**Папин А.В., ИУ5Ц-41Б.**

**РК-2. Перегрузка операции.**

**First.cpp (основной заголовочный файл для выполнения заданий)**

// RK\_25032022.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

#include <iostream>

#include <cstring>

#include "Rect.h"

#pragma warning(disable : 4996)

using namespace std;

int main()

{

system("chcp 1251 > nul");

cout << "Пункт №2" << endl;

Rect T1(1, 1);

// Перегрузка 1-ой арифмеичской операции

cout << "T1: " << T1 << endl;

//T1--;

cout << "T1: " << T1-- << endl;

//--T1;

cout << "T1: " << --T1 << endl;

// Перегрузка 2-ой арифмеичской операции

Rect T2(2, 2);

cout << "T2: " << T2 << endl;

cout << "T2 \* 10: " << T2 \* 10 << endl;

// Перегрузка операции сравнения

// S1 = 10; S2 = 20;

// 20 > 10 -> S2 > S1;

Rect T3(10, 1), T31(4, 5);

cout << "T3: " << T3 << endl;

cout << "T31: " << T31 << endl;

if (T3 < T31)

cout << "T31 больше\n" << endl;

else

cout << "T3 больше\n" << endl;

// Перегрузка операции потокового ввода и вывода

Rect T4;

cout << "Введите значения для Т4" << endl;

cin >> T4;

cout << T4;

system("pause");

}

**Rect.cpp (Вспомогательный заголовочный файл для класса)**

#include "Rect.h"

#include <iostream>

void Rect::set\_a(int \_a) { a = \_a; }

void Rect::set\_b(int \_b) { b = \_b; }

int Rect::get\_a() { return a; }

int Rect::get\_b() { return b; }

**Rect.h (Заголовочный файл класса)**

#pragma once

#pragma warning(disable : 4996)

#include <iostream>

using namespace std;

class Rect

{

private:

int a;

int b;

public:

Rect() { a = NULL; b = NULL; }

Rect(int \_a, int \_b) { a = \_a; b = \_b; }

~Rect() {}

Rect(const Rect& \_Rect) { \*this = \_Rect; }

Rect(Rect& obj) { a = obj.a; b = obj.b; };

void set\_a(int \_year\_work);

void set\_b(int \_salary);

int get\_a();

int get\_b();

// Внешняя - дружественная функция

// Постфиксная операция

friend Rect& operator--(Rect& OBJ, int VALUE) {

Rect tmp = OBJ;

OBJ.a--;

OBJ.b--;

return tmp;

}

// Префиксная операция

friend Rect& operator--(Rect& OBJ) {

--OBJ.a;

--OBJ.b;

return OBJ;

}

// Оператор умножения

Rect& operator\* (const int value) {

Rect tmp;

tmp.a = this->a \* value;

tmp.b = this->b \* value;

return tmp;

}

// Оператор сравнения

bool operator< (Rect OBJ) {

if (OBJ.a < this->a)

return true;

else

return false;

}

// Операция ввода и вывода

friend ostream& operator<< (ostream& stream, Rect k) {

stream << "a = " << k.a << ", " << "b = " << k.b << endl;

return stream;

}

friend istream& operator>>(istream& in, Rect& obj) {

cout << "a: ";

in >> obj.a;

cout << "b: ";

in >> obj.b;

return in;

}

};