Data Structures & Algorithms – Chapter 4

Linked Lists

Since you've indicated the concept of the LinkedList is confusing you at a fundamental level, we can discuss it, but at least let me try once to explain it quickly.  Each node has a reference to the next node, just like a line of people who all have their hand on the shoulder of the person in front of them. So let's say you and I are standing in said line and I'm in front. For somebody to cut into the line in between us, like the LinkedListInsertAtPosition operation, you would have to take your hand off my shoulder, and put it on the shoulder of the person who is cutting in line. Then, they would have to put their hand on my shoulder. Those two changes are all that's necessary to "add a link in the chain". So when doing an insert, you just have to locate the person in back and the person in front, and perform the previously stated operations. The only two exceptions (edge cases) are when the person is cutting into the very front or very back of the line, because the case of going to the front of the line, they don't need to put their arm on anybody's shoulder (since they are now in front), and in the case of going to the back of the line nobody needs to put their hand on their shoulder (since there's nobody there to do it).

Node.java looks good.

MainLinkedList.java has one problem. You first create the list and append all the nodes in such an order that you get [D, E, F, A, B, C, G], which is fine, but then you proceed to start to attempt to insert the nodes that already exist in the list into the list again. (For example, the first insert attempts to insert "n1" which is the "A" node already in the list, again into the list, which it is already in, into position 1. While that might not break, it might end up being [D, A, E, F, A, B, C, G], it's still very confusing. If you look at the code example I posted, I'm not appending/inserting the same node twice.

LinkedList.java has a couple problems:

1.) Inside the append method you should not be instantiating a new node. You only need the parameter "node" you are passing.

Come to <http://appear.in/donniesantos> and we can go through the rest.

Also, here's a really good visual example.

<http://www.algolist.net/Data_structures/Singly-linked_list/Insertion>