

C sharp NET/Capítulo 9

ArrayList

ArrayList redimensiona dinámicamente. Cuando se agregan elementos, crece con la capacidad para adaptarse a ellos.

Se utiliza con mayor frecuencia en los mayores programas de C #.

Agregar elementos

El método Add de ArrayList es de uso frecuente. Con éste método se anexa un objeto nuevo al final de la clase ArrayList. Puede seguir añadiendo elementos a su colección hasta que la memoria se agote. Los objetos se almacenan en el montón administrado.

Ejemplo:

```
using System.Collections;

class Program
{
    static void Main (){
// //
// // Crear un objeto ArrayList y añadir tres elementos.
// //
        ArrayList list = new ArrayList ();
        list.Add ("One");
        list.Add ("dos");
        list.Add ("Tres");
    }
}
```

Resultado

Se está creando un ArrayList con tres elementos.

Combinar

Hay diversas maneras de agregar un ArrayList a otro, pero la mejor manera es utilizando AddRange. Internamente, utiliza los métodos AddRange, Array.Copy o CopyTo, que tienen un mejor rendimiento que algunos bucles.

Ejemplo:

C sharp NET/Capítulo 9

```
using System;
using System.Collections;

class Program
{
    static void Main ()
    {
        //
        // Crear un objeto ArrayList con dos valores.
        //
        ArrayList list = new ArrayList ();
        list.Add (5);
        list.Add (7);
        //
        // Segunda ArrayList.
        //
        ArrayList list2 = new ArrayList ();
        list2.Add (10);
        list2.Add (13);
        //
        // Añadir ArrayList segunda a la primera.
        //
        list.AddRange (list2);
        //
        // Muestra los valores.
        //
        foreach (int i en la lista)
        {
            Console.WriteLine (i);
        }
    }
}
```

Resultado

```
5
7
10
13
```

C sharp NET/Capítulo 9

Count, Clear

La clase ArrayList proporciona la propiedad Count.

Ejemplo:

Aquí: El ejemplo muestra la propiedad Count. También se muestra el método Clear, y cómo esto afecta a la cuenta.

```
using System;
using System.Collections;

class Program
{
    static void Main()
    {
        //
        // Create an ArrayList with two values.
        //
        ArrayList list = new ArrayList();
        list.Add(9);
        list.Add(10);
        //
        // Show number of elements in ArrayList.
        //
        Console.WriteLine(list.Count);
        //
        // Clear the ArrayList.
        //
        list.Clear();
        //
        // Show count again.
        //
        Console.WriteLine(list.Count);
    }
}
```

Resultado

2
0

C sharp NET/Capítulo 9

La propiedad Count devuelve un int. Éste siempre será un valor positivo.

El programador puede necesitar llamar al método Clear en su instancia ArrayList. Internamente, éste llama al método Array.Clear. A veces, el código es más claro si se crea un ArrayList nuevo en su lugar.

Sort, Reverse

La ordenación de un ArrayList se hace a menudo. Proporciona una vista de los datos que son útiles tanto para los usuarios como para los equipos. El método de ordenación en las bibliotecas de clase base es un método de instancia sin parámetros en ArrayList.

Ejemplo:

```
using System;
using System.Collections;

class Program
{
    static void Main()
    {
        //
        // Create an ArrayList with four strings.
        //
        ArrayList list = new ArrayList();
        list.Add("Cat");
        list.Add("Zebra");
        list.Add("Dog");
        list.Add("Cow");
        //
        // Sort the ArrayList.
        //
        list.Sort();
        //
        // Display the ArrayList elements.
        //
        foreach (string value in list)
        {
            Console.WriteLine(value);
        }
        //
        // Reverse the ArrayList.
        //
        list.Reverse();
        //
    }
}
```

C sharp NET/Capítulo 9

```
// Display the ArrayList elements again.
//
foreach (string value in list)
{
    Console.WriteLine(value);
}
}
```

Resultado

```
Cat
Cow
Dog
Zebra
```

```
Zebra
Dog
Cow
Cat
```

Insert, remove

A continuación vemos cómo se puede insertar y quitar elementos con la familia de métodos Insert y Remove. En concreto, vemos el método RemoveAt para el borrado de un solo elemento, entonces el método de inserción y métodos RemoveRange.

Ejemplo:

```
using System;
using System.Collections;

class Program
{
    static void Main()
    {
        //
        // Create an ArrayList with three strings.
        //
        ArrayList list = new ArrayList();
        list.Add("Dot");
        list.Add("Net");
        list.Add("Perls");
```

C sharp NET/Capítulo 9

```
//  
// Remove middle element in ArrayList.  
//  
list.RemoveAt(1); // It becomes [Dot, Perls]  
//  
// Insert word at beginning of ArrayList.  
//  
list.Insert(0, "Carrot"); // It becomes [Carrot, Dot,  
Perls]  
//  
// Remove first two words from ArrayList.  
//  
list.RemoveRange(0, 2);  
//  
// Display the result ArrayList.  
//  
foreach (string value in list)  
{  
    Console.WriteLine(value); // <-- "Perls"  
}  
}
```

Resultado

Perls

Loop with for

El bucle for es muy popular y útil.

Ejemplo:

```
using System;  
using System.Collections;  
  
class Program  
{  
    static void Main()  
    {  
        //  
        // Create an ArrayList with three strings.  
        //  
        ArrayList list = new ArrayList();  
        list.Add("man");  
    }  
}
```

C sharp NET/Capítulo 9

```
list.Add("woman");
list.Add("plant");
//
// Loop over ArrayList with for.
//
for (int i = 0; i < list.Count; i++)
{
    string value = list[i] as string;
    Console.WriteLine(value);
}
}
```

Resultado

```
man
woman
plant
```

IMPORTANTE

Si queréis saber para qué sirve **as** mirad esta dirección web:
<http://www.dotnetperls.com/as>

Get range

Otro método que puede utilizar es el método `GetRange`. Esto devolverá un subconjunto de un `ArrayList` en un `ArrayList` nuevo.

Ejemplo:

```
using System;
using System.Collections;

class Program
{
    static void Main()
    {
        //
```

C sharp NET/Capítulo 9

```
// Create an ArrayList with 4 strings.
//
ArrayList list = new ArrayList();
list.Add("fish");
list.Add("amphibian");
list.Add("bird");
list.Add("plant");
//
// Get last two elements in ArrayList.
//
ArrayList range = list.GetRange(2, 2);
//
// Display the elements.
//
foreach (string value in range)
{
    Console.WriteLine(value); // bird, plant
}
}
```

Resultado

```
bird
plant
```

IndexOf, LastIndexOf

Los métodos IndexOf y LastIndexOf sobre ArrayList son similares a los de cadenas.

Ejemplo:

```
using System;

class Program
{
    static void Main()
    {
        // A.
        // The input string.
        const string s = "Darth Vader is really scary.";
    }
}
```


C sharp NET/Capítulo 9

```
// B.  
// Test with IndexOf.  
if (s.IndexOf("Vader") != -1)  
{  
    Console.WriteLine("string contains 'Vader'");  
}  
Console.ReadLine();  
}
```

Resultado

string contains 'Vader'

Estos apuntes han sido sacados de la web

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