

BenBaxter | comp2230 | February 2, 2015

Assignment 3(Linking things)

by ben baxter(t00032663)

Problem Definition:

Describe problem

Problem 1

The LinkedList in problem 1 had several missing methods to be completed by the student. The methods had to be written in a general format as both of the next problems classes inherited their methods from the original.

* First()
  + - Pre— there is an element at the head of the list
    - Post –-the method has returned the element stored at head.
    - Big(O)--- 1
* Last()
  + Pre--- there is an element at the tail of the list
  + Post---the method has returned the element stored at the tail
  + Big(O)---1
* RemoveFirst()
  + Pre---there is an element at the head of the list
  + Post---the method returns and removes the element
  + Big(O)---1
* RemoveLast()
  + Pre--- there is an element at the tail of the list
  + Post--- the method returns and removes the element
  + Big(O)---n
* Contains()
  + Pre---a defined target element and a list to check
  + Post--- The Method returns true or false if the target is located
  + Big(O)---n
* isEmpty()
  + Pre--- there is a list
  + Post---returns if the head of the list is null
  + Big(O)---1
* Size()
  + Pre---there is a list
  + Post—returns the size of the count
  + Big(O)---1
* toString()
  + Pre---there is a list with elements
  + Post---the method concatenates the elements into a single string
  + Big(O)---n

Problem 2

The LinkedOrderedList had the incomplete method of Add(). The method needed to implement the comparable function so that it can compare any type of data it is given. The method also had to check for empty strings and deal with the head and tail for lists of size 1 and 0.

* Add()
  + Pre---an element to add and a list to add to.
  + Post---the element is added in the proper order base on the comparable method.
  + Big(O)---n

Problem 3

The LinkedUnorderedList had three incomplete methods. All three methods need to account for size zero and one lists. They also needed to keep track of head and tail assignments.

* addToFront()
  + Pre---an element to add and a list to add to
  + Post----the element is added at the front of the list and becomes the new head
  + Big(O)---1
* addToRear()
  + Pre---an element to add and a list to add to
  + Post----the element is added at the rear of the list and becomes the new tail
  + Big(O)---1
* addAfter()
  + Pre---an element to add and a target element in an existing list to add after
  + Post---the method adds the element after the target element and deals with any remaining elements as well as head or tail conditions.
  + Big(O)---n

Output

