

Особенности перегрузки оператора != в языке C#

Оператор != (не равно) является оператором сравнения. Его особенность, как и других операторов сравнения, заключается в попарной перегрузке. Оператор != перегружается вместе с оператором ==. При перегрузке только одного из операторов компилятор выдаст ошибку. При перегрузке операторов == и != необходимо также перегрузить виртуальные методы Equals и GetHashCode. В противном случае компилятор выдаст предупреждение.

Пример программы перегрузки:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace RK1
{
    class Class1
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Quadrante 1 size 5");
            Quadrante Q1 = new Quadrante(5);

            Console.WriteLine("Quadrante 2 size 3");
            Quadrante Q2 = new Quadrante(3);

            Console.WriteLine("Q1 == Q2 ");
            if (Q1 == Q2)
            {
                Console.WriteLine(" True ");
            }
            else
            {
                Console.WriteLine(" False ");
            }

            Console.WriteLine("Q1 != Q2 ");
            if (Q1 != Q2)
            {
                Console.WriteLine(" True ");
            }
            else
            {
                Console.WriteLine(" False ");
            }

            Console.WriteLine("Q1 Equal Q2 ");
            if (Q1.Equals(Q2))
            {
                Console.WriteLine(" True ");
            }
            else
```

```

    {
        Console.WriteLine(" False ");
    }

    Console.WriteLine("HashCode Q1 {0}", Q1.GetHashCode());

    Console.ReadKey();
}

class Quadrate
{
    public Quadrate(int a)
    {
        this.Lenght = a;
    }
    public int Lenght;

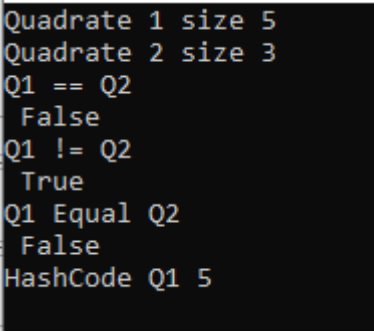
    public static bool operator ==(Quadrate Q1, Quadrate Q2)
    {
        return Q1.Lenght == Q2.Lenght;
    }

    public static bool operator !=(Quadrate Q1, Quadrate Q2)
    {
        return !(Q1 == Q2);
    }

    public override bool Equals(object obj)
    {
        Quadrate Q1 = (Quadrate)obj; return this == Q1;
    }
    public override int GetHashCode()
    {
        return this.Lenght.GetHashCode();
    }
}
}

```

Вывод:



```

Quadrate 1 size 5
Quadrate 2 size 3
Q1 == Q2
False
Q1 != Q2
True
Q1 Equal Q2
False
HashCode Q1 5

```

Сообщение об ошибке компилятора при попытке перегрузки только одного оператора сравнения

```

Ссылка: 7
class Quadrate
{
    Ссылка: 2
    public Quadrate(int a)
    {
        this.Lenght = a;
    }
    public int Lenght;

    Ссылка: 1
    public static bool operator ==(Quadrate Q1, Quadrate Q2)
    {
        return Q1.Lenght == Q2.Lenght;
    }
    /*
    public static bool operator !=(Quadrate Q1, Quadrate Q2)
    {
        return !(Q1 == Q2);
    }
}

```

bool Quadrate.operator ==(Quadrate Q1, Quadrate Q2)
CS0216: Для оператора "Class1.Quadrate.operator ==(Class1.Quadrate, Class1.Quadrate)" требуется, чтобы был определен соответствующий оператор "!=".

```

Ссылка: 7
class Quadrate
{
    Ссылка: 2
    public Quadrate(int a)
    {
        this.Lenght = a;
    }
    public int Lenght;
    /*
    public static bool operator ==(Quadrate Q1, Quadrate Q2)
    {
        return Q1.Lenght == Q2.Lenght;
    }
}

Ссылка: 1
public static bool operator !=(Quadrate Q1, Quadrate Q2)
{
    return !(Q1 == Q2);
}

```

bool Quadrate.operator !=(Quadrate Q1, Quadrate Q2)
CS0216: Для оператора "Class1.Quadrate.operator !=(Class1.Quadrate, Class1.Quadrate)" требуется, чтобы был определен соответствующий оператор "==".

Предупреждения компилятора о необходимости перегрузки методов

```

Ссылка: 9
class Quadrate
{
    Ссылка: 1
    public
    {
        CS0660: "Class1.Quadrate" определяет оператор "==" или оператор "!=", но не переопределяет Object.Equals(object o).
        CS0661: "Class1.Quadrate" определяет оператор "==" или оператор "!=", но не переопределяет Object.GetHashCode().
        Показать возможные решения (Alt+ВВОДилиCtrl+.)
    }
    Ссылка: 2
    public static bool operator ==(Quadrate Q1, Quadrate Q2)
    {
        return Q1.Lenght == Q2.Lenght;
    }
    Ссылка: 1
    public static bool operator !=(Quadrate Q1, Quadrate Q2)
    {
        return !(Q1 == Q2);
    }
    /*
    public override bool Equals(object obj)
    {
        Quadrate Q1 = (Quadrate)obj; return this == Q1;
    }
    public override int GetHashCode()
    {
        return this.Lenght.GetHashCode();
    }
}
}

```