Московский авиационный институт

(национальный исследовательский университет)

**Факультет информационных технологий и прикладной математики**

Кафедра вычислительной математики и программирования

**Лабораторная работа 0 по курсу ОOП:**

**основы программирования на языке С#**

СОЗДАНИЕ КЛАССА НА ЯЗЫКЕ C#

Работу выполнил:

Березнев Никита Вадимович

Группа: М8О-203Б-21

Подпись: \_\_\_\_\_\_\_\_

Дата: 29 октября 2022 г.

**Код программы:**

using System;

namespace CSharpLab0

{

class Reestr

{

public Reestr()

{

Console.WriteLine("Overloading constructor Reestr");

}

public Reestr(string f, int num, int old, int year)

{

this.f = f;

this.num = num;

this.old = old;

this.year = year;

}

public void print()

{

Console.Write("{0}\t", f);

Console.Write("{0}\t", num);

Console.Write("{0}\t", old);

Console.Write("{0}\t", year);

Console.WriteLine();

}

public void print(int num)

{

Console.WriteLine("Overloading operation 'print'");

Console.WriteLine("Value of parameter: {0}", num);

}

public int CurrentYear()

{

int CY = old + year;

return CY;

}

public int CurrentYear(int N)

{

int CY = old + year;

return CY + N;

}

public static int operator +(Reestr a, Reestr b)

{

return a.year + b.year;

}

string f { set; get; }

int num, old, year;

}

class Program

{

static void Main()

{

Reestr a = new Reestr("Homski", 1, 1950, 55);

Reestr b = new Reestr("Ulman", 2, 1990, 15);

Reestr c = new Reestr("Lindmar", 3, 1970, 35);

Console.WriteLine("Using constructor Reestr with parameters");

Console.WriteLine("and operation 'print' without parameters:");

Console.WriteLine();

a.print();

b.print();

c.print();

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.ReadKey();

Reestr d = new Reestr();

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.ReadKey();

int D = 30;

d.print(D);

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.ReadKey();

Console.WriteLine("Function CurrentYear without parameters:");

Console.WriteLine("Current year: {0}", a.CurrentYear());

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.ReadKey();

Console.WriteLine("Overloading function CurrentYear:");

Console.WriteLine("Current year + 30: {0}", a.CurrentYear(D));

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.ReadKey();

Console.WriteLine("Operator overload: {0}", a + b);

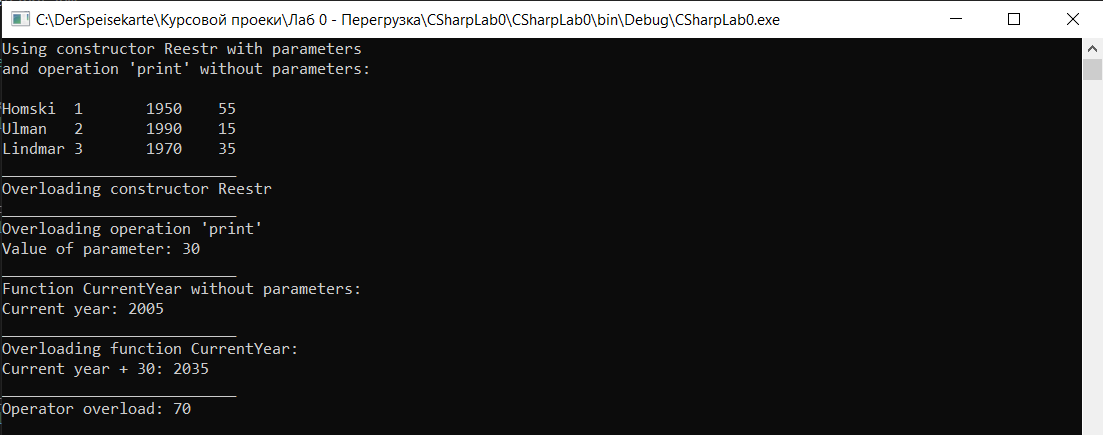
Console.ReadKey();

}

}

}

**Результат:**



**Вывод:**

Перегрузка конструкторов, функций, операторов и операций позволяет определить структурный полиморфизм, когда с данным именем существуют несколько функций с разным поведением, а также с различным списком аргументов, который определит, какая функция будет вызываться.