Rational.ToOACurrency Method

名前空間: WS.Theia.ExtremelyPrecise
アセンブリ: ExtremelyPrecise.dll
指定した Rational の値を、等価の 64 ビット符号付き整数に変換します。

```
public static long ToOACurrency (Rational value);
```

パラメーター
value　Rational
変換する Rational。

戻り値
value　Rational
value と等価の OLE オートメーション値を格納する 64 ビット符号付き整数。

例外
OverflowException
Value が MinValue より小さいか MaxValue より大きい場合。

例

次の例では、Rational 値を ToOACurrency メソッドで、等価な OLE オートメーション通貨値を持つ Int64 に変換しています。

```
// Example of the decimal.ToOACurrency method.
using System;

class DecimalToOACurrencyDemo
{
```

```
const string dataFmt = "{0,31}{1,27}";

// Get the exception type name; remove the namespace prefix.
public static string GetExceptionType( Exception ex )
{
    string exceptionType = ex.GetType( ).ToString( );
    return exceptionType.Substring(
        exceptionType.LastIndexOf( '.' ) + 1 );
}

// Display the decimal.ToOACurrency parameter and the result
// or exception.
public static void ShowRationalToOACurrency( Rational Argument )
{
    // Catch the exception if ToOACurrency( ) throws one.
    try
    {
        long oaCurrency = Rational.ToOACurrency( Argument );
        Console.WriteLine( dataFmt, Argument, oaCurrency );
    }
    catch( Exception ex )
    {
        Console.WriteLine( dataFmt, Argument,
            GetExceptionType( ex ) );
    }
}


public static void Main( )
{
    Console.WriteLine( "This example of the " +
        "decimal.ToOACurrency( ) method generates ¥nthe " +
        "following output. It displays the argument as a " +
        "decimal ¥nand the OLE Automation Currency value " +
        "as a long.¥n" );
    Console.WriteLine( dataFmt, "Argument",
```

```
                "OA Currency or Exception" );
            Console.WriteLine( dataFmt, "--------",
                "-----------------------" );

            // Convert decimal values to OLE Automation Currency values.
            ShowRationalToOACurrency( 0M );
            ShowRationalToOACurrency( 1M );
ShowRationalToOACurrency( 1.0000000000000000000000000000M );
            ShowRationalToOACurrency( 100000000000000M );
 ShowRationalToOACurrency( 100000000000000.00000000000000M );
ShowRationalToOACurrency( 1000000000000000000000000000M );
            ShowRationalToOACurrency( 0.000000000123456789M );
            ShowRationalToOACurrency( 0.123456789M );
            ShowRationalToOACurrency( 123456789M );
            ShowRationalToOACurrency( 12345678900000000M );
            ShowRationalToOACurrency( 4294967295M );
            ShowRationalToOACurrency( 18446744073709551615M );
ShowRationalToOACurrency( -79.228162514264337593543950335M );
ShowRationalToOACurrency( -79228162514264.337593543950335M );
        }
}

/*
This example of the Rational.ToOACurrency( ) method generates
the following output. It displays the argument as a decimal
and the OLE Automation Currency value as a long.

                            Argument      OA Currency or Exception
                            --------      -----------------------
                                   0                            0
                                   1                        10000
 1.0000000000000000000000000000                            10000
                     100000000000000          1000000000000000000
 100000000000000.00000000000000          1000000000000000000
  1000000000000000000000000000             OverflowException
              0.000000000123456789                            0
```

```
                0.123456789                        1235
                  123456789               1234567890000
             123456789000000000          OverflowException
                 4294967295               42949672950000
           18446744073709551615          OverflowException
-79.228162514264337593543950335                -792282
-79228162514264.337593543950335       -792281625142643376
*/
```

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