**Ratio**

Refactor the Ratio class. As a result:

• Numerator, Denominator and Value must remain fields of the Ratio class.

• After creating the Ratio object, it should not be possible to change it, that is, to change the Numerator, Denominator or Value fields.

• After creating a Ratio object, the denominator must always be greater than zero. Throw an ArgumentException when trying to set an invalid denominator value.

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;

public int Denominator;

public double Value;

}

**Code:**

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;

public readonly int Denominator;

public readonly double Value;

}

}