Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №18.3**

Дисциплина: «Информатика»

Тема: Объектно-ориентированное программирование. Перегрузка операций.

Вариант 4

Выполнил:

Студент группы РИС-20-1б

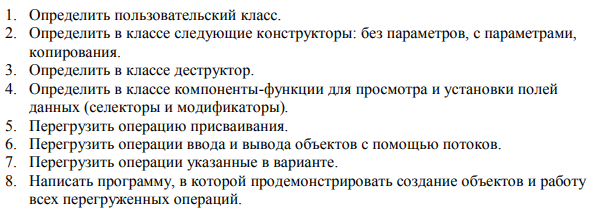
Еске Вячеслав Сергеевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Постановка задачи**



Создать класс Time для работы с временными интервалами. Интервал должен быть представлен в виде двух полей: минуты типа int и секунды типа int. при выводе минуты отделяются от секунд двоеточием. Реализовать:

 добавление секунд (учесть, что в минуте не может быть более 60 секунд)

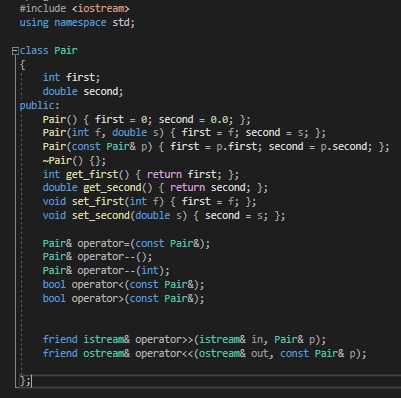
 сравнение временных интервалов (== и !=)

**Анализ задачи**

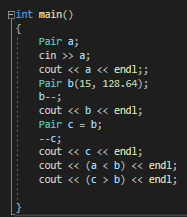
1. Для решения задачи необходимо…
   1. Организовать класс Pair, первое целое число и второе вещественное.



* 1. Организовать необходимые методы для ввода данных в поля first и second: гетторы, сетторы, конструкторы, деструктор, перегрузки операторов.



* 1. В функции main произвести тестирование.



1. В программе были использованы следующие типы данных:
   1. Тип данных int для хранения результата первого числа;



* 1. Тип данных double для хранения второго числа;



**Код**

Main.cpp

#include "Time.h"

#include <iostream>

using namespace std;

void main() {

setlocale(LC\_ALL, "Rus");

Time a;//конструктор без параметров

Time b; //конструктор без параметров

cout << "a: "<<endl;

cin>>a;//ввод переменной

cout << "b: "<<endl;

cin>>b;//ввод переменной

if (a == b) cout << "Временные промежутки равны" << endl;

if (a != b) cout << "Временные промежутки неравны" << endl;

cout << "Префиксный инкримент:" << endl;

++a;//префиксная операция инкремент

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

cout << "Постфиксный инкримент:" << endl;

b++;//сложение и постфиксная операция инкремент

cout << "b: " << b << endl;

if (a == b) cout << "Временные промежутки равны" << endl;

if (a != b) cout << "Временные промежутки неравны" << endl;

}

Time.h

#pragma once

#include <iostream>

using namespace std;

class Time {

int min, sec;

public:

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

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

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

~Time() {};

int get\_min() { return min; }

int get\_sec() { return sec; }

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

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

Time&operator=(const Time&);

Time& operator++();

Time operator++(int);

friend bool operator==(const Time& t1, const Time& t2);

friend bool operator!=(const Time& t1, const Time& t2);

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

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

};

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 глобальной функции-операции ввода

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

cout << "Введите минуты - ";

in >> t.min;

do {

cout << "Введите секунды - ";

in >> t.sec;

} while ((t.sec < 0) || (t.sec >= 60));

return in;

}

//перегрузка глобальной функции-операции вывода

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

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

}

//перегрузка функции сравнения

bool operator==(const Time& t1, const Time& t2) {

if (t1.min == t2.min && t1.sec == t2.sec)

return true;

else return false;

}

//перегрузка функции неравенства

bool operator!=(const Time& t1, const Time& t2) {

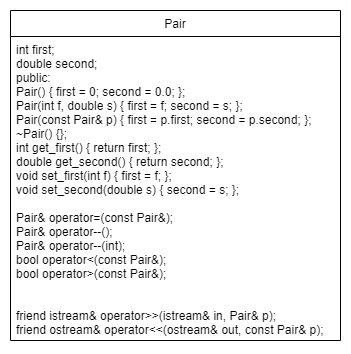
if (t1.min != t2.min && t1.sec != t2.sec)

return true;

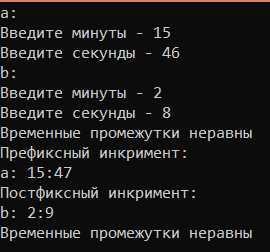
else return false;

}

**UML диаграмма**

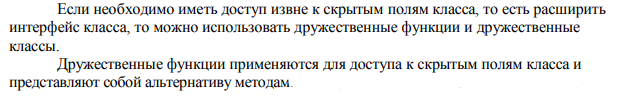


**Скриншоты тестов**

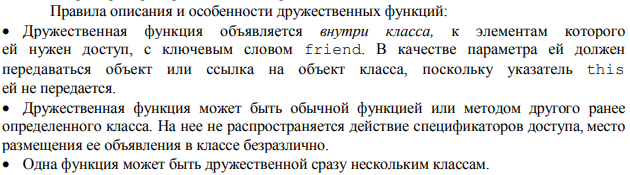


**Контрольные вопросы**



1. 

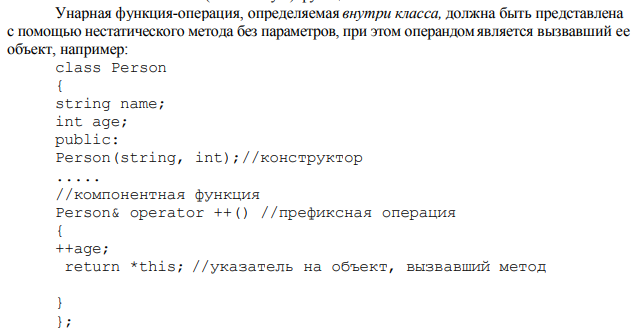


2. 

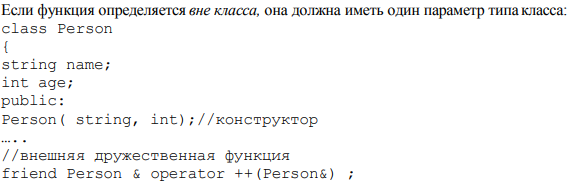


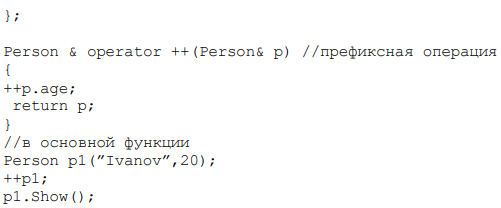
3. 



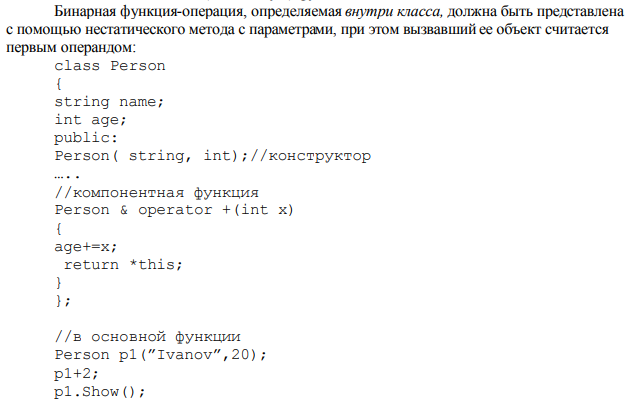
4.



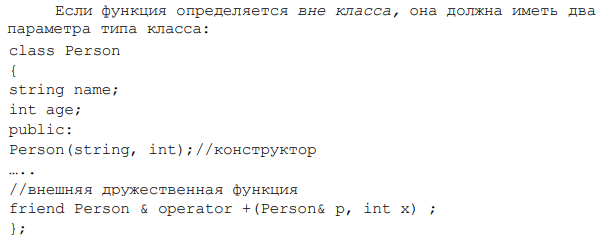
5.





6.

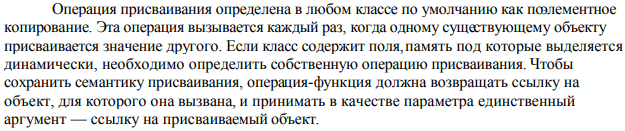


7. 



8. 

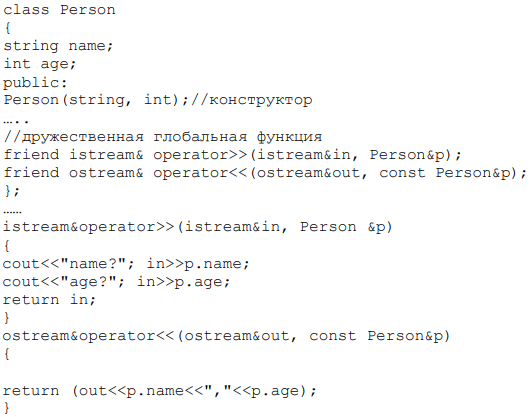


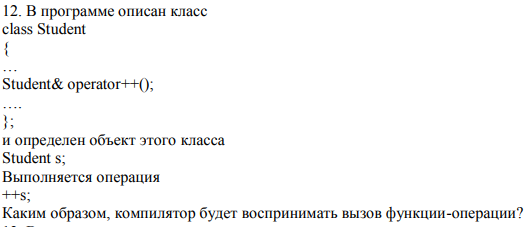
9. 



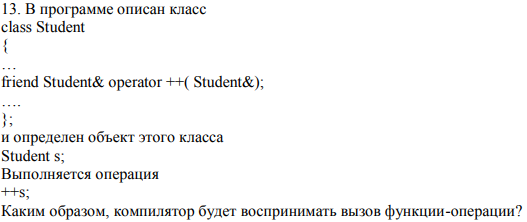
10. Ссылку на объект, для которого операция была вызвана.



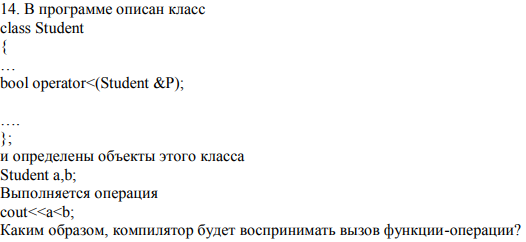




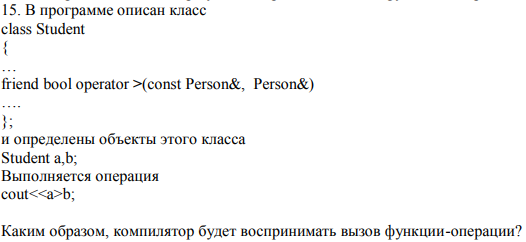
12. Будет осуществлён вызов функции перегрузки оператора префиксного инкремента.



13. Будет осуществлён вызов функции перегрузки оператора постфиксного инкремента.



14. Будет осуществлён вызов функции перегрузки оператора сравнения (<), затем через cout будет выведен результат сравнения (1 или 0).



15. Будет осуществлён вызов функции перегрузки оператора сравнения (>), затем через cout будет выведен результат сравнения (1 или 0).