**Московский авиационный институт**

**(национальный исследовательский университет)**

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Дисциплина «Объектно-ориентированное программирование»

**Лабораторная работа №1**

Тема: Простые классы на языке C++

Студент: Волков М А.

Группа: М8О-207Б-19

Преподаватель: Чернышев Л. Н.

Дата:

Оценка:

**Постановка задачи**

*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 сколько раз 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**

Программа написана в двух файлах main.cpp и main.h

В main.h в классе, под названием **TimePoint,** описаны методы которые умеют выполнять вышеперечисленные функции. В *private* описаны методы конвертирования и ему подобные операции. Однако перегрузки операторов сложения и вычитания написаны вне класса. Сделано это было для того, чтобы можно было отладить правильно суммирование и вычитание и, в случае необходимости, в main.cpp воспользоваться сложением.

В *public* описаны все методы, которые выполняют вышеизложенные функции, в том числе описанны геттеры и сеттеры.

Также в обязательном порядке почти каждый метод перегружен, чтобы пользователь мог использовать объект на сторонних данных, при этом не меняя исходные.

В обязательном порядке также сделана “защита от дураков”, которая завершает программу, когда пользователь в неправильном формате вводит время.

**Набор и результаты выполнения тестов**

Тест №1

Enter your time for show difference

12:45:00

12:34:08

time is 00:10:52

Enter time for sum

12:00:34

12:34:23

time is 00:34:57

Enter time for sum in sec. The First one is time.

The second one is seconds

12:34:59

2

time in sec: 45301

time is 12:35:01

Enter time for sub in sec. The First one is time.

The second one is seconds

12:00:00

1

time is 11:59:59

Enter time for how times bigger (smaller)

12:45:00

00:34:01

about 22 times

Enter time for show subtraction

12:30:58

14:23:45

time is 22:07:13

Enter time for convert in seconds and minutes

23:45:12

10:33:11

time in minutes 4080

time in seconds 37991

Enter sec and minutes for convert in time

234

45

time is 00:03:54

time is 00:45:00

Process finished with exit code 0

Тест №2

Enter your time for show difference

00:00:00

23:59:59

time is 23:59:59

Enter time for sum

23:58:58

00:00:02

time is 23:59:00

Enter time for sum in sec. The First one is time.

The second one is seconds

23:59:59

1234

time in sec: 87633

time is 00:20:33

Enter time for sub in sec. The First one is time.

The second one is seconds

00:00:00

0

time is 00:00:00

Enter time for how times bigger (smaller)

23:23:43

00:00:00

WARNING: division by zero

PROPOSAL: check your time entering

Process finished with exit code 1

Тест №3

Enter your time for show difference

45:12:60

Wrong time format

Process finished with exit code 1

Тест №4

Enter your time for show difference

00:45:12

F:12:t

Wrong time format

Process finished with exit code 1

Тест №5

Enter your time for show difference

12:34:00

16:33:59

time is 03:59:59

Enter time for sum

12:25:38

14:34:12

time is 02:59:50

Enter time for sum in sec. The First one is time.

The second one is seconds

12:34:09

1234562

time in sec: 1279811

time is 19:30:11

Enter time for sub in sec. The First one is time.

The second one is seconds

12:34:09

123

time is 12:32:06

Enter time for how times bigger (smaller)

12:34:12

12:00:00

about 1.0475 times

Enter time for show subtraction

00:00:1

00:22:00

time is 23:38:01

Enter time for convert in seconds and minutes

4231

Wrong time format

Process finished with exit code 1

**Листинг программы**

main.cpp

/\*Волков Матвей Андреевич

\*

\* 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 сколько раз один момент времени

\* больше (меньше) другого, сравнение моментов времени, перевод в секунды и обратно, перевод в

\* минуты (с округлением до минуты) и обратно.

\*/

#include "main.h"

int main(){

TimePoint time1, time2;

int sec;

std::cout<< "Enter your time for show difference" << std::endl;

time1.setTime\_in\_format();

time2.setTime\_in\_format();

time1.show\_difference(time2.get\_time\_in\_format());

std::cout << std::endl << "Enter time for sum" << std::endl;

time1.setTime\_in\_format();

time2.setTime\_in\_format();

time1.sum\_time\_in\_format(time2.get\_time\_in\_format());

std::cout << std::endl << "Enter time for sum in sec. The First one is time. \n"

"The second one is seconds" << std::endl;

time1.setTime\_in\_format();

std::cin >> sec;

time1.show\_seconds\_in\_time(time1.sum\_time\_seconds(sec));

std::cout << std::endl << "Enter time for sub in sec. The First one is time. \n"

"The second one is seconds" << std::endl;

time1.setTime\_in\_format();

std::cin >> sec;

time1.sub\_time\_seconds(sec);

std::cout << std::endl << "Enter time for how times bigger (smaller)" << std::endl;

time1.setTime\_in\_format();

time2.setTime\_in\_format();

time1.how\_times(time2.get\_time\_in\_format());

std::cout << std::endl << "Enter time for show subtraction" << std::endl;

time1.setTime\_in\_format();

time2.setTime\_in\_format();

time1.sub\_time\_in\_format(time2.get\_time\_in\_format());

std::cout << std::endl << "Enter time for convert in seconds and minutes" << std::endl;

time1.setTime\_in\_format();

time2.setTime\_in\_format();

time1.show\_time\_in\_minutes();

time2.show\_time\_in\_seconds();

std::cout << std::endl << "Enter sec and minutes for convert in time" << std::endl;

int min;

std::cin >> sec >> min;

time1.show\_seconds\_in\_time(sec);

time2.show\_minutes\_in\_time(min);

return 0;

}

main.h

#ifndef OOP\_EXERCISE\_01\_MAIN\_H

#define OOP\_EXERCISE\_01\_MAIN\_H

#include <string>

#include <iostream>

struct Time {

int m,

h,

s;

};

bool operator<(const Time &lhs, const Time &rhs){

if(lhs.h != rhs.h){

return lhs.h < rhs.h;

}

else if(lhs.m != rhs.m){

return lhs.m < rhs.m;

}

return lhs.s < rhs.s;

}

Time operator-(const Time &lhs, const Time &rhs){

Time result{};

if(lhs.h - rhs.h >= 0) {

result.h = lhs.h - rhs.h;

}

else {

result.h = 24 + lhs.h - rhs.h;

}

if(lhs.m - rhs.m >= 0) {

result.m = lhs.m - rhs.m;

}

else {

if(result.h > 0) {

result.h = result.h - 1;

}

else{

result.h = 24 - 1;

}

result.m = 60 + lhs.m - rhs.m;

}

if(lhs.s - rhs.s >= 0) {

result.s = lhs.s - rhs.s;

}

else {

if(result.m > 0) {

result.m = result.m - 1;

}

else{

result.m = 60 - 1;

if(result.h > 0) {

result.h = result.h - 1;

}

else{

result.h = 24 - 1;

}

}

result.s = 60 + lhs.s - rhs.s;

}

return result;

}

/\*

Time operator=(const Time &tmp\_time){

Time result{};

result.h = tmp\_time.h;

result.m = tmp\_time.m;

result.s = tmp\_time.s;

return result;

}

\*/

Time operator+(const Time &lhs, const Time &rhs){

Time result{};

result.h = (lhs.h + rhs.h) % 24;

if(lhs.m + rhs.m >= 60){

result.h = (result.h + 1) % 24;

result.m = (lhs.m + rhs.m) % 60;

}

else {

result.m = lhs.m + rhs.m;

}

if(lhs.s + rhs.s >= 60){

if(result.m == 59){

result.h = (result.h + 1) % 24;

result.m = 0;

result.s = (lhs.s + rhs.s) % 60;

}

else{

result.m++;

result.s = (lhs.s + rhs.s) % 60;

}

}

else {

result.s = lhs.s + rhs.s;

}

return result;

}

//----------------------the beginning of class----------------------

class TimePoint {

//---------private----------

private:

Time time{0,0,0};

Time convert\_to\_time(const std::string& a){

time = {0,0,0};

int x = 0;

for (auto i : a) {

if(i == ':') {

x++;

continue;

}

if(i >= '0' && i<= '9'){

if(x == 0)

time.h = time.h\*10 + (i - '0');

if (x == 1)

time.m = time.m\*10 + (i - '0');

if (x == 2)

time.s = time.s\*10 + (i - '0');

}

if(i < '0' || i> '9'){

std::cout << "Wrong time format";

exit (1);

}

}

if(x >= 3 || time.h>23 || time.m >=60 || time.s >=60 || (time.h<0 && time.m<0 && time.s<0) || x == 0) {

std::cout << "Wrong time format";

exit (1);

}

return time;

}

int time\_to\_seconds(){

return time.h\*60\*60 + time.m\*60 + time.s;

}

int time\_to\_seconds(Time time\_tmp){

return time\_tmp.h\*60\*60 + time\_tmp.m\*60 + time\_tmp.s;

}

Time seconds\_to\_time(int tmp){

Time result{};

result.h = (tmp/3600) % 24;

result.m = (tmp%3600)/60;

result.s = tmp%60;

return result;

}

Time minutes\_to\_time(int tmp){

Time result{};

result.h = (tmp / 60) % 24;

result.m = tmp % 60;

result.s = 0;

return result;

}

int time\_to\_minutes() {

int result;

result = time.h \* 60 + time.m \* 60;

if (time.s >= 30) {

result++;

}

return result;

}

int time\_to\_minutes(Time time\_tmp){

int result;

result = time\_tmp.h\*60 + time\_tmp.m\*60;

if (time\_tmp.s >=30){

result++;

}

return result;

}

//---------public-----------

public:

Time setTime\_in\_format(){

std::string a;

std::cin >> a;

return convert\_to\_time(a);

}

void show\_time\_in\_format() const{

if (time.h >= 10 && time.m >= 10 && time.s>=10) {

std::cout << "time is " << time.h << ':' << time.m << ':' << time.s << std::endl;

}

else {

if (time.h < 10)

std::cout << "time is " << '0' << time.h << ':';

else

std::cout << "time is " << time.h << ':';

if (time.m < 10)

std::cout << '0' << time.m << ':';

else

std::cout << time.m << ':';

if (time.s < 10)

std::cout <<'0' << time.s << std::endl;

else

std::cout << time.s << std::endl;

}

}

void show\_time\_in\_format(const Time &time\_tmp) const {

if (time\_tmp.h >= 10 && time\_tmp.m >= 10 && time\_tmp.s >=10) {

std::cout << "time is " << time\_tmp.h << ':' << time\_tmp.m << ':' << time\_tmp.s << std::endl;

}

else {

if (time\_tmp.h < 10)

std::cout << "time is " << '0' << time\_tmp.h << ':';

else

std::cout << "time is " << time\_tmp.h << ':';

if (time\_tmp.m < 10)

std::cout << '0' << time\_tmp.m << ':';

else

std::cout << time\_tmp.m << ':';

if (time\_tmp.s < 10)

std::cout <<'0' << time\_tmp.s << std::endl;

else

std::cout << time\_tmp.s << std::endl;

}

}

Time get\_time\_in\_format(){

return time;

}

void show\_difference(Time lhs, Time rhs){

if(lhs < rhs){

Time tmp;

tmp = lhs;

lhs = rhs;

rhs = tmp;

}

Time result = lhs - rhs;

show\_time\_in\_format(result);

}

void show\_difference(Time rhs){

Time lhs = time; //lhs здесь нужно,чтобы не менять значения в классе

if(lhs < rhs){

Time tmp{};

tmp = lhs;

lhs = rhs;

rhs = tmp;

}

Time result = lhs - rhs;

show\_time\_in\_format(result);

}

void sum\_time\_in\_format(const Time &lhs, const Time &rhs){

Time result = lhs + rhs;

show\_time\_in\_format(result);

}

void sum\_time\_in\_format(const Time &rhs){

time = time + rhs;

show\_time\_in\_format();

}

int sum\_time\_seconds(Time lhs, int &sec){

int result = time\_to\_seconds(lhs);

result += sec;

std::cout << "time in sec: " << result << std::endl;

return result;

}

int sum\_time\_seconds(int &sec){

int result = time\_to\_seconds(time);

result += sec;

std::cout << "time in sec: " << result << std::endl;

time = time + seconds\_to\_time(sec);

return result;

}

void sub\_time\_in\_format(const Time &lhs, const Time &rhs){

Time result{};

result = lhs - rhs;

show\_time\_in\_format(result);

}

void sub\_time\_in\_format(const Time &lhs){

time = time - lhs;

show\_time\_in\_format();

}

void sub\_time\_seconds(Time lhs, int &sec){

int result = time\_to\_seconds(lhs);

if(result < sec){

result += 60\*60\*24;

}

result -= sec;

show\_time\_in\_format(seconds\_to\_time(result));

}

void sub\_time\_seconds(const int &sec){

time = time - seconds\_to\_time(sec);

show\_time\_in\_format();

}

void how\_times(const Time &tmp\_time){

float time\_in\_sec = time\_to\_seconds();

float tmptime\_in\_sec = time\_to\_seconds(tmp\_time);

if(tmptime\_in\_sec == 0){

std::cout << "WARNING: division by zero\n"

"PROPOSAL: check your time entering" << std::endl;

exit(1);

}

std::cout << "about " << time\_in\_sec/tmptime\_in\_sec << " times" << std::endl;

}

void show\_time\_in\_minutes(){

std::cout << "time in minutes " << time\_to\_minutes() << std::endl;

}

void show\_time\_in\_minutes(const Time &tmp){

std::cout << "time in minutes " << time\_to\_minutes(tmp) << std::endl;

}

void show\_minutes\_in\_time(const int &min){

Time result = minutes\_to\_time(min);

show\_time\_in\_format(result);

}

void show\_seconds\_in\_time(const int &sec){

show\_time\_in\_format(seconds\_to\_time(sec));

}

void show\_time\_in\_seconds(){

std::cout << "time in seconds " << time\_to\_seconds() << std::endl;

}

void show\_time\_in\_seconds(Time tmp\_time){

std::cout << "time in seconds " << time\_to\_seconds(tmp\_time) << std::endl;

}

};

//---------------------the end of class---------------------------

#endif //OOP\_EXERCISE\_01\_MAIN\_H

**Выводы**

Я научился создавать классы, описывать их члены и методы на языке C++, познакомился с автоматической системой сборки CMake, вводом и выводом из библиотеки iostream и строками из библиотеки string.

**Список литературы**

1. Р. Лафоре «ООП в С++»