!IsAllSec(sec)) {cout << "ERROR INPUT"; return 1; }

input >> min; //ввод времени в минутах

if (!IsAllMin(min)) { cout << "ERROR INPUT"; return 1; }

TimePoint time1(hours1, minutes1, seconds1);

TimePoint time2(hours2, minutes2, seconds2);

cout << DifferenceMomentsTimes(time1, time2) << " - Difference moments times" << endl;

cout << SumMomentsTimes(time1, time2) << " - Moments times sum" << endl;

cout << DifferenceTimesAndSeconds(time1, sec) << " - Difference first times and seconds" << endl;

cout << SumTimesAndSeconds(time1, sec) << " - Sum first times and seconds" << endl;

cout << HowManyTimesMore(time1, time2) << endl;

cout << ComporisionMomentsTimes(time1, time2) << endl;

cout << MomentTimeToSeconds(time1) << " - Conversion of first time to seconds" << endl;

cout << SecondsToMomentTime(sec) << " - Conversion of seconds to time" << endl;

cout << MomentTimeToMinutes(time1) << " - Conversion of first time to minutes" << endl;

cout << MinutesToMomentTime(min) << " - Conversion of minutes to time" << endl;

}

else { cout << "ERROR"; }

\_getch();

return 0;

}

Список литературы

Курс «Основы разработки на С++: белый пояс», неделя 4. Ресурс: <https://www.coursera.org/learn/c-plus-plus-white>