**Evaluation Only. Created with Aspose.Words. Copyright 2003-2021 Aspose Pty Ltd.**

La instrucción foreach ejecuta una instrucción o un bloque de instrucciones para cada elemento en una instancia del tipo que implementa la

interfaz System.Collections.IEnumerable o System.Collections.Generic.IEnumerab le<T>. La instrucción foreach no se limita a esos tipos y puede aplicarse a una instancia de cualquier tipo que satisfaga las siguientes condiciones:

* tiene el método público sin parámetros GetEnumerator cuyo tipo de valor devuelto es clase, estructura o tipo de interfaz,
* el tipo de valor devuelto del método GetEnumerator tiene la propiedad pública Current y el método público sin parámetros MoveNext cuyo tipo de valor devuelto es Boolean.

A partir de C# 7.3, si la propiedad Current del enumerador devuelve un valor devuelto de referencia(ref T donde T es el tipo del elemento de colección), se puede declarar la variable de iteración con el modificador ref o ref readonly.

En cualquier punto del bloque de instrucciones foreach, se puede salir del bucle mediante la instrucción break, o bien se puede ir a la siguiente iteración del bucle mediante la instrucción continue. También se puede salir de un

bucle foreach mediante las instrucciones goto, return o throw.

Si la instrucción foreach se aplica a null, se produce NullReferenceException. Si la colección de origen de la instrucción foreach está vacía, el cuerpo del

bucle foreach no se ejecuta y se omite.

**Ejemplos**

El siguiente ejemplo muestra el uso de la instrucción foreach con una instancia del tipo List<T> que implementa la interfaz IEnumerable<T>:

var fibNumbers = new List<int> { 0, 1, 1, 2, 3, 5, 8, 13 }; int count = 0;

foreach (int element in fibNumbers)

{

count++;

Console.WriteLine($"Element #{count}: {element}");

}

Console.WriteLine($"Number of elements: {count}");

El siguiente ejemplo utiliza la instrucción foreach con una instancia del tipo System.Span<T>, que no implementa ninguna interfaz:

public class IterateSpanExample {

public static void Main()

{

Span<int> numbers = new int[] { 3, 14, 15, 92, 6 }; foreach (int number in numbers)

{

Console.Write($"{number} ");

}

Console.WriteLine();

}

}

En el ejemplo siguiente se usa una variable de iteración ref para establecer el valor de cada elemento de una matriz stackalloc. La versión ref readonly recorre en iteración la colección para imprimir todos los valores. En la declaración

de readonly, se usa una declaración de variable local implícita. Las declaraciones de variables implícitas se pueden usar con las declaraciones ref o ref readonly, al igual que las declaraciones de variables con tipo explícito.

public class ForeachRefExample

{

public static void Main()

{

Span<int> storage = stackalloc int[10]; int num = 0;

foreach (ref int item in storage)

{

item = num++;

}

foreach (ref readonly var item in storage) {

Console.Write($"{item} ");

}

// Output:

// 0 1 2 3 4 5 6 7 8 9

}

}