

El más buscado

Programme la clase siguiente.

```
namespace Weboo.Programming.ThirdRound
{
    public class MostWanted : Weboo.Collections.MostWanted.IMostWanted
    {
        public MostWanted()
        {
            ... ..
        }
        ... ..
    }
}
```

La interfaz `Weboo.Collections.MostWanted.IMostWanted` (implementada por la clase `Weboo.Programming.ThirdRound.MostWanted`) se brinda en la biblioteca de clases `Weboo.Collections.MostWanted.dll` y se define tal y como se muestra a continuación.

```
namespace Weboo.Collections.MostWanted
{
    public interface IMostWanted : System.Collections.IEnumerable
    {
        //Permite conocer la cantidad de elementos
        //almacenados en la estructura
        int Count {get;}

        //Adiciona el elemento o a la colección
        //Si el elemento o ya se encuentra
        //almacenado en la colección este método
        //no hace nada
        //Si se intenta añadir el elemento null
        //el método debe lanzar ArgumentNullException
        void Add(object o);

        //Elimina el elemento o de la colección
        //Si el elemento o no se encuentra
        //almacenado en la colección este método
        //no hace nada
        void Remove(object o);

        //Devuelve true si el elemento o
        //se encuentra en la colección y
        //false en otro caso
        bool Contains(object o);
    }
}
```

Note que la interfaz `Weboo.Collections.MostWanted.IMostWanted` implementa la interfaz `System.Collections.IEnumerable`, Para la clase que usted debe programar, o sea la clase `Weboo.Programming.ThirdRound.MostWanted`, el

enumerador asociado itera sobre todos los elementos de la colección según la cantidad de veces que se haya preguntado por cada elemento (con el método `Contains(object o)`). O sea, el primer elemento sobre el que itere el enumerador será aquel sobre el que se ha preguntado (con el método `Contains(object o)`) si pertenece a la colección una mayor cantidad de veces, el segundo elemento será aquel sobre el que se ha preguntado una segunda mayor cantidad de veces, y así sucesivamente hasta el último elemento sobre el que se itere, que será aquel sobre el que se ha preguntado la menor cantidad de veces, incluso cero veces. En caso de que por varios elementos se haya preguntado una misma cantidad de veces, sobre ellos se debe iterar según el orden en que fueron añadidos a la colección.

Recuerde que si ocurren cambios en la doble cola (se adicionan o se quitan elementos) todos los enumeradores obtenidos antes de la ocurrencia de dichos cambios deben quedar inválidos, o sea, al ejecutar `MoveNext()` o `Reset()` se debe lanzar `InvalidOperationException` y al ejecutar `Current` se debe obtener el mismo resultado que se hubiera obtenido si no se hubiera modificado la colección. Para más información puede consultar en la ayuda la especificación de los métodos de la interfaz `System.Collections.IEnumerator`.

Nota: En su implementación usted NO PUEDE utilizar las siguientes clases e interfaces ni herederos de las mismas:

- `System.Collections.ArrayList`.
- `System.Collections.CollectionBase`
- `System.Collections.IDictionary`
- `System.Collections.Stack`
- `System.Collections.Queue`

Ejemplos.

```
MostWanted mw = new MostWanted();
```

```
mw.Add("k");  
mw.Add(5);  
mw.Add("z");  
mw.Add("m");  
mw.Add("l");
```

```
//al iterar sobre la coleccion  
//los elementos se obtienen en el mismo  
//orden en el que fueron añadidos  
//ya que no se ha preguntado por  
//ninguno
```

```
foreach(object o in mw)  
    Console.Write(o + " ");
```

Aparece en la consola:
k 5 z m l

```
Console.WriteLine(mw.Contains("m"));  
Console.WriteLine(mw.Contains("i"));  
Console.WriteLine(mw.Contains("l"));  
Console.WriteLine(mw.Contains("m"));
```

```

Console.WriteLine(mw.Contains(5));

//por "m" se pregunto dos veces
//por "l" se pregunto una vez al
//igual que por 5, pero como 5
//se inserto primero 5 se obtiene antes
//que "l"
//por "k" y por "z" no se ha preguntado
//y se obtienen en el orden en que fueron
//insertados
foreach(object o in mw)
    Console.Write(o + " ");
    Aparece en la consola:
    m 5 l k z

Console.WriteLine(mw.Contains("l"));
mw.Remove("m");
mw.Add("y");
Console.WriteLine(mw.Contains("y"));
Console.WriteLine(mw.Contains("y"));

//"l" pasa a ser el primer elemento
//porque se ha preguntado dos veces por él
//y se añadió antes que "y" que pasa a ser
//el segundo
//5 es el tercero porque por el se ha preguntado
//una vez
//luego se obtienen "k" y "z" en ese orden
foreach(object o in mw)
    Console.Write(o + " ");
    Aparece en la consola:
    l y 5 k z

IEnumerator e = mw.GetEnumerator();

e.MoveNext();
Console.WriteLine(e.Current);
    Aparece en la consola:
    l

e.MoveNext();
Console.WriteLine(e.Current);
    Aparece en la consola:
    y

mw.Add(17);

Console.WriteLine(e.Current);
    Aparece en la consola:
    y

e.MoveNext();
    Lanza
    InvalidOperationException
    porque la colección ha sido
    modificada

e.Reset();
    Lanza
    InvalidOperationException
    porque la colección ha sido
    modificada.

```