

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Кафедра 806 «Вычислительная информатика и программирование»
Факультет: «Информационные технологии и прикладная математика»

Лабораторная работа
Дисциплина: «Объектно-ориентированное программирование»
III семестр
Задание 2: «Операторы, литералы»

Группа:	М8О-206Б-18, №14
Студент:	Орозбакиев Э.Д.
Преподаватель:	Журавлёв Андрей Андреевич
Оценка:	
Дата:	

Москва, 2019

1. Задание

(вариант № 14): Создать класс TimePoint для работы в 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 округление до минуты и обратно).

Операция сложения и вычитания в TimePoint, а также сравнения необходимо реализовать в виде перегрузки операторов.

Необходимо реализовать пользовательский литерал для работы с константами типа TimePoint.

2. Адрес репозитория на GitHub

https://github.com/p0kemo4ik/oop_exercise_02

3. Код программы на C++

Main.cpp

```
#include <iostream>
#include "TimePoint.hpp"
int main() {
    TimePoint a;
    std::cout<< "Введите первый момент времени" << std::endl;
    std::cin >> a;
    TimePoint b;
    std::cout<< "Введите второй момент времени" << std::endl;
    std::cin >> b;
    TimePoint c;
    c = a + b;
    std::cout<< "Сумма 1 и 2 момента времени" << std::endl;
    std::cout << c << std::endl;
    c = a - b;
    std::cout<< "Разница 1 и 2 момента времени" << std::endl;
    std::cout << c << std::endl;
    c = b - a;
    std::cout<< "Разница 2 и 1 момента времени" << std::endl;
    std::cout << c << std::endl;
    bool d;
    d = a > b;
```

```

std::cout<< "Сравнение 1 > 2" << std::endl;
std::cout << d << std::endl;
d = a < b;
std::cout<< "Сравнение 1 < 2" << std::endl;
std::cout << d << std::endl;
d = a == b;
std::cout<< "Сравнение 1 == 2" << std::endl;
std::cout << d << std::endl;
TimePoint g;
g = 90000_sec;
std::cout<< "Пользовательский литерал _sec" << std::endl;
std::cout<< g << std::endl;
g = 228_min;
std::cout<< "Пользовательский литерал _min" << std::endl;
std::cout<< g << std::endl;
}

```

TimePoint.cpp

```

#include "TimePoint.hpp"

#include <sstream>
TimePoint::TimePoint() {
    second = 0;
    minute = 0;
    hour = 0;
}
TimePoint::TimePoint(int h, int m, int s) {
    hour=h;
    minute=m;
    second=s;
}
void TimePoint::Read(std::istream &is){
    is >> hour >> minute >> second;
}
void TimePoint::Write(std::ostream &os) {
    os << hour << ":" << minute << ":" << second << std::endl;
}
int TimePoint::ToSec() {
    int result;
    result = second + minute*60 + hour*3600;
    return result;
}
int TimePoint::ToMinute() {
    int result;

```

```

    result = ToSec() / 60;
    return result;
}
TimePoint TimePoint::ToTpFromSec(int &sec) {
    TimePoint result;
    result.hour = sec / 3600;
    result.minute = sec / 60 - result.hour*60;
    result.second = sec - (result.minute)*60 -(result.hour)*3600;
    return result;
}
TimePoint TimePoint::ToTpFromMin(int &min) {
    TimePoint result;
    result.hour = min / 60;
    result.minute = min - (result.hour)*60;
    return result;
}
TimePoint operator+ (TimePoint &a, TimePoint &b){
    TimePoint result;
    int normal = (a.ToSec() + b.ToSec()) % 86400;
    return result.ToTpFromSec(normal);
}
TimePoint operator- (TimePoint &a, TimePoint &b) {
    TimePoint result;
    int normal = (a.ToSec() - b.ToSec());
    if(normal >= 0){
        return result.ToTpFromSec(normal);
    } else {
        normal+=86400;
        return result.ToTpFromSec(normal);
    }
}
TimePoint TimePoint::AddSec(int &sec) {
    TimePoint result;
    int normal = (ToSec() + sec) % 86400;
    return result.ToTpFromSec(normal);
}
TimePoint TimePoint::SubtractSec(int &sec) {
    TimePoint result;
    int normal = (ToSec() - sec);
    if(normal >= 0){
        return result.ToTpFromSec(normal);
    } else {
        normal+=86400;
        return result.ToTpFromSec(normal);
    }
};

```

```

}
int TimePoint::HowManyTimesBigger(TimePoint &a) {
    int result = (ToSec()) / (a.ToSec());
    return result;
}
bool TimePoint::IsCorrect(){
    if(hour>24){
        return false;
    } else if(minute>60){
        return false;
    } else if(second>60){
        return false;
    } else {
        return true;
    }
}
bool operator> (TimePoint &a, TimePoint &b){
    return (a.ToSec() > b.ToSec());
}
bool operator< (TimePoint &a, TimePoint &b){
    return (a.ToSec() < b.ToSec());
}
bool operator== (TimePoint &a, TimePoint &b){
    return (a.ToSec()==b.ToSec());
}
std::ostream& operator<< (std::ostream &out, const TimePoint &point) {
    out << point.hour << ":" << point.minute << ":" << point.second;
    return out;
}
std::istream& operator>> (std::istream &in, TimePoint &point) {
    in >> point.hour >> point.minute >> point.second;
    return in;
}
TimePoint operator "" _min(unsigned long long a){
    unsigned long long b = a / 60;
    unsigned long long c = a % 60;
    TimePoint d(0, 0, 0);
    d.minute=c;
    d.hour=b;
    return d;
}
TimePoint operator "" _sec(unsigned long long a){
    unsigned long long h = a / 3600;
    unsigned long long m = a/60 - h*60;
    unsigned long long s = a - h*3600 - m*60;

```

```
TimePoint d(0, 0, 0);  
d.hour =h;  
d.minute=m;  
d.second=s;  
return d;  
}
```

4. Результаты выполнения тестов

Тест 1 :

Введите первый момент времени
Введите второй момент времени
Сумма 1 и 2 момента времени
23:59:59
Разница 1 и 2 момента времени
23:59:59
Разница 2 и 1 момента времени
0:0:1
Сравнение 1 > 2
1
Сравнение 1 < 2
0
Сравнение 1 == 2
0
Пользовательский литерал _sec
25:0:0
Пользовательский литерал _min
3:48:0

Тест 2:

Сумма 1 и 2 момента времени
0:0:0
Разница 1 и 2 момента времени
0:0:0
Разница 2 и 1 момента времени
0:0:0
Сравнение 1 > 2
0

Сравнение $1 < 2$

0

Сравнение $1 == 2$

1

Тест 3

Сумма 1 и 2 момента времени

16:23:12

Разница 1 и 2 момента времени

7:36:48

Разница 2 и 1 момента времени

16:23:12

Сравнение $1 > 2$

1

Сравнение $1 < 2$

0

Сравнение $1 == 2$

0

Тест 4

Сумма 1 и 2 момента времени

6:29:44

Разница 1 и 2 момента времени

19:21:22

Разница 2 и 1 момента времени

4:38:38

Сравнение $1 > 2$

0

Сравнение $1 < 2$

1

Сравнение $1 == 2$

0

Тест 5

Сумма 1 и 2 момента времени

17:57:0

Разница 1 и 2 момента времени

14:43:0

Разница 2 и 1 момента времени

9:17:0

Сравнение $1 > 2$

0

Сравнение $1 < 2$

1
Сравнение 1 == 2
0

5. Объяснение результатов работы программы

Программа получает на стандартный ввод два момента времени и производит все необходимые операции обусловленные моим вариантом.

Арифметические и логические операторы класса TimePoint.

Оператор	Функция оператора
operator+, operator-,	Сложение полей TimePoint
Операторы сравнения	Сравнивают разницу во времени

6. Вывод

Благодаря перегрузке логических операторов программист, который будет использовать данный класс, получает более гибкое и простое управление над экземплярами класса, а благодаря литералам пользователь в одной строке может задать параметры экземпляра и начать работу с ним.