

C sharp NET/Capítulo 6

Arrays

Los arrays funcionan como colecciones de elementos. Se pueden utilizar para reunir muchos elementos en un solo grupo, y así poder realizar diversas acciones con ellos, por ejemplo, su clasificación.

Declaración

Las matrices se declaran igual que las variables pero añadiendo un conjunto de corchetes [] después de poner el tipo de dato, como por ejemplo:

```
String [] nombres;
```

Es necesario crear una instancia de la matriz para utilizarla, lo que se hace de esta manera:

```
string [] names = new string [2];
```

El número (2) se refiere al tamaño de la matriz, es decir, la cantidad de artículos que se pueden poner en él. Colocar elementos en la matriz es bastante simple, por ejemplo:

```
nombres [0] = "John Doe";
```

IMPORTANTE: El primer elemento está indexado como 0, el siguiente como 1 y así sucesivamente.

Recorrido

Ejemplo 6.1:

```
using System;
using System.Collections;

namespace ConsoleApplication1
{
    class Program
```

C sharp NET/Capítulo 6

```
{
    static void Main(string[] args)
    {
        string[] names = new string[2];

        names[0] = "John Doe";
        names[1] = "Jane Doe";

        foreach(string s in names)
            Console.WriteLine(s);

        Console.ReadLine();
    }
}
```

Usamos el bucle foreach, porque es el más fácil, pero por supuesto que podríamos haber utilizado uno de los otros tipos de bucle en su lugar. El bucle for es bueno con los Arrays, así, por ejemplo, también podríamos hacer lo siguiente:

```
for(int i = 0; i < names.Length; i++)
    Console.WriteLine("Item number " + i + ": " + names[i]);
```

En este ejemplo, se usa la propiedad Length del vector para decidir cuántas veces se debe repetir el bucle, a continuación, se utiliza el contador (i) para saber por dónde empezar a contar (i=0) y hasta dónde contar (i < names.Length) y, por supuesto, de cuánto avanzar gracias al incremento i++ (de uno en uno).

Inicialización

Hay diferentes maneras. Aquí tenemos dos ejemplos. El primero es más largo que el segundo!

```
int[] numbers = new int[5] { 4, 3, 8, 0, 5 };
int[] numbers = { 4, 3, 8, 0, 5 };
```

C sharp NET/Capítulo 6

Métodos

La clase Array tiene muchos métodos para usar, veamos el ejemplo:

```
using System;
using System.Collections;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] numbers = { 4, 3, 8, 0, 5 };

            Array.Sort(numbers);

            foreach(int i in numbers)
                Console.WriteLine(i);

            Console.ReadLine();
        }
    }
}
```

Más métodos de esta clase:

- Array.AsReadOnly
- Array.BinarySearch
- Array.Clear
- Array.ConstrainedCopy
- Array.ConvertAll
- Array.Copy
- Array.CreateInstance
- Array.Exists
- Array.Find
- Array.FindIndex
- Array.ForEach
- Array.IndexOf
- Array.LastIndexOf
- Array.Resize
- Array.Reverse
- Array.Sort
- Array.TrueForAll

C sharp NET/Capítulo 6

Propiedades

Continuando, se examinan las propiedades de la clase Array. La propiedad más comúnmente utilizados en las matrices es la propiedad Length.

Length
Count
IsFixedSize,
IsReadOnly
IsSynchronized

Para más información sobre cadauno de estos métodos y sobre las propiedades ves a la web

<http://www.dotnetperls.com/array>

y mira en la sección sobre Methods i Properties. Allí encontrarás enlaces con ejemplos de cadauno de ellos.