Lab 04 Report: To Do List

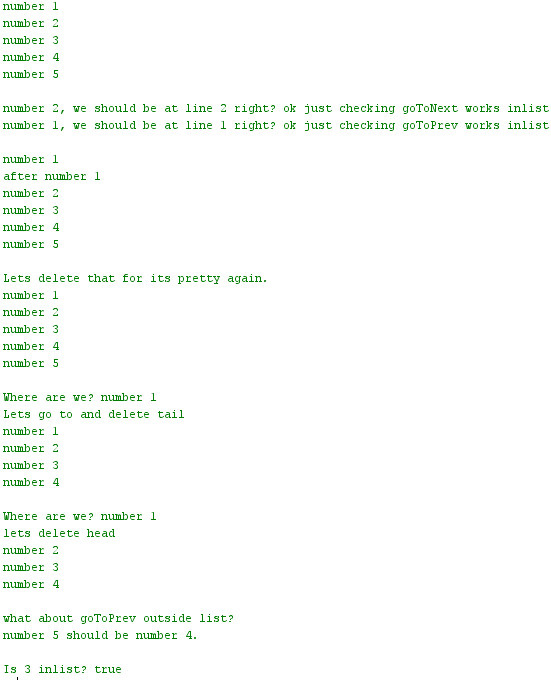
Problem

We had to a write a class called GenCircularLinkedList which is a generic circular linked list- meaning it is a linked list which the element following the tail is the head, and it can take in any type of data. This is just like the last lab except tail.link is set to the head.

Proposed Solution

1. Write a class GenCircularLinkedList
   1. Implement an Internal class ListNode
      1. Include instance variables data of type T, and link of type ListNode
      2. Include a default and parameterized constructor
   2. Three instance variables, head, tail, current, and previous all of type ListNode
   3. Default constructor which initializes head to an empty ListNode, setting current and previous and tail to point at head
   4. Write 9 methods:
      1. goToNext- moves current forward, can’t move forward if next is null
      2. goToPrev- moves backwards, can’t move back if current is the head therefore previous = null.
      3. getDataAtCurrent- returns the current node current.data if current != null.
      4. setDataAtCurrent- takes in a String and sets current.data to String unless current is null.
      5. insert- adds a new node at end of list, if head is null then it starts the list.
      6. insertAfterCurrent- creates a new node based on String passed in, and creates node after current position.
      7. deleteCurrentNode- removes current node by resetting links
      8. showList- prints out the contents of the list line-by-line by using a while loop checking if the next spot is not null.
      9. inList – returns true or false value based on whether or not the data passed in via a parameter is in the list.
2. Test all methods in a driver class which implements an instance of GenCircularLinkedList.
   1. Call all methods using appropriate parameters to demonstrate that each of the 9 methods mentioned earlier work.

Tests and Results



Problems Encountered

I could not get goToPrev to point to the tail if current was at head.

Conclusions and Discussion

This lab was very helpful in learning circular LinkedList and how pointers (.link) works in this lab to see where everything goes or what it is referring to. There are many ways to complete linked list, but for mine a really good insert method was key.

While this solution works, my goToPrev method was not properly written as I did not understand how to to point to the tail when current is at head, but it could just iterate throughout the linkedlist as long as data is not null, when the first thing read is null, set current to the previous.

Additional Questions

1. If the reference to an object is lost and thus the object is now unreachable, what does the Java Virtual Machine (JVM) do with said object?

The JVM throws it away, as it can no longer have a memory link to it.

1. Is it possible to make a circular doubly linked list? If so how can this be achieved, and if not why?

Yes, all elements such as nextLink, prevLink, tail, head, must be accounted for. The tail must still point to the head.