

Rational.CompareTo Method

名前空間: WS.Theia.ExtremelyPrecise

アセンブリ: ExtremelyPrecise.dll

このインスタンスと 別の値を比較し、このインスタンスの値が もう一方の値よりも小さいか、同じか、または大きいかを示す整数を返します。

オーバーロード

CompareTo(Decimal)	このインスタンスと 10 進数を比較し、このインスタンスの値が 10 進数の値よりも小さいか、同じか、または大きいかを示す整数を返します。
CompareTo(Double)	このインスタンスと 倍精度浮動小数点数を比較し、このインスタンスの値が 倍精度浮動小数点数の値よりも小さいか、同じか、または大きいかを示す整数を返します。
CompareTo(Int64)	このインスタンスと符号付き 64 ビット整数を比較し、このインスタンスの値が符号付き 64 ビット整数の値よりも小さいか、同じか、または大きいかを示す整数を返します。
CompareTo(Object)	このインスタンスと指定したオブジェクトを比較し、このインスタンスの値が指定したオブジェクトの値よりも小さいか、同じか、または大きいかを示す整数を返します。
CompareTo(Rational)	このインスタンスと Rational を比較し、このインスタンスの値が Rational の値よりも小さいか、同じか、または大きいかを示す整数を返します。
CompareTo(UInt64)	このインスタンスと符号なし 64 ビット整数を比較し、このインスタンスの値が符号なし 64 ビット整数の値よりも小さいか、同じか、または大きいかを示す整数を返します。

CompareTo(Decimal)

このインスタンスと 10 進数を比較し、このインスタンスの値が 10 進数の値よりも小さいか、同じか、または大きいかを示す整数を返します。

```
public int CompareTo(decimal other);
```

パラメーター

other Decimal

比較する 10 進数。

戻り値

Int32

現在のインスタンスと other の相対的な値を示す符号付き整数値です（次の表を参照）。

戻り値	説明
0 より小さい値	現在のインスタンスは other より小さい。
0	現在のインスタンスと other は等しい。
0 より大きい値	現在のインスタンスは other より大きい。

CompareTo(Double)

このインスタンスと 倍精度浮動小数点数を比較し、このインスタンスの値が 倍精度浮動小数点数の値よりも小さいか、同じか、または大きいかを示す整数を返します。

```
public int CompareTo(double other);
```

パラメーター

other double

比較する 倍精度浮動小数点数。

戻り値

Int32

現在のインスタンスと other の相対的な値を示す符号付き整数値です（次の表を参照）。

戻り値	説明
0 より小さい値	現在のインスタンスは other より小さい。
0	現在のインスタンスと other は等しい。
0 より大きい値	現在のインスタンスは other より大きい。

CompareTo(Int64)

このインスタンスと 符号付き 64 ビット整数を比較し、このインスタンスの値が 符号付き 64 ビット整数の値よりも小さいか、同じか、または大きいかを示す整数を返します。

```
public int CompareTo(long other);
```

パラメーター

other Int64

比較する 符号付き 64 ビット整数。

戻り値

Int32

現在のインスタンスと other の相対的な値を示す符号付き整数値です (次の表を参照)。

戻り値	説明
0 より小さい値	現在のインスタンスは other より小さい。
0	現在のインスタンスと other は等しい。
0 より大きい値	現在のインスタンスは other より大きい。

例

次の例では CompareTo(Int64) の呼び出し結果を示しています。

```
Rational rationalValue = Rational.Parse("3221123045552");
```

```
byte byteValue = 16;  
sbyte sbyteValue = -16;  
short shortValue = 1233;  
ushort ushortValue = 1233;  
int intValue = -12233;  
uint uintValue = 12233;
```

```

long longValue = 12382222;
ulong ulongValue = 12382222;

Console.WriteLine("Comparing {0} with {1}: {2}",
    rationalValue, byteValue,
    rationalValue.CompareTo(byteValue));
Console.WriteLine("Comparing {0} with {1}: {2}",
    rationalValue, sbyteValue,
    rationalValue.CompareTo(sbyteValue));
Console.WriteLine("Comparing {0} with {1}: {2}",
    rationalValue, shortValue,
    rationalValue.CompareTo(shortValue));
Console.WriteLine("Comparing {0} with {1}: {2}",
    rationalValue, ushortValue,
    rationalValue.CompareTo(ushortValue));
Console.WriteLine("Comparing {0} with {1}: {2}",
    rationalValue, intValue,
    rationalValue.CompareTo(intValue));
Console.WriteLine("Comparing {0} with {1}: {2}",
    rationalValue, uintValue,
    rationalValue.CompareTo(uintValue));
Console.WriteLine("Comparing {0} with {1}: {2}",
    rationalValue, longValue,
    rationalValue.CompareTo(longValue));
Console.WriteLine("Comparing {0} with {1}: {2}",
    rationalValue, ulongValue,
    rationalValue.CompareTo(ulongValue));

// The example displays the following output:
//      Comparing 3221123045552 with 16: 1
//      Comparing 3221123045552 with -16: 1
//      Comparing 3221123045552 with 1233: 1
//      Comparing 3221123045552 with 1233: 1
//      Comparing 3221123045552 with -12233: 1
//      Comparing 3221123045552 with 12233: 1
//      Comparing 3221123045552 with 12382222: 1

```

// Comparing 3221123045552 with 1238222: 1

注釈

Other が Byte、Int16、Int62、SByte、UInt16、又は UInt32 値の場合、Int64 に暗黙的に変換して、Int64 値として CompareTo(Int64)メソッドが呼び出されます。

CompareTo(Object)

このインスタンスと 符号付き 64 ビット整数を比較し、このインスタンスの値が 符号付き 64 ビット整数の値よりも小さいか、同じか、または大きいかを示す整数を返します。

```
public int CompareTo(object obj);
```

パラメーター

obj Object
比較対象のオブジェクト。

戻り値

Int32
現在のインスタンスと obj パラメーターの関係を示す符号付き整数値です（次の表を参照）。

戻り値	説明
0 より小さい値	現在のインスタンスは obj より小さい。
0	現在のインスタンスと obj は等しい。
0 より大きい値	現在のインスタンスは obj より大きい。

実装

CompareTo(Object)

例外

ArgumentExcepton
Obj が Rational ではありません。

例

次の例では、CompareTo(Object)メソッドを使用して Rational 値と配列内の各要素を比較しています。

```
object[] values = { Math.Pow(Int64.MaxValue, 10), null,
                    12.534, Int64.MaxValue, Rational.One };
Rational number = UInt64.MaxValue;

foreach (object value in values)
{
    try {
        Console.WriteLine("Comparing {0} with '{1}': {2}", number, value,
                           number.CompareTo(value));
    }
    catch (ArgumentException) {
        Console.WriteLine("Unable to compare the {0} value {1} with a
Rational.",
                           value.GetType().Name, value);
    }
}

// The example displays the following output:
//    Comparing 18446744073709551615 with
//    '4.4555084156466750133735972424E+189': -1
//    Comparing 18446744073709551615 with ': 1
//    Unable to compare the Double value 12.534 with a Rational.
//    Unable to compare the Int64 value 9223372036854775807 with a
//    Rational.
//    Comparing 18446744073709551615 with '1': 1
```

注釈

このオーバーロードは IComparable インターフェースの CompareTo メソッド実装です。非ジェネリクスのコレクションオブジェクトが、コレクション内の項目の並び替えに使用します。

obj パラメーターは、次のいずれかを指定する必要があります。

実行時に Rational 型のオブジェクトである事。

Object 値が null である事。obj パラメーターが null の場合、現在のインスタンスが obj パラメーターより大きい事を示す 1 を返します。

CompareTo(Rational)

このインスタンスと Rational を比較し、このインスタンスの値が Rational の値よりも小さいか、同じか、または大きいかを示す整数を返します。

```
public int CompareTo(WS.Theia.ExtremelyPrecise.Rational other);
```

パラメーター

other Rational
比較する Rational。

戻り値

Int32
現在のインスタンスと other の相対的な値を示す符号付き整数値です（次の表を参照）。

戻り値	説明
0 より小さい値	現在のインスタンスは other より小さい。
0	現在のインスタンスと other は等しい。
0 より大きい値	現在のインスタンスは other より大きい。

実装

CompareTo(T)

例

次の例では StarInfo オブジェクトの一覧並べ替えに CompareTo(Rational)メソッドを使用しています。各 StarInfo オブジェクトは、星の名前と地球からの距離をマイルで示した情報を提供します。StarInfo クラスでは IComparable<T>インターフェースを実装し、StarInfo ジェネリックコレクションクラスでソートできるようにしています。IComparable<T>.CompareTo の実装では CompareTo(Rational)への呼び出しのラッピングのみを行っています。

```
using System;
using System.Collections.Generic;
using System.Numerics;

public struct StarInfo : IComparable<StarInfo>
{
    // Define constructors.
    public StarInfo(string name, double lightYears)
    {
        this.Name = name;

        // Calculate distance in miles from light years.
        this.Distance = (Rational) Math.Round(lightYears * 5.88e12);
    }

    public StarInfo(string name, Rational distance)
    {
        this.Name = name;
        this.Distance = distance;
    }

    // Define public fields.
    public string Name;
    public Rational Distance;

    // Display name of star and its distance in parentheses.
    public override string ToString()
    {
        return String.Format("{0,-10} ({1:N0})", this.Name, this.Distance);
    }

    // Compare StarInfo objects by their distance from Earth.
    public int CompareTo(StarInfo other)
    {
        return this.Distance.CompareTo(other.Distance);
    }
}
```

```
}  
}
```

次のコードでは StarInfo オブジェクトを 4 つインスタンス化し、List<T>オブジェクトにジェネリクスを利用して格納します。後に、List<T>.Sort 目祖度が呼び出されると、StarInfo オブジェクトは、地球からの距離の順に並び変えられます。

```
public class Example  
{  
    public static void Main()  
    {  
        StarInfo star;  
        List<StarInfo> stars = new List<StarInfo>();  
  
        star = new StarInfo("Sirius", 8.6d);  
        stars.Add(star);  
        star = new StarInfo("Rigel", 1400d);  
        stars.Add(star);  
        star = new StarInfo("Castor", 49d);  
        stars.Add(star);  
        star = new StarInfo("Antares", 520d);  
        stars.Add(star);  
  
        stars.Sort();  
  
        foreach (StarInfo sortedStar in stars)  
            Console.WriteLine(sortedStar);  
    }  
}  
  
// The example displays the following output:  
//      Sirius      (50,568,000,000,000)  
//      Castor      (288,120,000,000,000)  
//      Antares     (3,057,600,000,000,000)  
//      Rigel       (8,232,000,000,000,000)
```

注釈

このオーバーロードは `ICompareble<T>` インターフェースの `CompareTo` メソッドの実装です。ジェネリクスコレクションがコレクション内の項目の並べ替えに使用します。

こちらをご覧ください `Compare(Rational,Rational)`、`ICompareble<T>`、`Equals(Rational)`

CompareTo(UInt64)

⚠ 重要

この API は CLS 準拠ではありません。

このインスタンスと 符号なし 64 ビット整数を比較し、このインスタンスの値が 符号なし 64 ビット整数の値よりも小さいか、同じか、または大きいかを示す整数を返します。

```
public int CompareTo(ulong other);
```

パラメーター

other UInt64

比較する 符号なし 64 ビット整数。

戻り値

Int32

現在のインスタンスと other の相対的な値を示す符号付き整数値です（次の表を参照）。

戻り値	説明
0 より小さい値	現在のインスタンスは other より小さい。
0	現在のインスタンスと other は等しい。
0 より大きい値	現在のインスタンスは other より大きい。

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