Contents

AList

Description

Purpose

This project is designed to teach you how to implement lists using a more familiar datastructure, arrays. It requires to understand abstract data types, array manipulations and interfaces (as a bonus).

Challenge

In short Our goal is to provide an implementation of the list abstract data type using arrays. Our implementation should be "from an external point of view" exactly like lists, with features similar to the C# implementation of lists.

In more details We want to implement (a simplified version of) the list abstract data type. A list should be

- a finite collection of elements,
- in a particular order,
- capable of containing the same element multiple times.

and we should have operations to…

- … create an empty list,
- … test for emptiness,
- … obtain the number of elements in the list,
- … add an element at "the beginning" of the list ("to the left"),
- … remove the first element with a particular value if it exists,
- … empty the list.

The main challenge of this project is to implement those features using only an array as an attribute. Your class should begin as follows:

```
using System;
```

```
public class AList<T>
{

  private T[] alist;
  …
}
```

with the array `alist` being used to store the elements of the list. **Pay attention to details**:

- make sure that the class always returns the right value for the number of elements in the list,
- make sure your methods correctly insert and remove values without losing any information,
- write good test cases in your `main` method,
- **do not load any additional libraries**, in particular, **do not use C# native lists or LINQ**.

Bonuses

Bonus points will be given if:

- (easy) You provide at least one additional methods among the following:
  - A relevant `ToString`,
  - A method to insert a value "at the end" ("to the right") of the list,
  - A method to insert a value at a given index.
- (medium) Your `main` method is particularly well thought, and showcase many test cases. **A proper `main` method will help you in debugging your code.**
- (hard) Make your class realizes the `ICollection<T>` interface:
  - You are then allowed to load the additional two libraries:

    ```
    using System.Collections;
    using System.Collections.Generic;
    ```

  - You will need to add a second attribute to your class,
  - Add all the required methods, in particular write a meaningful `CopyTo` method.

Submission

Please, follow our guideline on project submission. In particular, make sure you write your name and the date in a delimited comment at the beginning of your file.