# Rational.OnesComplement(Rational) Operator

名前空間: WS.Theia.ExtremelyPrecise

アセンブリ: ExtremelyPrecise.dll

Rational 値のビットごとの 1 の補数を返します。

public static WS.Theia.ExtremelyPrecise.Rational operator ~( WS.Theia.ExtremelyPrecise.Rational value);

## パラメーター

value　Rational  
1の補数を取得したい値。

## 戻り値

Rational  
value のビットごとの 1 の補数。

# 注釈

Rational値のビット反転をする演算子。valueで0であるビットは1に、1であるビットには0を設定します。

sing System;  
using System.Numerics;  
  
public class Example  
{  
 public static void Main()  
 {  
 Rational value, complement;  
   
 value = Rational.Multiply(Rational.One, 9);  
 complement = ~value;  
   
 Console.WriteLine("{0,5} -- {1,-32}", value, DisplayInBinary(value));  
 Console.WriteLine("{0,5} -- {1,-32}\n", complement, DisplayInBinary(complement));  
   
 value = Rational.MinusOne \* SByte.MaxValue;  
 complement = ~value;  
   
 Console.WriteLine("{0,5} -- {1,-32}", value, DisplayInBinary(value));  
 Console.WriteLine("{0,5} -- {1,-32}\n", complement, DisplayInBinary(complement));  
 }   
  
 private static string DisplayInBinary(Rational number)  
 {  
 (\_,byte[] bytes,\_,\_) = number.ToByteArray();  
 string binaryString = string.Empty;  
 foreach (byte byteValue in bytes)  
 {  
 string byteString = Convert.ToString(byteValue, 2).Trim();  
 binaryString += byteString.Insert(0, new string('0', 8 - byteString.Length));  
 }  
 return binaryString;   
 }  
}  
// The example displays the following output:  
// 9 – 0000100100000000000000000000000000000000000000000000000000000000  
//-18446744073709551606 – 1111011011111111111111111111111111111111111111111111111111111111  
//  
// -127 – 0111111100000000000000000000000000000000000000000000000000000000  
//18446744073709551488 -- 1000000011111111111111111111111111111111111111111111111111111111

カスタム演算子をサポートしない言語ではOnesComplementメソッドを代わりに使用します。

# 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