Mandatory Activity. Object Oriented Paradigm. Lab 04.

This activity must be autonomously done by the student. **It must be done prior to the following laboratory class**. It will be used as part of the following laboratory.

Activity

The following two .NET collections are widely used:

- `List<T>`: Class that implements the `IList<T>` interface, represents a collection of objects that can be individually accessed by index (a Vector)
- `Dictionary<TKey,TValue>`: Class that implements the `IDictionary<TKey,TValue>` interface, which represents a generic collection of key/value pairs (`TKey` is the key type and `TValue` the value type). This type of collection is commonly called maps or associative arrays. A usual implementation is by means of a hash table, although it can also be developed with a tree data structure.

Both classes belong to the `System.Collections.Generic` namespace. Read this brief description of these two classes: http://www.csharp-station.com/Tutorials/Lesson20.aspx

After reading the previous page, create a `vector.test` testing project that, using `IList<T>` references (instead of `List<T>`), tests the following features:

1. Add elements
2. Obtain the number of elements
3. Get and set the element of the i_{th} position
4. Consult whether or not an element is in the vector
5. Obtain the index of the first occurrence of an element in the vector
6. Delete the first occurrence of a given element
7. Iterate throughout the elements with a `foreach` loop

Now, create a `dictionary.test` project that, using `IDictionary<TKey,TValue>` references (instead of `Dictionary<TKey,TValue>`), tests the following features:

1. Add elements with a given key and value
2. Obtain the number of pairs in the collection
3. Get and set the value of a given key
4. Consult whether or not a key exists in the dictionary
5. Delete a pair giving its key
6. Iterate throughout the pairs (both key and value) with a `foreach` loop

To implement tests, use the types created in previous laboratories, apart from `int` and `string`