CMPSCI 182L Data Structures and Program Design: Lab

Project 2 Linked Lists (30 points)

Due October 4, 2022

Implement the ADT List discussed on pages 204 through 209 of the text book as the class ***ADTMyList***. *ADTMyList* will "simulate" an unsorted list using an interface and a linear linked list "behind" that interface. Include the following *ADTMyList* constructors and methods:

`public` **`ADTMyList`**`()`

A constructor which allocates an empty linear linked list with a null head reference and a size of 0.

`public` **`ADTMyList`**`(String[] listItems)`

A constructor which initializes a linear linked list so that it represents the same sequence of items as the items in the `listItems` array.

`public void` **`add`**`(int index, String newItem)`

Inserts `newItem` at position `index` of the linear linked list if 0 <= `index` <= `size()`. Outputs an error message that `index` is an invalid index number for the item `newItem` if `index` is less than zero or greater than `size()`. Remember, there are three "cases" for adding a Node to a linear linked list:

1. The request may be to add the item to the **head** of the linked list.
2. The request may be to add the item **somewhere in the body** of the linked list.
3. The request may be to add the item to the **end** of the linked list.

Your add method must implement these three "cases" and make its "decision" of which one to use based on the value of `index`.

If the given `index` is invalid, the method should display an error message like "index is an invalid index number for the item" where index is the value of `index` and item is the value of `newItem`.

`public String` **`get`**`(int index)`

Returns the item at position `index` of the linear linked list if 0 <= `index` < `size()`. Returns an empty String if `index` is less than zero or greater than or equal to `size()`.

`public void` **`remove`**`(int index)`

Removes the item at position `index` of the linear linked list if 0 <= `index` <= `size()`. Remember, there are two cases for removing a Node from a linear linked list:

1. The request may be to remove the **head** of the linear linked list.
2. The request may be to remove a Node from the **body** of the linked list.

Your remove method must implement these two "cases" and make its "decision" of which one to use based on the value of `index`.

If the given `index` is invalid, the method should display an error message like "index is an invalid index number" where index is the value of `index`.

`public boolean` **`isEmpty`**`()`

Returns true if, and only if `size()` is 0.

`public int` **`size`**`()`

Returns the number of items, or size, of this linear linked list.

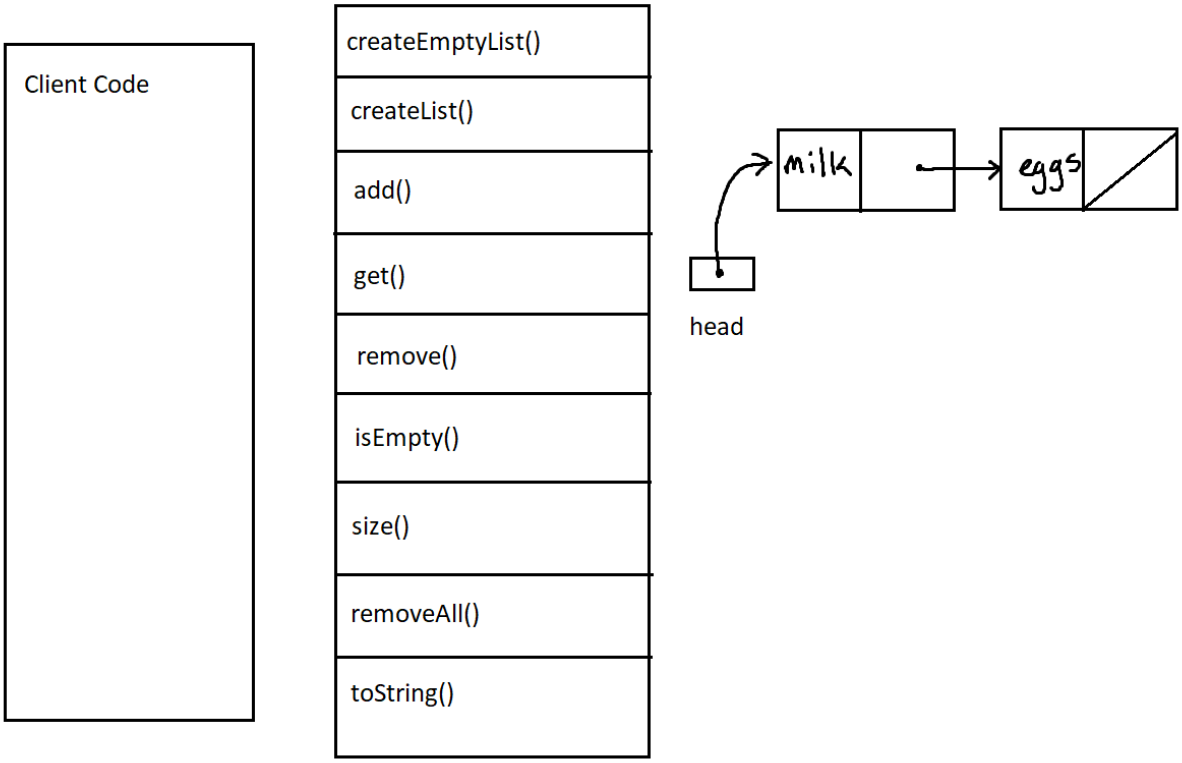
`public void` **`removeAll`**`()`

Removes all of the nodes from this linear linked list by setting the head reference to null and the size variable back to 0.

`public String` **`toString`**`()`

This method should return a String which contains the "index number" and item data field of each node in the linear linked list.

Keep in mind, the basic idea that you're trying to implement is this:



Remember to include a Test class which creates one or more ADTMyList objects and invokes each and every method in your ADTMyList interface. I have included an example test program to give you the idea of how extensive your tests should be. You may use my test program, but be aware that there are somewhere around 40 tests in the program and many of the tests will not work properly until you have completed writing the ADTMyList class.

I have also uploaded a Node class to the Canvas website. **You must use MY Node class in this project**. However, you will have to change the data type of the item data field in this Node class from int to String, and **you must change the universal comment header in my Node class to show that YOU are the programmer who has written this project.**