МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования” Гомельский государственный университет им Ф. Скорины”

**ЛАБОРАТОРНАЯ РАБОТА КФ-Программирование-1-1-08 ПЕРЕГРУЗКА ОПЕРАЦИЙ В КЛАССАХ НА ЯЗЫКЕ C#**

Выполнил:

Студент группы КФ-17

Гуревич П.А.

Проверил Дей Е.А.

Гомель 2022

**Цель работы:** изучение правил реализации и применения перегрузки операций в классах на языке C#.

**ЗАДАЧА 1.** Создайте новый проект и повторите реализацию класса «Комплексные числа».

Выполните программу несколько раз, изменяя значения чисел.

Дополните проект «Комплексные числа» перегрузкой операций вычитания и умножения двух комплексных чисел.

Реализуйте вызов этих операций в функции Main() и вывод их результатов на экран.

Результаты включите в отчет.

**ЗАДАЧА 2.** В проекте, содержащем собственный класс, созданный в соответствии с вариантом, реализуйте перегрузку операций сложения объектов, вычитания объектов и умножения объектов на вещественное число справа и слева.

Смысл и реализацию операций, для конкретного класса обсудите с преподавателем.

Переопределение каждой операции пояснить комментарием.

В функции Main() реализуйте вывод на экран параметров исходных объектов и результатов выполнения каждой переопределенной операции.

Результаты включите в отчет.

**ЗАДАЧА 3.** В проекте, содержащем собственный класс, созданный в соответствии с вариантом (см. предыдущую лабораторную работу), реализуйте перегрузку всех операций сравнения объектов класса по значению выбранного параметра.

Реализацию операций обсудите с преподавателем.

В функции Main() реализуйте вывод на экран параметров исходных объектов и результатов выполнения каждой операции сравнения.

Результаты включите в отчет.

**ЗАДАЧА 1.** Создайте новый проект и повторите реализацию класса «Комплексные числа».

Выполните программу несколько раз, изменяя значения чисел.

Дополните проект «Комплексные числа» перегрузкой операций вычитания и умножения двух комплексных чисел.

Реализуйте вызов этих операций в функции Main() и вывод их результатов на экран.

class Complex

{

public double re { get; set; } //--вещественная часть

public double im { get; set; } //--мнимая часть

public Complex(double re, double im) //--конструктор

{ this.re = re; this.im = im; }

public double Modul()

//--метод вычисления модуля

{ return Math.Sqrt(re \* re + im \* im); }

public override string ToString() //--преобразование в строку

{

if (im >= 0) return " " + re + "+i" + im + " ";

else return " " + re + "-i" + Math.Abs(im) + " ";

}

public void Info() //--вывод информации о числе

{

if (im >= 0) Console.WriteLine("Комплексное число {0:f3}+i{1:f3}", re, im);

else Console.WriteLine("Комплексное число {0:f3}-i{1:f3}", re, Math.Abs(im));

Console.WriteLine(" модуль = {0:f3}", Modul());

}

public static Complex operator +(Complex a, Complex b)

{

Complex rezult = new Complex(0, 0);

rezult.re = a.re + b.re;

rezult.im = a.im + b.im;

return rezult;

}

public static Complex operator \*(Complex a, double b)

//--операция умножения комплексного числа на вещественное

{

Complex rezult = new Complex(0, 0);

rezult.re = a.re \* b;

rezult.im = a.im \* b;

return rezult;

}

public static Complex operator \*(double a, Complex b)

//--операция умножения вещественного числа на комплексное

{

Complex rezult = new Complex(0, 0);

rezult.re = a \* b.re;

rezult.im = a \* b.im;

return rezult;

}

public static bool operator >(Complex a, Complex b)

//--операция сравнения двух комплексных чисел по модулю

{

return a.Modul() > b.Modul();

}

public static bool operator <(Complex a, Complex b)

//--операция сравнения двух комплексных чисел по модулю

{

return a.Modul() < b.Modul();

}

static void Main(string[] args)

{

Complex p1 = new Complex(1, 2);

Complex p2 = new Complex(2, 3);

Complex p4 = new Complex(2, 4);

Complex p5 = new Complex(1, 1);

Complex p6 = new Complex(2, 2);

Complex i = p1 + p2;

Complex j = 2 \* p4;

bool k = p5 > p6;

i.Info();

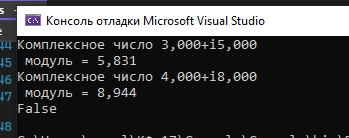
j.Info();

Console.WriteLine(k);

}

}

Результат:



**ЗАДАЧА 2.** В проекте, содержащем собственный класс, созданный в соответствии с вариантом, реализуйте перегрузку операций сложения объектов, вычитания объектов и умножения объектов на вещественное число справа и слева.

Смысл и реализацию операций, для конкретного класса обсудите с преподавателем.

Переопределение каждой операции пояснить комментарием.

В функции Main() реализуйте вывод на экран параметров исходных объектов и результатов выполнения каждой переопределенной операции.

Результаты включите в отчет.

using System;

using System.Numerics;

namespace ConsoleApplication6

{

class Vector

{

private int x2, y2, x1, y1; double Q; //--поля

public int A

{ set { x2 = value; } get { return x2; } } //--свойства

public int B

{ set { y2 = value; } get { return y2; } }//--свойства

public int Y

{ set { x1 = value; } get { return x1; } } //--свойства

public int N

{ set { y1 = value; } get { return y1; } }//--свойства

public double C

{ set { Q = value; } get { return Q; } }//--свойства

public Vector()//--пустой конструктор

{

Q = 0; x2 = 0; y2 = 0; x1 = 0; y1 = 0;

}

public Vector(double Q)//--Q в начале координат

{

this.Q = Q; this.x2 = 0; this.y2 = 0; this.x1 = 0; this.y1 = 0;

}

public Vector(int x2, int x1, int y1, int y2)//--единичный заряд в точке x1, y1, x2, y2

{

this.x2 = x2; this.y2 = y2; Q = 1.602176; this.x1 = x1; this.y1 = y1;

}

public Vector(double Q, int x2, int x1, int y1, int y2) //--общий случай

{

this.Q = Q; this.x2 = x2; this.x1 = x1; this.y1 = y1; this.y2 = y2;

}

public double Distanc()//--расстояние

{ return Math.Sqrt(Math.Pow((x2 - x1), 2) + Math.Pow((y2 - y1), 2)); }

public void Poten()//--потенциал в точке (0;0)

{ Console.WriteLine("Несуществует потенциала в точке "); }

public double Poten1() //потенциал в точке (x,y)

{ return (1 / (4 \* Math.PI) \* Q) / Math.Sqrt(Math.Pow((x2 - x1), 2) + Math.Pow((y2 - y1), 2)); }

public double Ox()//проекция на x

{ return Math.Sqrt(Math.Pow((x2 - x1), 2)); }

public double Oy()//проекция на y

{ return Math.Sqrt(Math.Pow((y2 - y1), 2)); }

public void Info()

{

Console.WriteLine("Значение Q = {0}, Значение х2 = {1}, Значение х1 = {2}, Значение y1 = {3}, Значение y2 = {4}, ",Q,x2,x1,y1,y2);

}

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

{

Vector rezult = new Vector(0, 0, 0, 0, 0);

rezult.Q = a.Q + b;

rezult.x1 = a.x1 + b;

rezult.y1 = a.y1 + b;

rezult.x2 = a.x2 + b;

rezult.y2 = a.y2 + b;

return rezult;

}

public static Vector operator \*(Vector a, int b)

{

Vector rezult = new Vector(0, 0, 0, 0, 0);

rezult.Q = a.Q \* b;

rezult.x1 = a.x1 \* b;

rezult.y1 = a.y1 \* b;

rezult.x2 = a.x2 \* b;

rezult.y2 = a.y2 \* b;

return rezult;

}

public static Vector operator -(Vector a, int b)

{

Vector rezult = new Vector(0, 0, 0, 0, 0);

rezult.Q = a.Q - b;

rezult.x1 = a.x1 - b;

rezult.y1 = a.y1 - b;

rezult.x2 = a.x2 - b;

rezult.y2 = a.y2 - b;

return rezult;

}

}

class Program

{

static void Main(string[] args)

{

Vector p = new Vector(2, 6, 23, 2, 6);

Console.WriteLine("Растояние => " + p.Distanc());

Console.WriteLine("Потенциал в точке x,y => " + p.Poten1());

Console.WriteLine("Проекция на ось Ox => " + p.Ox());

Console.WriteLine("Проекция на ось Oy => " + p.Oy());

Console.WriteLine("\n");

Console.WriteLine("Перегрузка");

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

Vector r = new Vector();

r = p1 + 2;

Console.WriteLine("Перегрузка символа +");

r.Info();

Vector p2 = new Vector(1, 4, 6, 2, 0);

Vector j = new Vector();

j = p2 \* 4;

Console.WriteLine("Перегрузка символа \*");

j.Info();

Vector p3 = new Vector(3, 4, 4, 7, 9);

Vector g = new Vector();

g = p3 - 2;

Console.WriteLine("Перегрузка символа -");

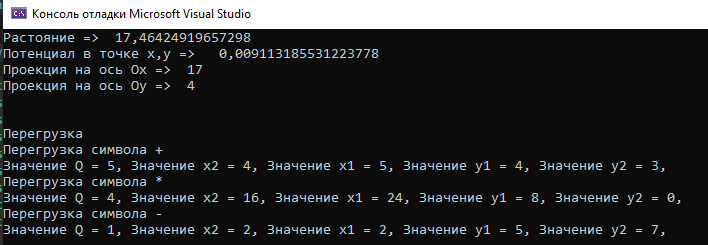
g.Info();

}

}

}

Результат:



**ЗАДАЧА 3.** В проекте, содержащем собственный класс, созданный в соответствии с вариантом (см. предыдущую лабораторную работу), реализуйте перегрузку всех операций сравнения объектов класса по значению выбранного параметра.

Реализацию операций обсудите с преподавателем.

В функции Main() реализуйте вывод на экран параметров исходных объектов и результатов выполнения каждой операции сравнения.

Результаты включите в отчет.

using System;

using System.Numerics;

namespace ConsoleApplication6

{

class Vector

{

private int x2, y2, x1, y1; double Q; //--поля

public int A

{ set { x2 = value; } get { return x2; } } //--свойства

public int B

{ set { y2 = value; } get { return y2; } }//--свойства

public int Y

{ set { x1 = value; } get { return x1; } } //--свойства

public int N

{ set { y1 = value; } get { return y1; } }//--свойства

public double C

{ set { Q = value; } get { return Q; } }//--свойства

public Vector()//--пустой конструктор

{

Q = 0; x2 = 0; y2 = 0; x1 = 0; y1 = 0;

}

public Vector(double Q)//--Q в начале координат

{

this.Q = Q; this.x2 = 0; this.y2 = 0; this.x1 = 0; this.y1 = 0;

}

public Vector(int x2, int x1, int y1, int y2)//--единичный заряд в точке x1, y1, x2, y2

{

this.x2 = x2; this.y2 = y2; Q = 1.602176; this.x1 = x1; this.y1 = y1;

}

public Vector(double Q, int x2, int x1, int y1, int y2) //--общий случай

{

this.Q = Q; this.x2 = x2; this.x1 = x1; this.y1 = y1; this.y2 = y2;

}

public double Distanc()//--расстояние

{ return Math.Sqrt(Math.Pow((x2 - x1), 2) + Math.Pow((y2 - y1), 2)); }

public void Poten()//--потенциал в точке (0;0)

{ Console.WriteLine("Несуществует потенциала в точке "); }

public double Poten1() //потенциал в точке (x,y)

{ return (1 / (4 \* Math.PI) \* Q) / Math.Sqrt(Math.Pow((x2 - x1), 2) + Math.Pow((y2 - y1), 2)); }

public double Ox()//проекция на x

{ return Math.Sqrt(Math.Pow((x2 - x1), 2)); }

public double Oy()//проекция на y

{ return Math.Sqrt(Math.Pow((y2 - y1), 2)); }

public void Info()

{

Console.WriteLine("Значение Q = {0}, Значение х2 = {1}, Значение х1 = {2}, Значение y1 = {3}, Значение y2 = {4}, ",Q,x2,x1,y1,y2);

}

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

{

Vector rezult = new Vector(0, 0, 0, 0, 0);

rezult.Q = a.Q + b;

rezult.x1 = a.x1 + b;

rezult.y1 = a.y1 + b;

rezult.x2 = a.x2 + b;

rezult.y2 = a.y2 + b;

return rezult;

}

public static Vector operator \* (Vector a, int b)

{

Vector rezult = new Vector(0, 0, 0, 0, 0);

rezult.Q = a.Q \* b;

rezult.x1 = a.x1 \* b;

rezult.y1 = a.y1 \* b;

rezult.x2 = a.x2 \* b;

rezult.y2 = a.y2 \* b;

return rezult;

}

public static Vector operator - (Vector a, int b)

{

Vector rezult = new Vector(0, 0, 0, 0, 0);

rezult.Q = a.Q - b;

rezult.x1 = a.x1 - b;

rezult.y1 = a.y1 - b;

rezult.x2 = a.x2 - b;

rezult.y2 = a.y2 - b;

return rezult;

}

public static bool operator > (Vector a, Vector b)

{

return a.Poten1() > b.Poten1();

}

public static bool operator <(Vector a, Vector b)

{

return a.Poten1() < b.Poten1();

}

}

class Program

{

static void Main(string[] args)

{

Vector p = new Vector(2, 6, 23, 2, 6);

Console.WriteLine("Растояние => " + p.Distanc());

Console.WriteLine("Потенциал в точке x,y => " + p.Poten1());

Console.WriteLine("Проекция на ось Ox => " + p.Ox());

Console.WriteLine("Проекция на ось Oy => " + p.Oy());

Console.WriteLine("\n");

Console.WriteLine("Перегрузка");

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

Vector r = new Vector();

r = p1 + 2;

Console.WriteLine("Перегрузка символа +");

r.Info();

Vector p2 = new Vector(1, 4, 6, 2, 0);

Vector j = new Vector();

j = p2 \* 4;

Console.WriteLine("Перегрузка символа \*");

j.Info();

Vector p3 = new Vector(3, 4, 4, 7, 9);

Vector g = new Vector();

g = p3 - 2;

Console.WriteLine("Перегрузка символа -");

g.Info();

Vector p4 = new Vector(4, 4, 4, 7, 9);

Vector p5 = new Vector(5, 7, 8, 9, 10);

Vector f = new Vector();

bool x = p4 < p5;

Console.WriteLine(x);

}

}

}

Результат:

