Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

**ОТЧЕТ**

**Тема:** Лабораторная работа №3

Семестр: 2

Выполнил студент ИВТ-23-2б:

Синицын Владислав Сергеевич

(дата, подпись)

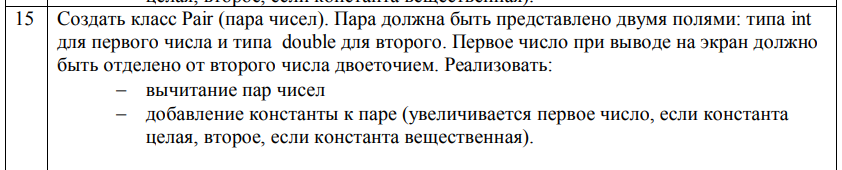
Проверила:

Полякова Ольга Андреевна

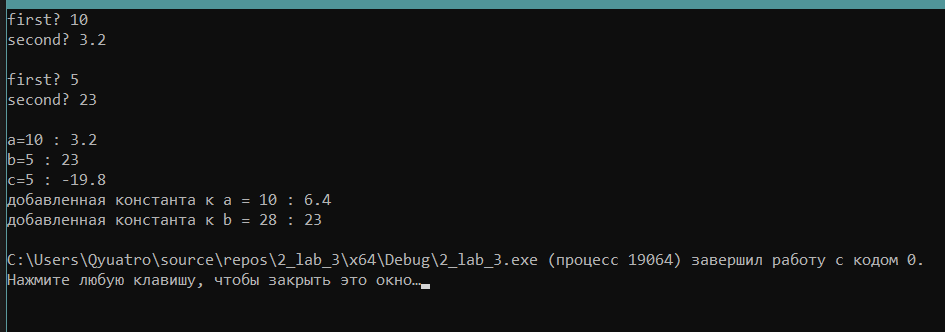
(дата, подпись)

Пермь 2024

Задание:



Программное решение:



***Main.cpp***

#include "Time.h"

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

system("chcp 1251 > NULL");

Time a;//конструктор без параметров

Time b; //конструктор без параметров

Time c; //конструктор без параметров

cin >> a;//ввод переменной

cin >> b;//ввод переменной

c = a - b;//сложение и постфиксная операция инкремент

cout << "a=" << a << endl; //вывод переменной

cout << "b=" << b << endl; //вывод переменной

cout << "c=" << c << endl; //вывод переменной

--a; //производим добавление константы

--b; //производим добавление константы

cout << "добавленная константа к a = " << a << endl; //вывод переменной

cout << "добавленная константа к b = " << b << endl; //вывод переменной

return 0;

}

***Time.cpp***

#include "Time.h"

#include <iostream>

using namespace std;

//перегрузка операции присваивания

Time& Time::operator=(const Time& t)

{

//проверка на самоприсваивание

if (&t == this) return \*this;

min = t.min;

sec = t.sec;

return \*this;

}

//перегрузка префиксной операции инкремент

Time& Time::operator++()

{

int temp = min \* 60 + sec;

temp++;

min = temp / 60;

sec = temp % 60;

return \*this;

}

//перегрузка постфиксной операции инкремент

Time Time::operator ++(int)

{

int temp = min \* 60 + sec;

temp++;

Time t(min, sec);

min = temp / 60;

sec = temp % 60;

return t;

}

//перегрузка префиксной операции инкремент

Time& Time::operator--()

{

if (int(sec) - sec == 0) {

min = min + sec;

sec = sec;

}

else {

min = min;

sec = sec + sec;

}

return \*this;

}

//перегрузка постфиксной операции инкремент

Time Time::operator--(int)

{

if (int(sec) - sec == 0) {

min = min + sec;

sec = sec;

}

else {

min = min;

sec = sec + sec;

}

Time t(min, sec);

min = min;

sec = sec;

return t;

}

//перегрузка бинарной операции сложения

Time Time::operator-(const Time& t)

{

double temp1 = min - t.min;

double temp2 = sec - t.sec;

Time p;

p.min = temp1;

p.sec = temp2;

return p;

}

//перегрузка глобальной функции-операции ввода

istream& operator>>(istream& in, Time& t)

{

cout << "first? "; in >> t.min;

cout << "second? "; in >> t.sec;

cout << endl;

return in;

}

//перегрузка глобальной функции-операции вывода

ostream& operator<<(ostream& out, const Time& t)

{

return (out << t.min << " : " << t.sec);

}

***Time.h***

#pragma once

//описание класса

#include <iostream>

using namespace std;

class Time

{

int min;

double sec;

public:

Time() { min = 0; sec = 0; };

Time(int m, double s) { min = m; sec = s; }

Time(const Time& t) { min = t.min; sec = t.sec; }

~Time() {};

int get\_min() { return min; }

double get\_sec() { return sec; }

void set\_min(int m) { min = m; }

void set\_sec(double s) { sec = s; }

//перегруженные операции

Time& operator=(const Time&);

Time& operator++();

Time operator++(int);//постфиксная операция

Time& operator--();

Time operator--(int);//постфиксная операция

Time operator-(const Time&);

//глобальные функции ввода-вывода

friend istream& operator>>(istream& in, Time& t);

friend ostream& operator<<(ostream& out, const Time& t);

};

