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

**Пермский национальный исследовательский политехнический университет**

Факультет Электротехнический Кафедра ИТАС

Специальность Промышленная Робототехника

**ОТЧЁТ**

**о лабораторной работе №10**

Сохранение данных в файле с использованием потоков

|  |  |
| --- | --- |
|  | Выполнил:  Студент группы ПРТ-21-1Б  Торган Г.А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Проверил:  Доцент кафедры ИТАС Полякова О.А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Пермь 2022**

**Постановка задачи:**

Создать класс Pair (пара чисел). Пара должна быть представлено двумя полями: типа int

для первого числа и типа double для второго. Первое число при выводе на экран должно

быть отделено от второго числа двоеточием. Реализовать:

 операции сравнения (==,!=).

 вычитание константы из пары (уменьшается первое число, если константа

целая, второе, если константа вещественная).

Задание:

 Удалить все записи из интервала от k1 до k2, где k1 и k2 переменные типа Pair.

 Увеличить все записи с заданным значением в два раза.

 Добавить K записей в начало файла

**Текст программы:**

**Lab3.cpp**

#include "stdafx.h"

#include "Header.h"

#include <iostream>

double const con = 15;

class Pair

{

private:

int \_a;

double \_b;

public:

Pair()

: \_a(0), \_b(0) { }

Pair(int a, double b)

: \_a(a), \_b(b) { }

~Pair() { }

Pair& operator-- () {

\_a=\_a-con;

return \*this;

}

Pair& operator-- (int) {

\_b=\_b-con;

return \*this;

}

friend bool operator == (const Pair&, const Pair&);

friend bool operator != (const Pair&, const Pair&);

friend std::ostream& operator<< (std::ostream&, const Pair&);

};

bool operator == (const Pair& lhs, const Pair& rhs) {

return (lhs.\_a == rhs.\_a) && (lhs.\_b == lhs.\_b);

}

bool operator != (const Pair& lhs, const Pair& rhs) {

return (lhs.\_a != rhs.\_a) && (lhs.\_b != lhs.\_b);

}

std::ostream& operator<< (std::ostream& ost, const Pair& obj) {

return (ost << obj.\_a << " : " << obj.\_b);

}

int main()

{

//double const con = 15.5;

Pair p(20, 20.50);

std::cout << "P -> " << p << "\n\n";

int pr = con;

std::cout << "Const. = " << con << std::endl;

if (pr == con)

{

--p; std::cout << "P - Const. -> " << p << std::endl;

}

else

{

p--; std::cout << "P - Const. -> " << p << std::endl;

}

std::cout << std::endl;

Pair p2(5, 20.50);

std::cout << "P2 -> " << p2 << "\n\n";

if (p == p2)

std::cout << "P == P2 \n";

else

std::cout << "P != P2 \n";

system("PAUSE");

return 0;

}

**Header.h**

#pragma once

double const cons = 15.5;

**stdafx.h**

#pragma once

#include "targetver.h"

#include <stdio.h>

#include <tchar.h>

**Fraction.cpp**

#include <iostream>

#include "fraction.h"

using namespace std;

//реализация метода для инициализации полей структуры

void fraction::Init(int F, double S)

{

first = F; second = S;

}

//реализация метода для чтения значений полей структуры

void fraction::Read()

{

cout << "\nfirst?"; cin >> first;

cout << "\nsecond?"; cin >> second;

}

//реализация метода для вывода значений полей структуры

void fraction::Show()

{

cout << "\nfirst=" << first;

cout << "\nsecond=" << second;

cout << "\n";

}

//метод для возведения в степень

double fraction:: Multiply(int k)

{

double cel = first;

cout << "\nK?"; cin >> k;

return (k\*(cel+second));

}

**Ответ для варианта №14**

