

Московский Авиационный Институт
(Национальный исследовательский Университет)

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

**Лабораторная работа
по курсу «ООП»**

**Тема:
Простые классы.**

Студент:	Николаев В.А.
Группа:	М80-206Б-18
Преподаватель:	Журавлев А.А.
Вариант:	14
Оценка:	
Дата:	

Москва
2019

1. Код программы на языке C++:

timepoint.h:

```
#include <iostream>

class TimePoint
{
public:
    int hours;
    int min;
    int sec;

    TimePoint summing(TimePoint second);

    TimePoint difference(TimePoint second);

    TimePoint secadding(int addsec);

    TimePoint minadding(int addmin);

    long long trans_to_sec();

    long long trans_to_min();
};
```

timepoint.cpp:

```
#include <iostream>
#include <cmath>
#include "timepoint.h"

TimePoint TimePoint::summing(TimePoint second)
{
    this->sec += second.sec;
    this->min += second.min + (this->sec / 60);
    this->hours += second.hours + (this->min / 60);
    this->min = this->min % 60;
    this->sec = this->sec % 60;
    this->hours = this->hours % 24;
}

TimePoint TimePoint::difference(TimePoint second)
{
    this->sec -= second.sec;
    if (this->sec < 0) {
        this->min -= 1;
        this->sec += 60;
    }
}
```

```

    this->min -= second.min;
    if (this->min < 0) {
        this->hours -= 1;
        this->min += 60;
    }
    this->hours -= second.hours;
    this->hours = abs(this->hours);
}
TimePoint TimePoint::secadding(int addsec)
{
    this->hours = 0, this->min = 0, this->sec = 0;
    this->sec = addsec % 60;
    this->min = (addsec / 60) % 60;
    this->hours = addsec / 3600;
}
TimePoint TimePoint::minadding(int addmin)
{
    this->hours = 0, this->min = 0, this->sec = 0;
    if ((addmin / 60) > 1) {
        this->min = addmin % 60;
        this->hours = addmin / 60;
    }
    else {
        this->sec = addmin;
    }
}
}
long long TimePoint::trans_to_sec() {
    this->min = (this->hours * 60) + this->min;
    this->sec = (this->min * 60) + this->sec;
}
long long TimePoint::trans_to_min() {
    this->min = (this->hours * 60) + this->min;
    if (this->sec >= 30) {
        this->min += 1;
    }
}
}

```

main.cpp:

```
#include <iostream>
```

```
#include "timepoint.h"
```

```
int main()
```

```
{
```

```
    std::cout << "Меню\n1)Вычисление разницы между двумя моментами
времени\n2)Сумма моментов времени\n3)Сложение момента времени и
заданного количества секунд\n4)Вычитание из момента времени заданного
```

количества секунд\n5)Вычисление, во сколько раз один момент времени больше(меньше) другого\n6)Сравнение моментов времени\n7)Перевод в секунды\n8)Перевод в минуты\n9)Перевод из секунд в формат час:минута:секунда\n10)Перевод из минут в формат час:минута:секунда\n";

```
    TimePoint* arrow;
    TimePoint res;
    std::string npt;
    char tmp;
    std::cin >> npt;
    if (npt == "1") {
        std::cout << "Ввод осуществляется в формате час:минута:секунда\nВведите момент, из которого необходимо произвести вычитание: ";
        TimePoint first;
        TimePoint second;
        std::cin >> first.hours >> tmp >> first.min >> tmp >> first.sec;
        std::cout << "Введите момент времени, который необходимо вычесть: ";
        std::cin >> second.hours >> tmp >> second.min >> tmp >> second.sec;
        first.difference(second);
        std::cout << "Результат:\n" << first.hours << " часов " << first.min << " минут " << first.sec << " секунд ";
    }
    if (npt == "2") {
        std::cout << "Ввод осуществляется в формате час:минута:секунда\nВведите первый момент времени: ";
        TimePoint first;
        std::cin >> first.hours >> tmp >> first.min >> tmp >> first.sec;
        std::cout << "Введите второй момент времени: ";
        TimePoint second;
        std::cin >> second.hours >> tmp >> second.min >> tmp >> second.sec;
        first.summaring(second);
        std::cout << "Результат:\n" << first.hours << " часов " << first.min << " минут " << first.sec << " секунд ";
    }
    if (npt == "3") {
        std::cout << "Ввод осуществляется в формате час:минута:секунда\nВведите момент времени: ";
        TimePoint moment;
        int addsec;
        std::cin >> moment.hours >> tmp >> moment.min >> tmp >> moment.sec;
        std::cout << "Введите количество секунд: ";
        std::cin >> addsec;
        res = arrow->secadding(addsec);
        res.summaring(moment);
        std::cout << "Результат:\n" << res.hours << " часов " << res.min << " минут " << res.sec << " секунд ";
    }
```

```

    }
    if (npt == "4") {
        std::cout << "Ввод осуществляется в формате час:минута:секунда\nВведите
момент времени: ";
        TimePoint moment;
        std::cin >> moment.hours >> tmp >> moment.min >> tmp >> moment.sec;
        int addsec;
        std::cout << "Введите количество секунд: ";
        std::cin >> addsec;
        res = arrow->secadding(addsec);
        res.difference(moment);
        std::cout << "Результат:\n" << res.hours << " часов " << res.min << " минут "
<< res.sec << " секунд ";
    }
    if (npt == "5") {
        std::cout << "Ввод осуществляется в формате час:минута:секунда\nВведите
первый момент времени: ";
        TimePoint first;
        double answ = 0;
        std::cin >> first.hours >> tmp >> first.min >> tmp >> first.sec;
        std::cout << "Введите второй момент времени: ";
        TimePoint second;
        std::cin >> second.hours >> tmp >> second.min >> tmp >> second.sec;
        long long a = 0;
        long long b = 0;
        a = first.trans_to_sec();
        b = second.trans_to_sec();
        if ((a / b) > 0) {
            answ = double(a) / double(b);
            std::cout << "Первый момент больше второго в " << answ << " раз\n";
        }
        else {
            answ = double(b) / double(a);
            std::cout << "Первый момент меньше второго в " << answ << " раз\n";
        }
    }

    if (npt == "6") {
        std::cout << "Ввод осуществляется в формате час:минута:секунда\nВведите
первый момент времени: ";
        TimePoint first;
        std::cin >> first.hours >> tmp >> first.min >> tmp >> first.sec;
        std::cout << "Введите второй момент времени: ";
        TimePoint second;
        std::cin >> second.hours >> tmp >> second.min >> tmp >> second.sec;
    }
}

```

```

long long a = 0;
long long b = 0;
a = first.trans_to_sec();
b = second.trans_to_sec();
if (a > b) {
    std::cout << "Первый момент времени больше второго";
}
if (a < b) {
    std::cout << "Второй момент времени больше первого";
}
if (a == b) {
    std::cout << "Они равны";
}

}
if (npt == "7") {
    std::cout << "Ввод осуществляется в формате час:минута:секунда\nВведите
момент времени: ";
    TimePoint moment;
    std::cin >> moment.hours >> tmp >> moment.min >> tmp >> moment.sec;
    long long answ = moment.trans_to_sec();
    std::cout << "Результат:\n" << answ << " секунд ";
}
if (npt == "8") {
    std::cout << "Ввод осуществляется в формате час:минута:секунда\nВведите
момент времени: ";
    TimePoint moment;
    std::cin >> moment.hours >> tmp >> moment.min >> tmp >> moment.sec;
    long long answ = moment.trans_to_min();
    std::cout << "Результат:\n" << answ << " минут ";
}
if (npt == "9") {
    int addsec;
    std::cout << "Введите количество секунд: ";
    std::cin >> addsec;
    res.secadding(addsec);
    std::cout << "Результат:\n" << res.hours % 24 << " часов " << res.min << "
минут " << res.sec << " секунд ";
}
if (npt == "10") {
    int addmin;
    std::cout << "Введите количество минут: ";
    std::cin >> addmin;
    res.minadding(addmin);
}

```

```
std::cout << "Результат:\n" << res.hours % 24 << " часов " << res.min << "
минут " << res.sec << " секунд ";
}
}
```

CmakeLists.txt:

```
project(lab1)

add_executable(lab1
main.cpp
timepoint.cpp
timepoint.h
)
```

2. Ссылка на репозиторий на GitHub.

https://github.com/a1dv/oop_exercise_01.git

3. Набор тестов.

```
test_00.test:
22:13:52
14:37:58
8833123
```

```
test_01.test:
14:59:22
18:34:21
12312
```

```
test_02.test:
1:1:1
23:59:59
14124124
```

4. Результаты выполнения тестов.

test_00.result:

Сложение:

Результат:

12 часов 51 минут 50 секунд

Вычитание:

Результат:

7 часов 35 минут 54 секунд

Добавление секунд:

Результат:

3 часов 52 минут 35 секунд

Вычитание секунд:

Результат:

8 часов 35 минут 9 секунд

Сравнение:

Первый момент больше второго в 1.51927 раз

Перевод в секунды:

Результат:

80032 секунд

Перевод в минуты:

Результат:

1334 минут

Перевод из секунд:

Результат:

5 часов 38 минут 43 секунд

Перевод из минут:

Результат:

2 часов 43 минут 0 секунд

test_01.result:

Сложение:

Результат:

9 часов 33 минут 43 секунд

Вычитание:

Результат:

20 часов 25 минут 1 секунд
Добавление секунд:
Результат:
18 часов 24 минут 34 секунд
Вычитание секунд:
Результат:
11 часов 34 минут 10 секунд
Сравнение:
Первый момент меньше второго в 1.23904 раз
Перевод в секунды:
Результат:
53962 секунд
Перевод в минуты:
Результат:
899 минут
Перевод из секунд:
Результат:
3 часов 25 минут 12 секунд
Перевод из минут:
Результат:
13 часов 12 минут 0 секунд

test_02.result:
Сложение:
Результат:
1 часов 1 минут 0 секунд
Вычитание:
Результат:
1 часов 1 минут 2 секунд
Добавление секунд:
Результат:
12 часов 23 минут 5 секунд
Вычитание секунд:
Результат:
11 часов 38 минут 57 секунд
Сравнение:
Первый момент меньше второго в 23.5998 раз
Перевод в секунды:
Результат:
3661 секунд
Перевод в минуты:
Результат:
61 минут
Перевод из секунд:
Результат:

11 часов 22 минут 4 секунд

Перевод из минут:

Результат:

10 часов 4 минут 0 секунд

5. Объяснение результатов работы программы.

- 1) Ввод осуществляется через поток стандартного ввода
- 2) Вывод осуществляется через поток стандартного вывода.
- 3) Ввод осуществляется в формате час:минута:секунда.
- 5) С помощью метода `summarizing` класса `Timepoint` производится сложение двух моментов времени
- 6) С помощью метода `difference` класса `Timepoint` вычисляется разность между двумя моментами времени

- 7) С помощью метода `secadding` класса `Timepoint` к заданному моменту времени прибавляется заданное количество секунд
- 8) С помощью метода `difference` класса `Timepoint` из заданного момента времени вычитается заданное количество секунд
- 9) С помощью метода `trans_to_sec` класса `Timepoint` производится сравнение двух моментов времени
- 10) С помощью метода `trans_to_sec` класса `Timepoint` производится перевод в секунды
- 11) С помощью метода `trans_to_min` класса `Timepoint` производится перевод в минуты

6. Вывод.

Stake оказался очень удобным инструментом для работы с проектами, включающими в себя несколько файлов, а git оказался очень удобен для контроля версий. Работа с классами и структурами, а так же реализация для них некоторых арифметических и логических операций позволили глубже понять их различия и общие черты.