Single-Linked List

CS 221 Programming Assignment

*“Strange how paranoia can link up with reality now and then.”*– Philip K. Dick

# Objectives

* Create a singly-linked node implementation of the IndexedUnsortedList interface called IUSingleLinkedList.
* Create a fully functional Iterator for your IUSingleLinkedList.
* Expand your test suite to ensure correct functionality of your IUSingleLinkedList and Iterator.

# Tasks

For this homework, you will:

1. Implement a generic typed list class called IUSingleLinkedList.
   * Your IUSingleLinkedList class should use a chain of singly-linked nodes to manage list elements and implement all of the methods required by the IndexedUnsortedList interface.
   * You can use this [Node](https://raw.githubusercontent.com/BoiseState/CS221-Public/master/assignments/DataStructures/files/Node.java) class in your implementation or create your own similar class.
2. Complete your test plan.
   * Implement the remaining change scenarios from your [test plan](https://docs.google.com/document/d/10LZeFle3DBsyoOnUMzv-RsgESrz14IFB1iYs00_NHOA/edit?usp=sharing). In total, you should have at least 21 scenarios from your original test plan in addition to the original given scenarios.
   * Be sure to uncomment the lines in the newList() method of ListTester, so the class can create IUSingleLinkedList objects.
   * Run tests frequently during development to be sure your list class is working properly.
3. Implement a fully functional Iterator for your IUSingleLinkedList class.
   * Your iterator should implement the Java [Iterator](http://docs.oracle.com/javase/8/docs/api/java/util/Iterator.html) interface.
   * Iterator only requires you to implement two of its methods, hasNext() and next(), but you will also implement the optional remove() method.
   * The Iterator documentation says the result of calling its methods after a change has occurred to the list is undefined, because fail-fast behavior cannot be *guaranteed* for all Iterators. *Your* implementation, however, will be expected to throw a [ConcurrentModificationException](http://docs.oracle.com/javase/8/docs/api/java/util/ConcurrentModificationException.html) if any Iterator method is called after the list has been modified by any source other than the current Iterator.
4. Include change scenarios resulting from Iterator's remove() method in your test suite.
   * See scenarios 44-49 in this [expanded set of change scenarios](https://raw.githubusercontent.com/BoiseState/CS221-Public/master/assignments/DataStructures/files/ListTestingScenarios.txt).

# Files

You may use the following class to support creation of your IUSingleLinkedList:

* [Node.java](https://raw.githubusercontent.com/BoiseState/CS221-Public/master/assignments/DataStructures/files/Node.java)

This [IUSingleLinkedList.java](https://raw.githubusercontent.com/BoiseState/CS221-Public/master/assignments/DataStructures/files/IUSingleLinkedList.java) includes the basic class structure and a sample method.

ListTester is the same test class you've been developing through previous assignments ([List Test Part 1](https://docs.google.com/document/d/10LZeFle3DBsyoOnUMzv-RsgESrz14IFB1iYs00_NHOA/edit?usp=sharing), [List Test Part 2](https://docs.google.com/document/d/1zOTB8CkSVN4ntanyxlD3xPtcpnt6h6na_fMTDjRpSFM/edit?usp=sharing), and [ArrayList](https://docs.google.com/document/d/1PYpeaKb1FSlWFdp9d5s4Y1CnFHnuXIdav2cFCOTrNR4/edit?usp=sharing)) for testing implementations of [IndexedUnsortedList](https://raw.githubusercontent.com/BoiseState/CS221-Public/master/assignments/DataStructures/files/IndexedUnsortedList.java)

# Grading

Points will be awarded according to the following breakdown:

| **Tasks** | **Points** |
| --- | --- |
| ListTester testing all change scenarios from your test plan | 20 |
| IUSingleLinkedList class and Iterator functionality and quality | 20 |

# Required Files

Submit the following files:

* IUSingleLinkedList.java
* ListTester.java with all tests from your test plan, plus Iterator tests
* *Any other source files necessary to compile and run your project*

# Submission

Submit all files from the same directory. Do not include any unnecessary files.

Use the submission command given on your section's class web page from the directory containing your files.