**Пермский национальный исследовательский политехнический университет**

Кафедра “Информационные технологии и автоматизированные системы”

**Лабораторная работа №22**

«По дисциплине «Основы алгоритмизации и программирования»

**Тема:**

Перегрузка операций.

Выполнила:

студентка группы ИВТ-19-1б

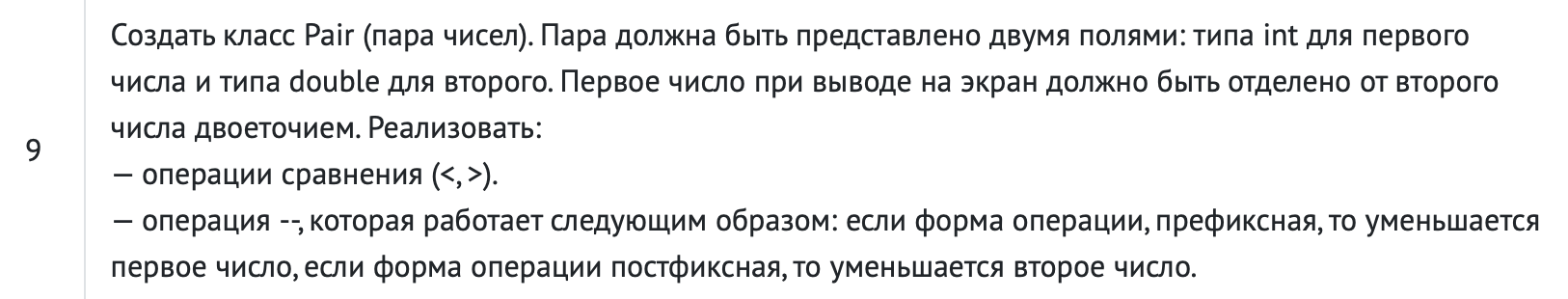
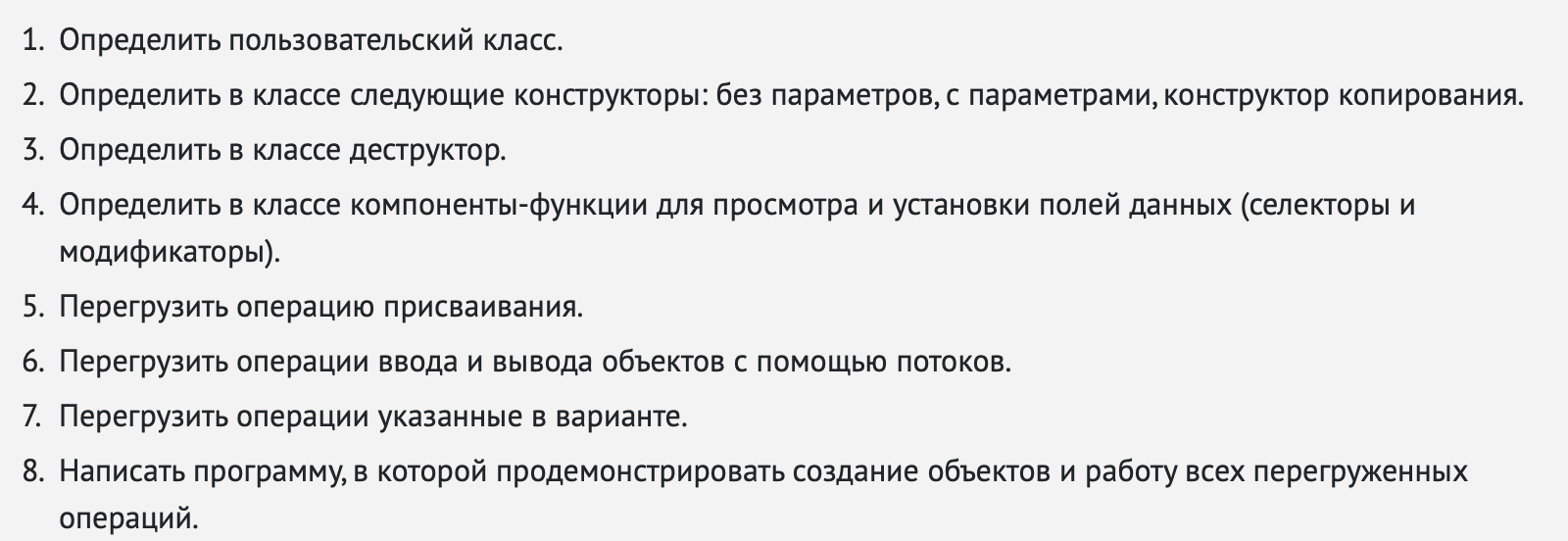
Еремеева Полина Алексеевна

Проверила:

доцент кафедры “ИТАС”

Полякова О.А.

Пермь, 2020

**Постановка задачи**

**Код**

Main.cpp

#include <iostream>

#include "Pair.hpp"

**using** **namespace** std;

**int** main() {

setlocale(LC\_ALL, "ru");

Pair a;

Pair b;

Pair c;

cout << "Ввод переменной a" << endl;

cin >> a;

cout << "Ввод переменной b" << endl;

cin >> b;

a = --a;

cout << "a = " << a;

b = b--;

cout << "b = " << b;

cout << "a < b?\t";

c = a < b;

cout << "b > a?\t";

c = b > a;

}

Pair.cpp

#include "Pair.hpp"

**using** **namespace** std;

Pair& Pair::**operator**=(**const** Pair& t) {

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

first = t.first;

second = t.second;

**return** \***this**;

}

Pair& Pair::**operator** --() {

**this**->first--;

**return** \***this**;

}

Pair Pair::**operator**--(**int**) {

Pair temp(\***this**);

**this**->second--;

**return** temp;

}

Pair Pair::**operator** <(**const** Pair& t) {

**if** (**this**->first < t.first && **this**->second < t.second) {

cout << "Да " << endl;

**return** \***this**;

}

**else** {

cout << "Нет " << endl;

**return** t;

}

}

Pair Pair::**operator** >(**const** Pair& t) {

**if** (**this**->first > t.first && **this**->second > t.second)

{

cout << "Да " << endl;

**return** \***this**;

}

**else** {

cout << "Нет " << endl;

**return** t;

}

}

istream& **operator**>>(istream& in, Pair& t) {

cout << "first: "; cin >> t.first;

cout << "second: "; cin >> t.second;

**return** in;

}

ostream& **operator**<<(ostream& out, **const** Pair& t) {

**return**(out<<t.first << " : " << t.second << endl);

}

Pair.hpp

#include <iostream>

**using** **namespace** std;

**class** Pair {

**int** first;

**double** second;

**public**:

Pair() { first = 0; second = 0; };

Pair(**int** f, **double** s) { first = f; second = s; }

Pair(**const** Pair& t) { first = t.first; second = t.second; }

~Pair() {};

**int** get\_first() { **return** first; }

**double** get\_second() { **return** second; }

**void** set\_first (**int** f) { first = f; }

**void** set\_second (**double** s) { second = s; }

Pair& **operator**=(**const** Pair&);

Pair& **operator**--();

Pair **operator**--(**int**);

Pair **operator**>(**const** Pair&);

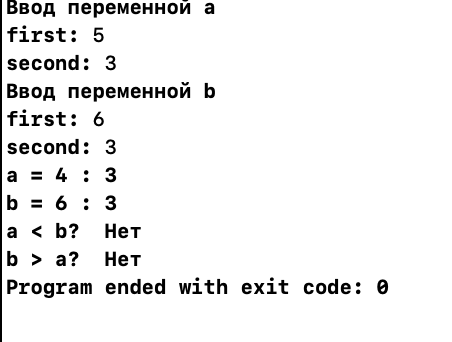
Pair **operator**<(**const** Pair&);

**friend** istream& **operator**>>(istream& in, Pair& t);

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

};

**Тестирование**

****