**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 2**

Тема: Перегрузка операторов в С++

Студент: Черемисинов Максим (староста)

Группа: 80-201

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2019

1. **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.

Операции сложения и вычитания TimePoint, а так же сравнения (больше, меньше и равно) необходимо реализовать в виде перегрузки операторов.

Необходимо реализовать пользовательский литерал для работы с константами типа TimePoint.

1. **Репозиторий github**

<https://github.com/devepodete/oop_exercise_02>

1. **Описание программы**

Реализовал класс TimePoint, время хранится в секундах для удобства проведения операций. Также реализованы операторы -, +, =, ==, !=, <, <=, >, >=. Некоторые операторы реализованы как friend-функции, чтобы были справедливы выражения вида “Число”+”Объект”. Так же справедливы выражения с несколькими членами: a+b+c и тп.

Функция time\_get() возвращает время, time\_set(int time) устанавливает время, show(TimePoint a) выводит время в обычном формате.

Функции gt(TimePoint a, TimePoint b) и lt(TimePoint a, TimePoint b) сравнивают, во сколько раз первый операнд больше или меньше второго соответственно при помощи деления. Если происходит деление на 0, то создаётся исключение.

Функция in\_minutes(int sec) переводит время из секунд в минуты (с округлением).

Для удобства пользования сделал меню с основными функциями, состоящее из 12 пунктов. Есть возможность самостоятельно создать моменты времени при помощи вектора.

Реализовал возможность создавать константные объекта класса при помощи пользовательских литералов.

1. **Набор testcases**

Тестовые файлы: **test\_\_01.txt**, **test\_\_02.txt**.

**test\_\_01.txt:**

1

10 20 40

1

9 40 50

2

6

2

0 1

7

2

0 1

3

0

4

0

5

0

0

Проверка ввода, вывода времени в стандартном формате, минутах и секундах, суммы и разницы моментов времени.

**test\_\_02.txt:**

1

5 0 0

1

1 0 0

1

0 0 0

2

8

1

22 50 10

200

9

2

0 2

10

g

2

0 1

10

l

1

0 0 0

1 2 3

Проверка добавления к моменту времени секунд, сравнения двух моментов времени, работы исключения при делении на ноль.

1. **Результаты выполнения тестов.**

Все тесты успешно пройдены, программа выдаёт верные результаты, корректно обрабатывает время.

1. **Листинг программы**

/\*

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.

Операции сложения и вычитания TimePoint, а так же сравнения (больше, меньше и равно) необходимо реализовать в виде перегрузки операторов.

Необходимо реализовать пользовательский литерал для работы с константами типа TimePoint.

Черемисинов Максим Леонидович

М8О-201Б-18

\*/

#include <iostream>

#include <vector>

#include <stdexcept>

using namespace std;

class TimePoint{

int time;

public:

TimePoint(){

time = 0;

}

TimePoint(int sec){

if(sec < 0){

time = -sec;

}else{

time = sec;

}

}

TimePoint(int hrs, int min, int sec){

hrs = hrs < 0 ? -hrs : hrs;

min = min < 0 ? -min : min;

sec = sec < 0 ? -sec : sec;

time = hrs\*3600 + min\*60 + sec;

}

int time\_get();

void time\_set(int t);

friend void show(TimePoint a);

void operator=(TimePoint a);

TimePoint operator+(TimePoint a);

TimePoint operator+(int t);

friend TimePoint operator+(int t, TimePoint a);

TimePoint operator-(TimePoint a);

TimePoint operator-(int t);

friend TimePoint operator-(int t, TimePoint a);

bool operator<(TimePoint a);

bool operator<=(TimePoint a);

bool operator>(TimePoint a);

bool operator>=(TimePoint a);

bool operator!=(TimePoint a);

bool operator==(TimePoint a);

friend double gt(TimePoint a, TimePoint b); //во сколько раз больше

friend double lt(TimePoint a, TimePoint b); //во сколько раз меньше

};

TimePoint operator "" \_hrs(unsigned long long x){

return TimePoint(x\*60\*60);

}

TimePoint operator "" \_min(unsigned long long x){

return TimePoint(x\*60);

}

TimePoint operator "" \_sec(unsigned long long x){

return TimePoint(x);

}

//парсер строки

TimePoint parse(const char \*str, size\_t size){

int arr[3] = {0};

int cur = 0;

for(size\_t i = 0; i < size; i++){

if(str[i] == ':'){

cur++;

continue;

}

arr[cur] \*= 10;

arr[cur] += str[i]-'0';

}

if(cur != 2){

cout << "Неверный формат строки. Установлено время по-умолчанию 00:00:00" << endl;

cout << "Верный формат: ЧАСЫ:МИНУТЫ:СКУНДЫ" << endl;

return TimePoint(0);

}

return TimePoint(arr[0], arr[1], arr[2]);

}

//полный формат времени

TimePoint operator "" \_full(const char \*str, size\_t size){

return parse(str, size);

}

int TimePoint::time\_get(){

return time;

}

void TimePoint::time\_set(int t){

time = t;

}

void show(TimePoint a){

int temp = a.time;

if(temp < 0){

temp = -temp;

}

cout << temp/3600;

temp = temp%3600;

cout << ":" << temp/60;

temp = temp%60;

cout << ":" << temp << endl;

}

void TimePoint::operator=(TimePoint a){

this->time = a.time;

}

TimePoint TimePoint::operator+(TimePoint a){

TimePoint res(this->time + a.time);

return res;

}

TimePoint TimePoint::operator-(TimePoint a){

TimePoint res(this->time - a.time);

return res;

}

TimePoint TimePoint::operator+(int t){

TimePoint res(this->time + t);

return res;

}

TimePoint TimePoint::operator-(int t){

TimePoint res(this->time - t);

return res;

}

TimePoint operator+(int t, TimePoint a){

TimePoint res(t+a.time);

return res;

}

TimePoint operator-(int t, TimePoint a){

TimePoint res(t-a.time);

return res;

}

bool TimePoint::operator<(TimePoint a){

return(this->time < a.time);

}

bool TimePoint::operator>(TimePoint a){

return(this->time > a.time);

}

bool TimePoint::operator<=(TimePoint a){

return(this->time <= a.time);

}

bool TimePoint::operator>=(TimePoint a){

return(this->time >= a.time);

}

bool TimePoint::operator!=(TimePoint a){

return(this->time != a.time);

}

bool TimePoint::operator==(TimePoint a){

return(this->time == a.time);

}

double gt(TimePoint a, TimePoint b){

if(b.time == 0){

throw overflow\_error("Value of second argument is equal to 0. Can not compare.");

}

return static\_cast<double> (a.time) / b.time;

}

double lt(TimePoint a, TimePoint b){

if(a.time == 0){

throw overflow\_error("Value of first argument is equal to 0. Can not compare.");

}

return static\_cast<double> (b.time) / a.time;

}

int in\_minutes(int sec){

return sec/60;

}

int main(){

cout << "Следующие моменты времени созданы с помощью пользовательских";

cout << " литералов:" << endl;

cout << "----------------" << endl;

show("23:50:45"\_full);

show(20\_hrs);

show(100\_min);

show(999\_sec);

cout << "----------------" << endl;

vector<TimePoint\*> array;

int k = -1;

cout << "Доступные действия:" << endl;

cout << "0. Выход" << endl;

cout << "1. Создать объект" << endl;

cout << "2. Вывести список всех объектов" << endl;

cout << "3. Вывести время в стандартном формате" << endl;

cout << "4. Вывести время в минутах" << endl;

cout << "5. Вывести время в секундах" << endl;

cout << "6. Найти сумму моментов времени" << endl;

cout << "7. Найти разницу между моментами времени" << endl;

cout << "8. Прибавить ко времени заданное количество секунд" << endl;

cout << "9. Сравнить два объекта времени" << endl;

cout << "10. Во сколько раз один момент времени больше/меньше другого" << endl;

cout << "11. Вывести этот список" << endl;

while(k != 0){

begin:

cout << "Введите действие [0 - 11]: ";

cin >> k;

switch(k){

case 0:

return 0;

case 1:{

cout << "Введите время в стандартном формате: ЧАСЫ МИНУТЫ СЕКУНДЫ" << endl;

int a, b, c;

cin >> a >> b >> c;

TimePoint \*T = new TimePoint(a, b, c);

array.push\_back(T);

break;

}

case 2:{

cout << "Список объектов:" << endl;

for(int i = 0; i < array.size(); i++){

cout << i << ". ";

show(\*array[i]);

}

break;

}

case 3:{

if(array.size() == 0){

cout << "Ошибка: нет моментов времени" << endl;

break;

}

cout << "Доступные индексы: [" << 0 << " - " << array.size()-1 << "]" << endl;

cout << "Введите индекс: ";

int a;

cin >> a;

while(a < 0 || a >= array.size()){

cout << "Неверно введён индекс. Для выхода введите -1" << endl;

cin >> a;

if(a == -1){

goto begin;

}

}

cout << "Момент времени: ";

show(\*array[a]);

break;

}

case 4:{

if(array.size() == 0){

cout << "Ошибка: нет моментов времени" << endl;

break;

}

cout << "Доступные индексы: [" << 0 << " - " << array.size()-1 << "]" << endl;

cout << "Введите индекс: ";

int a;

cin >> a;

while(a < 0 || a >= array.size()){

cout << "Неверно введён индекс. Для выхода введите -1" << endl;

cin >> a;

if(a == -1){

goto begin;

}

}

cout << "Момент времени в минутах: ";

cout << in\_minutes(array[a]->time\_get());

cout << endl;

break;

}

case 5:{

if(array.size() == 0){

cout << "Ошибка: нет моментов времени" << endl;

break;

}

cout << "Доступные индексы: [" << 0 << " - " << array.size()-1 << "]" << endl;

cout << "Введите индекс: ";

int a;

cin >> a;

while(a < 0 || a >= array.size()){

cout << "Неверно введён индекс. Для выхода введите -1" << endl;

cin >> a;

if(a == -1){

goto begin;

}

}

cout << "Момент времени в секундах: ";

cout << array[a]->time\_get();

cout << endl;

break;

}

case 6:{

int k;

cout << "Выберите действие:" << endl;

cout << "1. Ввести два момента времени" << endl;

cout << "2. Выбрать два уже введенных момента времени" << endl;

cin >> k;

while(k < 1 || k > 2){

cout << "Неверно введено действие. Для выхода введите -1" << endl;

cin >> k;

if(k == -1){

goto begin;

}

}

if(k == 1){

cout << "Введите время в стандартном формате: ЧАСЫ МИНУТЫ СЕКУНДЫ" << endl;

int a, b, c;

cin >> a >> b >> c;

TimePoint t1(a, b, c);

cin >> a >> b >> c;

TimePoint t2(a, b, c);

show(t1+t2);

}else{

if(array.size() == 0){

cout << "Ошибка: нет моментов времени" << endl;

break;

}

cout << "Введите два индекса уже заведенных элементов.";

cout << "Доступные индексы: [" << 0 << " - " << array.size()-1 << "]" << endl;

int a, b;

cin >> a >> b;

if(a < 0 || a >= array.size() || b < 0 || b >= array.size()){

cout << "Неверно введён один из индексов. Для выхода введите -1 -1" << endl;

cin >> a >> b;

if(a == -1 && b == -1){

goto begin;

}

}

show(\*array[a]+\*array[b]);

}

break;

}

case 7:{

int k;

cout << "Выберите действие:" << endl;

cout << "1. Ввести два момента времени" << endl;

cout << "2. Выбрать два уже введенных момента времени" << endl;

cin >> k;

while(k < 1 || k > 2){

cout << "Неверно введено действие. Для выхода введите -1" << endl;

cin >> k;

if(k == -1){

goto begin;

}

}

if(k == 1){

cout << "Введите время в стандартном формате: ЧАСЫ МИНУТЫ СЕКУНДЫ" << endl;

int a, b, c;

cin >> a >> b >> c;

TimePoint t1(a, b, c);

cin >> a >> b >> c;

TimePoint t2(a, b, c);

show(t1-t2);

}else{

if(array.size() == 0){

cout << "Ошибка: нет моментов времени" << endl;

break;

}

cout << "Введите два индекса уже заведенных элементов.";

cout << "Доступные индексы: [" << 0 << " - " << array.size()-1 << "]" << endl;

int a, b;

cin >> a >> b;

if(a < 0 || a >= array.size() || b < 0 || b >= array.size()){

cout << "Неверно введён один из индексов. Для выхода введите -1 -1" << endl;

cin >> a >> b;

if(a == -1 && b == -1){

goto begin;

}

}

show(\*array[a]-\*array[b]);

}

break;

}

case 8:{

int k;

cout << "Выберите действие:" << endl;

cout << "1. Ввести момент времени" << endl;

cout << "2. Выбрать уже введенный момент времени" << endl;

cin >> k;

while(k < 1 || k > 2){

cout << "Неверно введено действие. Для выхода введите -1" << endl;

cin >> k;

if(k == -1){

goto begin;

}

}

if(k == 1){

cout << "Введите время в стандартном формате: ЧАСЫ МИНУТЫ СЕКУНДЫ" << endl;

int a, b, c;

cin >> a >> b >> c;

TimePoint t1(a, b, c);

cout << "Введите время в секундах: ";

int sec;

cin >> sec;

show(t1+sec);

}else{

if(array.size() == 0){

cout << "Ошибка: нет моментов времени" << endl;

break;

}

cout << "Введите индес уже заведенного элемента.";

cout << "Доступные индексы: [" << 0 << " - " << array.size()-1 << "]" << endl;

int a;

cin >> a;

if(a < 0 || a >= array.size()-1){

cout << "Неверно введён индекс. Для выхода введите -1 -1" << endl;

cin >> a;

if(a == -1){

goto begin;

}

}

cout << "Изменить элемент? y/n: ";

char c;

cin >> c;

cout << "Введите время в секундах: ";

int sec;

cin >> sec;

cout << "Результат: ";

if(c == 'y'){

array[a]->time\_set(array[a]->time\_get()+sec);

show(\*array[a]);

}else{

show(\*array[a]+sec);

}

}

break;

}

case 9:{

int k;

cout << "Выберите действие:" << endl;

cout << "1. Ввести моменты времени" << endl;

cout << "2. Выбрать уже введенные моменты времени" << endl;

cin >> k;

while(k < 1 || k > 2){

cout << "Неверно введено действие. Для выхода введите -1" << endl;

cin >> k;

if(k == -1){

goto begin;

}

}

if(k == 1){

cout << "Введите время в стандартном формате: ЧАСЫ МИНУТЫ СЕКУНДЫ" << endl;

int a, b, c;

cin >> a >> b >> c;

TimePoint t1(a, b, c);

cin >> a >> b >> c;

TimePoint t2(a, b, c);

if(t1 < t2){

cout << "Первый момент меньше второго" << endl;

}else if(t1 == t2){

cout << "Моменты времени равны" << endl;

}else{

cout << "Первый момент больше второго" << endl;

}

}else{

if(array.size() == 0){

cout << "Ошибка: нет моментов времени" << endl;

break;

}

cout << "Введите два индекса уже заведенных элементов.";

cout << "Доступные индексы: [" << 0 << " - " << array.size()-1 << "]" << endl;

int a, b;

cin >> a >> b;

if(a < 0 || a >= array.size() || b < 0 || b >= array.size()){

cout << "Неверно введён один из индексов. Для выхода введите -1 -1" << endl;

cin >> a >> b;

if(a == -1 && b == -1){

goto begin;

}

}

if(\*array[a] < \*array[b]){

cout << "Первый момент меньше второго" << endl;

}else if(\*array[a] == \*array[b]){

cout << "Моменты времени равны" << endl;

}else{

cout << "Первый момент больше второго" << endl;

}

}

break;

}

case 10:{

cout << "Больше или меньше? g/l: ";

char c;

cin >> c;

int k;

cout << "Выберите действие:" << endl;

cout << "1. Ввести два момента времени" << endl;

cout << "2. Выбрать два уже введенных момента времени" << endl;

cin >> k;

while(k < 1 || k > 2){

cout << "Неверно введено действие. Для выхода введите -1" << endl;

cin >> k;

if(k == -1){

goto begin;

}

}

if(k == 1){

cout << "Введите время в стандартном формате: ЧАСЫ МИНУТЫ СЕКУНДЫ" << endl;

int a, b, c;

cin >> a >> b >> c;

TimePoint t1(a, b, c);

cin >> a >> b >> c;

TimePoint t2(a, b, c);

if(c == 'g'){

cout << gt(t1, t2) << endl;

}else{

cout << lt(t1, t2) << endl;

}

}else{

if(array.size() == 0){

cout << "Ошибка: нет моментов времени" << endl;

break;

}

cout << "Введите два индекса уже заведенных элементов.";

cout << "Доступные индексы: [" << 0 << " - " << array.size()-1 << "]" << endl;

int a, b;

cin >> a >> b;

if(a < 0 || a >= array.size() || b < 0 || b >= array.size()){

cout << "Неверно введён один из индексов. Для выхода введите -1 -1" << endl;

cin >> a >> b;

if(a == -1 && b == -1){

goto begin;

}

}

if(c == 'g'){

cout << gt(\*array[a], \*array[b]) << endl;

}else{

cout << lt(\*array[a], \*array[b]) << endl;

}

}

break;

}

case 11:

cout << "Доступные действия:" << endl;

cout << "0. Выход" << endl;

cout << "1. Создать объект" << endl;

cout << "2. Вывести список всех объектов" << endl;

cout << "3. Вывести время в стандартном формате" << endl;

cout << "4. Вывести время в минутах" << endl;

cout << "5. Вывести время в секундах" << endl;

cout << "6. Найти сумму моментов времени" << endl;

cout << "7. Найти разницу между моментами времени" << endl;

cout << "8. Прибавить ко времени заданное количество секунд" << endl;

cout << "9. Сравнить два объекта времени" << endl;

cout << "10. Во сколько раз один момент времени больше/меньше другого" << endl;

cout << "11. Вывести этот список" << endl;

break;

default:

cout << "Неверно введено действие" << endl;

}

}

}

1. **Вывод**

Реализовал при помощи классов программу на языке С++ по работе с моментами времени. Используя перегрузку операторов и дружелюбные функции сделал возможным применение к объектам класса операторов сложения, вычитания и пр. в соответствии с заданием. Так же добавил возможность создания константных объектов класса при помощи пользовательских литералов. Корректность работы программы проверил на двух тестовых файлах.

**Список литературы**

1. Шилдт, Герберт. С++: базовый курс, 3-е изд. : Пер. с англ. - М. : ООО “И.Д. Вильямс”, 2018. - 624 с. : ил. - Парал. тит. англ.
2. Справочник по языку С++ [Электронный ресурс]. URL: <http://www.cplusplus.com/reference/deque/> (дата обращения: 17.09.2019).