# Math.IEEERemainder(Rational,Rational) Method

名前空間: WS.Theia.ExtremelyPrecise

アセンブリ: ExtremelyPrecise.dll

指定した数を別の指定数で除算した結果の剰余を返します。

public static WS.Theia.ExtremelyPrecise.Rational Floor(WS.Theia.ExtremelyPrecise.Rational value);

## パラメーター

dividend　Rational

被除数。

Divisor　Rational

除数。

## 戻り値

Rational  
dividend - (divisor Q) に等しい数値。Q は dividend / divisor の商を丸めた近似整数を示します。dividend / divisor が 2 つの整数の中間に位置する場合は、偶数の整数が返されます。

dividend - (divisor Q) がゼロのとき、dividend が正である場合は値 +0、dividend が負である場合は -0 が返されます。

divisor = 0 の場合は、NaN が返されます。

# 例

次の例ではIEEERemainder(Rational)メソッドと剰余演算子で返される値の比較をしています。

using System;  
using WS.Theia.ExtremelyPrecise;  
  
public class Example  
{  
 public static void Main()  
 {  
 Console.WriteLine($"{"IEEERemainder",35} {"Remainder operator",20}");  
 ShowRemainders(3, 2);  
 ShowRemainders(4, 2);  
 ShowRemainders(10, 3);  
 ShowRemainders(11, 3);  
 ShowRemainders(27, 4);  
 ShowRemainders(28, 5);  
 ShowRemainders(17.8, 4);  
 ShowRemainders(17.8, 4.1);  
 ShowRemainders(-16.3, 4.1);  
 ShowRemainders(17.8, -4.1);  
 ShowRemainders(-17.8, -4.1);  
 }  
  
 private static void ShowRemainders(double number1, double number2)  
 {  
 var formula = $"{number1} / {number2} = ";  
 var ieeeRemainder = Math.IEEERemainder(number1, number2);  
 var remainder = number1 % number2;  
 Console.WriteLine($"{formula,-16} {ieeeRemainder,18} {remainder,20}");  
 }  
}  
// The example displays the following output:  
//   
//  
// IEEERemainder Remainder operator  
// 3 / 2 = -1 1  
// 4 / 2 = 0 0  
// 10 / 3 = 1 1  
// 11 / 3 = -1 2  
// 27 / 4 = -1 3  
// 28 / 5 = -2 3  
// 17.8 / 4 = 1.8 1.8  
// 17.8 / 4.1 = 1.4 1.4  
// -16.3 / 4.1 = 0.0999999999999979 -4  
// 17.8 / -4.1 = 1.4 1.4  
// -17.8 / -4.1 = -1.4 -1.4

# 注釈

この操作は、ANSI/IEEE Std 754-1985; のセクション 5.1 で定義されている剰余演算に準拠しています。

# 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