Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра Информатики

Дисциплина «Программирование»

**ОТЧЕТ**

к лабораторной работе №7

на тему:

**«ПЕРЕГРУЗКА ОПЕРАТОРОВ»**

БГУИР 6-05-0612-02 05

|  |
| --- |
| Выполнила студент группы 353504  АНТОНОВА Лидия Сергеевна |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 5.** Класс полином ax+by+cz. Поля – int (a, b, c). Перегрузить +, -, ++, --, \* на число, / на число. Сравнить на == и !=. (d) если a=b=c=0, полином = false. Преобразовать в число (a) и назад (ax) – в обоих случаях явно.

# 2 Выполнение работы

Класс Polynomial в C# представляет собой полином вида ax + by + cz. Он содержит три поля типа int: A, B и C, которые представляют коэффициенты полинома. Конструктор класса принимает три целочисленных аргумента и инициализирует поля A, B и C. В классе перегружены операторы для выполнения математических операций над полиномами, таких как сложение, вычитание, умножение и деление на число, а также для сравнения полиномов на равенство и неравенство. Класс также предоставляет возможность преобразования полинома в число и обратно, а также проверки полинома на “истинность” и “ложность”. Ниже представлен листинг кода.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

public class Polynomial

{

public int A { get; set; }

public int B { get; set; }

public int C { get; set; }

public Polynomial(int a, int b, int c)

{

A = a;

B = b;

C = c;

}

public static Polynomial operator +(Polynomial p1, Polynomial p2)

{

return new Polynomial(p1.A + p2.A, p1.B + p2.B, p1.C + p2.C);

}

public static Polynomial operator -(Polynomial p1, Polynomial p2)

{

return new Polynomial(p1.A - p2.A, p1.B - p2.B, p1.C - p2.C);

}

public static Polynomial operator ++(Polynomial p)

{

return new Polynomial(p.A + 1, p.B + 1, p.C + 1);

}

public static Polynomial operator --(Polynomial p)

{

return new Polynomial(p.A - 1, p.B - 1, p.C - 1);

}

public static Polynomial operator \*(Polynomial p, int num)

{

return new Polynomial(p.A \* num, p.B \* num, p.C \* num);

}

public static Polynomial operator /(Polynomial p, int num)

{

return new Polynomial(p.A / num, p.B / num, p.C / num);

}

public static bool operator ==(Polynomial p1, Polynomial p2)

{

return p1.A == p2.A && p1.B == p2.B && p1.C == p2.C;

}

public static bool operator !=(Polynomial p1, Polynomial p2)

{

return !(p1 == p2);

}

public static explicit operator int(Polynomial p)

{

return p.A;

}

public static explicit operator Polynomial(int num)

{

return new Polynomial(num, 0, 0);

}

public static bool operator true(Polynomial p)

{

return p.A != 0 || p.B != 0 || p.C != 0;

}

public static bool operator false(Polynomial p)

{

return p.A == 0 && p.B == 0 && p.C == 0;

}

public override string ToString()

{

return $"{A}x + {B}y + {C}z";

}

public int this[int index]

{

get

{

switch (index)

{

case 0: return A;

case 1: return B;

case 2: return C;

default: throw new IndexOutOfRangeException("Index must be between 0 and 2");

}

}

set

{

switch (index)

{

case 0: A = value; break;

case 1: B = value; break;

case 2: C = value; break;

default: throw new IndexOutOfRangeException("Index must be between 0 and 2");

}

}

}

}

}

Программа демонстрирует использование класса Polynomial. В методе main создаются два объекта класса Polynomial, выполняются различные операции над ними и выводятся результаты. Это позволяет продемонстрировать работу всех методов и операций, определенных в классе Polynomial. Результаты операций выводятся на консоль, что позволяет увидеть, как работает класс Polynomial, и убедиться в корректности его работы. Ниже представлен листинг кода.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

Polynomial p1 = new Polynomial(1, 2, 3);

Polynomial p2 = new Polynomial(4, 5, 6);

Console.WriteLine("p1: " + p1);

Console.WriteLine("p2: " + p2);

Polynomial p3 = p1 + p2;

Console.WriteLine("p1 + p2: " + p3);

Polynomial p4 = p1 - p2;

Console.WriteLine("p1 - p2: " + p4);

Polynomial p5 = p1 \* 2;

Console.WriteLine("p1 \* 2: " + p5);

Polynomial p6 = p1 / 2;

Console.WriteLine("p1 / 2: " + p6);

Console.WriteLine("p1 == p2: " + (p1 == p2));

Console.WriteLine("p1 != p2: " + (p1 != p2));

Console.WriteLine("p1 is true: " + (p1 ? "true" : "false"));

int a = (int)p1;

Console.WriteLine("p1 to int: " + a);

Polynomial p7 = (Polynomial)a;

Console.WriteLine("int to Polynomial: " + p7);

}

}

}

# Вывод

В ходе выполнения данной лабораторной работы были изучены и применены ключевые принципы перегрузки операторов в C#.