Matthew Curry – March 3, 2019 – CS 150 Lab 5
Lab Notes

Introduction

This lab dealt with creating several classes related in both similarity and through inheritance. The first of these classes is a "MyIntegerNode" class that holds an element of type integer and the next node that it is linked to. The second class is a MyIntegerList that is a linked list of MyIntegerNode's, instantiating a first and last element in the constructor as well as the length of the list. The class contains various methods to add elements to the list, retrieve the size of the list, and convert the list to a string. The next two classes, MyLinkedNode and MyLinkedList are the generic versions of the aforementioned Integer implementations. Further, MyLinkedList is also an abstract class in that it includes an abstract compareTo() method. A MyWeirdList class inherits from this abstract class and implements the compareTo method to compare the length of the Integers of consecutive nodes, as well as a selectionSort method. Finally, there is a Controller class that instantiates an instance of MyWeirdList before adding items to the beginning of the list as well as the end, printing this list, then sorting and printing again.

Method

The first class, the MyIntegerNode, holds an element of type integer as well as a link to the next node in the list. It includes a constructor that takes an Integer and link to a node to instantiate the node. Also, it includes a default constructor that simply sets these values to null. Further, it includes a getElement() method that returns the element in the node, a setElement() that sets the element in the node, a getLink() that returns the link in the node to the next node, and a setLink() that redefines the next node the node is linked to.

The second class is a linked list of MyIntegerNodes, MyIntegerList. It includes methods addBgn() that adds an item to the beginning, addEnd() that adds an item to the end, toString() that converts the list to a string and returns the string, and lastly a size() method that returns the length of the list.

Next, there are two classes that are effectively generic versions of the previously mentioned classes, taking generic objects rather than Integers. These are MyLinkedNode and MyLinkedList, respectively. The MyLinkedList is also abstract; it includes an unimplemented compareTo() method that takes a node and compares the node next to it in the LinkedList.

The MyWeirdList class implements the abstract MyLinkedList class. However, it deals with Integer nodes, defines the compareTo() method to compare number of digits between consecutive nodes, and holds a selectionSort method to sort the list.

Finally, there is a controller class that calls a run method that adds items to the beginning of a weird list, adds items to the end of the list, prints the list, sorts the list, then prints again.

Unit tests
Unit test were completed for both the every class other than the controller and the abstract MyLinkedList. Ultimately, all passed.

Experiments conducted

No experiments conducted.

Results (analysis of the data)

No results.

Conclusion
While all units tests passed, the command line execution ultimately yielded null strings.

Trouble report

The program ultimately yielded null strings rather than the values inputted into the link list.

References

Selection sort with iterator
http://penguin.ewu.edu/~trolfe/LinkedSort/Iterator.html

This keyword

https://www.guru99.com/java-this-keyword.html

Source on the Iterator

https://javapapers.com/core-java/java-iterator/

Nested Classes

https://docs.oracle.com/javase/tutorial/java/javaOO/nested.html

Integer API

https://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html

Exception throwing

https://www.youtube.com/watch?v=pV7AGogWHsk

Implementing a custom iterator

https://www.youtube.com/watch?v=arkoC146TfQ