Rational.Divide(Rational, Rational) Method

名前空間: WS.Theia.ExtremelyPrecise
アセンブリ: ExtremelyPrecise.dll
一方の Rational 値をもう一方の値で除算し、その結果を返します。

```
public static WS.Theia.ExtremelyPrecise.Rational
Add(WS.Theia.ExtremelyPrecise.Rational dividend,
WS.Theia.ExtremelyPrecise.Rational divisor);
```

パラメーター
dividend　Rational
被除数。
divisor　Rational
除数。

戻り値
Rational
除算の結果。

例

次の例では Rational 配列の各要素に、Divide メソッド、除算演算子（/）。及び DivRem メソッドを使用している。

```
using System;
using WS.Theia.ExtremelyPrecise;
```

```
public class Example
{
    public static void Main()
    {
        Rational divisor = Math.Pow(Int64.MaxValue, 2);

        Rational [] dividends = { Rational.Multiply((Rational) Single.MaxValue,
2),

Rational.Parse("906123451238755090918275600071000 99"),
                                  Rational.One,
                                  Rational.Multiply(Int32.MaxValue,
Int64.MaxValue),
                                  divisor + Rational.One };
        // Divide each dividend by divisor in three different ways.
        foreach (Rational dividend in dividends)
        {
            Rational quotient;
            Rational remainder = 0;

            Console.WriteLine("Dividend: {0:N0}", dividend);
            Console.WriteLine("Divisor:   {0:N0}", divisor);
            Console.WriteLine("Results:");
            Console.WriteLine("     Using Divide method:       {0:N0}",
                              Rational.Divide(dividend, divisor));
            Console.WriteLine("     Using Division operator: {0:N0}",
                              dividend / divisor);
            (quotient, remainder)=Math.DivRem(dividend,divisor);
            Console.WriteLine("     Using DivRem method:       {0:N0},
remainder {1:N0}",
                              quotient, remainder);

            Console.WriteLine();
        }
    }
}
```

```
// The example displays the following output:
//       Dividend: 680,564,693,277,057,719,623,408,366,969,033,850,880
//       Divisor:  85,070,591,730,234,615,847,396,907,784,232,501,249
//       Results:
//           Using Divide method:        7
//           Using Division operator: 7
//           Using DivRem method:        7, remainder
85,070,551,165,415,408,691,630,012,479,406,342,137
//
//       Dividend: 90,612,345,123,875,509,091,827,560,007,100,099
//       Divisor:  85,070,591,730,234,615,847,396,907,784,232,501,249
//       Results:
//           Using Divide method:        0
//           Using Division operator: 0
//           Using DivRem method:        0, remainder
90,612,345,123,875,509,091,827,560,007,100,099
//
//       Dividend: 1
//       Divisor:  85,070,591,730,234,615,847,396,907,784,232,501,249
//       Results:
//           Using Divide method:        0
//           Using Division operator: 0
//           Using DivRem method:        0, remainder 1
//
//       Dividend: 19,807,040,619,342,712,359,383,728,129
//       Divisor:  85,070,591,730,234,615,847,396,907,784,232,501,249
//       Results:
//           Using Divide method:        0
//           Using Division operator: 0
//           Using DivRem method:        0, remainder
19,807,040,619,342,712,359,383,728,129
//
//       Dividend: 85,070,591,730,234,615,847,396,907,784,232,501,250
//       Divisor:  85,070,591,730,234,615,847,396,907,784,232,501,249
//       Results:
//           Using Divide method:        1
```

```
//          Using Division operator: 1
//          Using DivRem method:     1, remainder 1
```

注釈

演算子のオーバーロードや、カスタム演算子をサポートしない言語用の、Rational 値を割り算する代替メソッドです。

適用対象

.NET Core
2.0
.NET Framework
4.6.1
.NET Standard
2.0
UWP
10.0.16299
Xamarin.Android
8.0
Xamarin.iOS
10.14
Xamarin.Mac
3.8