

**Universidad de la Habana.**  
**Facultad de Matemática y Computación.**  
**Carrera Ciencia de la Computación, Asignatura Programación**  
**3er Problema de Programación, Curso 2006 - 2007**

En la biblioteca de clases que se le suministra (*Interfaces.dll*) se definen las siguientes interfaces.

```
namespace Interfaces
{
    public interface IClassifier
    {
        object Classify(object o);
    }
    public interface IClassifierEnumerator : IEnumerator
    {
        object CurrentClassification {get;}
        bool MoveNextClassification();
    }
}
```

Programa una clase `ClassifierEnumerator` que implemente la interfaz `IClassifierEnumerator`. Note que la interfaz `IClassifierEnumerator` hereda de `IEnumerator`, por lo tanto, su clase será un enumerador con operaciones adicionales. La clase deberá ser implementada dentro de un proyecto de tipo “*Class Library*” con nombre *Pregunta3* (**y no *Pregunta3.dll* !!**).

El constructor de su clase debe tener la siguiente signatura:

```
public ClassifierEnumerator(IEnumerator baseEnum,
                           IClassifier classifier){ ...}
```

Como se observa, una instancia de `ClassifierEnumerator` se construirá a partir de otro enumerador y un objeto de alguna clase que implemente `IClassifier`. Se entenderá por clasificación de un objeto el resultado de evaluar en el mismo el método `Classify` del `IClassifier` suministrado. El propósito del `ClassifierEnumerator` es iterar sobre los mismos objetos devueltos por el enumerador original, pero **en orden de clasificación**, o sea, los objetos correspondientes a la primera clasificación serán devueltos antes de los correspondientes a la segunda clasificación y así sucesivamente. La precedencia entre clasificaciones estará determinada por el orden en que aparezca el primer objeto de cada clasificación en la enumeración original.

En el siguiente ejemplo, asuma que se utiliza un objeto clasificador que para cada cadena de caracteres devuelve como clasificación su longitud.

enumeración original (la clasificación se indica entre paréntesis)	"enero" (5), "febrero" (7), "marzo" (5), "abril" (5), "mayo" (4), "junio" (5), "julio" (5), "agosto" (6), "septiembre" (10)
orden de las clasificaciones	5, 7, 4, 6, 10
enumeración con ClassifierEnumerator	"enero", "marzo", "abril", "junio", "julio", "febrero", "mayo", "agosto", "septiembre"

Adicionalmente a las operaciones habituales aplicables a todo enumerador, un `IClassifierEnumerator` deberá devolver mediante la propiedad `CurrentClassification` la clasificación del objeto actual y permitirá avanzar hasta el primer objeto de la clasificación siguiente mediante el método `MoveNextClassification`.

#### Aclaraciones:

- Dos objetos tendrán la misma clasificación si al aplicarle el método `Equals` a sus clasificaciones se devuelve `true`.
- Dos objetos de igual clasificación deberán ser devueltos en el mismo orden en que aparecen en la enumeración original.
- Acceder a la propiedad `CurrentClassification` antes de haber ejecutado `MoveNext` o `MoveNextClassification` provocará una excepción de tipo `InvalidOperationException`.
- Antes de haber ejecutado `MoveNext`, aplicar `MoveNextClassification` a un `ClassifierEnumerator` provoca que se avance hasta el primer objeto de la primera clasificación (y pasa a ser legal aplicar las propiedades `Current` y `CurrentClassification`).
- Debe utilizar los nombres por defecto para el ensamblado generado (Pregunta3.dll) y el *namespace* que contendrá la clase (Pregunta3).