La Instrucción *Using*

.Net Framework proporciona administración de recursos para objetos administrados a través del recolector de basura: no es necesario asignar y liberar memoria explícitamente para los objetos administrados. Las operaciones de limpieza para cualquier recurso no administrado deben realizarse en el destructor en C #. Para permitir que el programador realice explícitamente estas actividades de limpieza, los objetos pueden proporcionar un método de eliminación que se puede invocar cuando el objeto ya no se necesita. La instrucción *using* en C # define un límite para el objeto fuera del cual, el objeto se destruye automáticamente. La instrucción *using* se termina cuando el bloque de instrucciones "*using*" o la ejecución sale del bloque de instrucciones "*using*" indirectamente, como, por ejemplo, cuando se lanza una excepción. La instrucción "*using*" nos permite especificar múltiples recursos en una sola declaración. El objeto también se puede crear fuera de la instrucción "*using*". Los objetos especificados dentro del bloque *using* deben implementar la interfaz IDisposable. El marco invoca el método Dispose de los objetos especificados dentro de la instrucción "*using*" cuando se sale del bloque.

La instrucción *using* define el alcance del objeto / recurso. Cuando el alcance de la instrucción *using* es (implícitamente cuando se llega a "}" del bloque *using* o explícitamente es decir lanzando una excepción desde el bloque de uso), entonces el recurso es eliminado automáticamente por el CLR en lugar del recolector de basura que no está determinado.

El uso de la declaración *using* afectará el rendimiento. Si almacenamos un conjunto de datos grande y un objeto pesado, y si se utilizan con la instrucción *using*, entonces se liberará el recurso innecesario tan pronto como se termine de ejecutar el bloque *using*.

Un programa / clase que hace uso de la instrucción *using* debe implementar la interfaz IDisposable. A continuación se muestra el método proporcionado por la interfaz IDisposable:

```
class Libro : IDisposable
{
    private String nombre;
    private int numeroPaginas;

    public String Nombre
    {
        get {return nombre;}
        set {nombre = value;}
   }

    public int NumeroPaginas
    {
        get {return numeroPaginas;}
        set {numeroPaginas = value;}
   }

    public void Mostrar()
    {
        Console.WriteLine("Libro Nombre:{0}", Nombre);
        Console.WriteLine("Libro NumeroPaginas:{0}", NumeroPaginas);
   }
```

```
    void IDisposable.Dispose()
    {
        Console.WriteLine("El Alcance de using ha finalizado. El recurso
Libro se ha dispuesto.");
   }
}
```

La clase Libro implementa la interfaz requerida. Tiene dos miembros de datos (nombre y numeroPaginas) que almacenan el nombre y el número de páginas del libro, respectivamente. El método Mostrar () muestra el valor de ambos miembros de datos.

También notará que la clase Libro también proporciona la definición del dispose () de la interfaz IDisposable. Contiene una declaración simple de escritura que se utiliza para notificarnos que el uso del recurso ha terminado.

Ahora sobre el programa principal, el programa principal se muestra a continuación:

```
class Program
{
   static void Main(string[] args)
   {
      Console.WriteLine("Antes de la instrucción using");
      using (Libro mv = new Libro())
      {
         Console.WriteLine("Dentro de la instrucción using");
         mv.Nombre = "Juan";
         mv.NumeroPaginas = 204;
         mv.Mostrar();
      }
      Console.WriteLine("Después de la instrucción using");
  }
}
```

Como se muestra arriba, en primer lugar, el programa principal muestra " Antes de la instrucción *using* " utilizando el método writeline. Este método es solo para notificar que la siguiente declaración es una instrucción *using*. Después de que el programa principal del bloque de *using* muestre " Después de la instrucción *using*" usando el método de writeline. Este método es solo para notificar que el alcance de la declaración *using* ha terminado.

Esta aplicación realizará las siguientes acciones:

En primer lugar, la aplicación imprime " Antes de la instrucción *using* " usando el método Console.WriteLine ("Antes de la instrucción *using* ");. La aplicación imprime " Dentro de la instrucción *using* " usando el método Console.WriteLine ("Dentro de la instrucción *using* ");. A continuación, la aplicación asigna los valores y los imprime utilizando el método Mostrar (). Cuando el alcance de la instrucción *using* termina, es decir, cuando se llega a la instrucción "}" de *using*, el CLR llama automáticamente al método Dispose () de la interfaz iDisposable implementada por la clase Libro. Por último, la aplicación imprime " Después de la instrucción *using* " usando el método Console.WriteLine ("Después de la instrucción *using*");.

¿Por qué usamos la instrucción *Using*?

C# nos proporciona una instrucción "*using*" para llamar al método Dispose explícitamente. La instrucción *using* nos proporciona una forma adecuada de invocar el método Dispose en el objeto.

En la declaración *using*, instanciamos un objeto en la declaración. Al final del uso del bloque de instrucciones, se llama automáticamente al método Dispose.

La instrucción *using* proporciona algunas características únicas.

- Administrar el alcance: también gestiona el alcance del objeto. Al final del bloque *using*, al usar llamadas al método Dispose y en el método, el objeto libera todos sus recursos y no debería estar disponible más.
- Crea una instancia del objeto como de solo lectura: las instancias de objeto en la instrucción *using* son de solo lectura y no puede modificar ni reasignar el objeto. Esta función garantiza que se invoque el método Dispose al objeto que se crea una instancia.
- Asegura el método Dispose siempre se llamará: la instrucción *using* también garantiza que el método Dispose se invocará siempre si se produjo o no una excepción.

FIN