wa8

Due Fri Nov 20 at 6:00 PM.
Submit as wa8.cs at fileupload.ca.

Assignment

The following provided files comprise one of the data files and all of the C# code files for the linked list of `Drug` objects studied in class to date. You should place all seven files in a C# project folder. As provided, the program should compile and run to display 100 drugs on the console.

> RXQT1503-100.txt
> Bme121-LinkedList-AddFirst,AddLast.cs
> Bme121-LinkedList-Contains,Remove.cs
> Bme121-LinkedList-Node,properties,ctor.cs
> Bme121-LinkedList-ToArray.cs
> MediCal-Drug-2.cs
> wa8.cs

Here, we will modify only the wa8.cs file which holds both the `Program` class containing the `Main` method and a part of the `LinkedList` class containing the `IsTarget`, `Compare`, and `InsertInOrder` methods. The goal of this assignment is to complete the `InsertInOrder` method. Nothing else should be altered.

Recall that insertion sort works by picking one element at a time and placing it into its proper position in an already sorted collection of elements. Using this approach, we can create a sorted linked list by adding elements one at a time using a method which places elements in their sorted position with respect to those elements already added to the list. Note that this will only work if we maintain the list in sorted order, i.e., we don't make other calls to methods like `AddFirst` or `AddLast` which don't maintain the ordering.

Complete the `InsertInOrder` method so it will put the `Drug` object passed as its argument into the linked list just before the first `Drug` on the list which compares as larger. If there are no larger ones in the list, the new one is added at the end. Use the provided `Compare` method to compare the `Drug` objects. It will compare drugs based on an alphabetical ordering of their names.

The existing `Main` method will test your `InsertInOrder` method. It collects 100 `Drug` objects from the file into an array. It then calls `InsertInOrder` for each `Drug` in the array. Finally, it displays the linked list size and contents. As provided, `InsertInOrder` just calls `AddLast` so elements are added to the list in the order they appeared in the file and array. This seems to be in order by the drug identification number. Once `InsertInOrder` is working, the linked list will hold/display the drugs in order by their names.

When coding any method for a linked list, it is useful to draw or consult the diagrams for all cases.

Aside: In the provided `IsTarget` and `Compare` methods, I tried to follow a best practice for string comparisons which says we should always try to be explicit about how the comparison is done. Many of the string methods allow an argument specifying the comparison by a member of the `StringComparison` enumeration. I used `StringComparison.OrdinalIgnoreCase` which is fast and

culturally agnostic. This might not be appropriate if we were sorting the drugs for display to a user in a specific culture.