**Дробь**

Проведите рефакторинг класса Ratio. В результате:

* Numerator, Denominator и Value должны остаться полями класса Ratio.
* После создания объекта Ratio не должно быть возможности его изменить, то есть поменять поля Numerator, Denominator или Value.
* После создания объекта Ratio знаменатель всегда должен быть больше нуля. Бросайте исключение ArgumentException при попытке установить неверное значение знаменателя.

public static void Check(int num, int den)

{

var ratio = new Ratio(num, den);

Console.WriteLine("{0}/{1} = {2}",

ratio.Numerator, ratio.Denominator,

ratio.Value.ToString(CultureInfo.InvariantCulture));

}

public class Ratio

{

public Ratio(int num, int den)

{

...

}

public int Numerator; //числитель

public int Denominator; //знаменатель

public double Value; //значение дроби Numerator / Denominator

}

**Программа:**

using System;

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace umop14o12zRatio

{

class Program

{

static void Main(string[] args)

{

Check(5, 11);

Console.ReadKey();

}

public static void Check(int num, int den)

{

var ratio = new Ratio(num, den);

Console.WriteLine("{0}/{1} = {2}",

ratio.Numerator, ratio.Denominator,

ratio.Value.ToString(CultureInfo.InvariantCulture));

}

}

public class Ratio

{

public Ratio(int num, int den)

{

Numerator = num;

Denominator = den;

Value = (double)num / den;

if (den <= 0) throw new ArgumentException();

}

public readonly int Numerator; //числитель

public readonly int Denominator; //знаменатель

public readonly double Value; //значение дроби Numerator / Denominator

}

}