Recitation 3: Iterators

Due: See T-Square

Important

There are general assignment guidelines you must always follow. If you fail to follow any of the following guidelines you risk receiving a **0** for the entire assignment.

1. All submitted code must compile under **JDK 8**. This includes unused code, so don't submit extra files that don't compile. This assignment requires Java 8.
2. Do not include any package declarations in your classes.
3. Do not change any existing class headers, constructors, or method signatures.
4. Do not add **additional public methods when implementing an interface.**
5. Do not use anything that would trivialize the assignment. (e.g. don't import/use `java.util.LinkedList` for a Linked List assignment. Ask if you are unsure.)
6. You must submit your source code, the `.java` files, not the compiled `.class` files.
7. Code exactly what is asked for. No more and no less.
8. After you submit your files redownload them and run them to make sure they are what you intended to submit. **You are responsible** if you submit the wrong files.

There are two parts to this assignment:

Circular Linked List Iterator

We have given you code for a fully functioning Circular Linked List. Write an iterator that will allow you to loop through the entire linked list just once (not forever!). To do this, your class will implement Iterable (already added on) and return an Iterator (also added on). To create this iterator, you should use a private inner class that implements Iterator (Also added on, we're too nice to you guys). Remember that an inner class can access its parent class's variables; this will come in handy.

Fibonacci Iterator

Now you will be writing a specific iterator, one that will iterate over the first n Fibonacci numbers. This is just like a standard non-recursive implementation of Fibonacci, but where the next number is calculated only when the user calls next. Do not worry about n being less than 1 for this assignment, assume it will never be.

For a refresher - nth Fibonacci number = $(n-1)$th Fibonacci number + $(n-2)$th Fibonacci number. The first Fibonacci number is 1 and the second is 1 as is reflected in the provided code.

Style

While we will not be grading using CheckStyle, your code should be formatted neatly according to Java conventions.

Provided

The following file(s) have been provided to you.

1. `Node.java` The Node class encapsulates the data and the next pointer for this linked list. **Do not alter this file.**
2. `LinkedListInterface.java` This is the interface for the linked list that is implemented. All instructions for what the methods should do are in the Javadoc. **Do not alter this file.**
3. `CircularLinkedList.java` This is the Linked List implementation provided to you. Fill in the iterator methods in this file.
4. `FibonacciIterator.java` Your Fibonacci sequence iterator.

Deliverables

You must submit all of the following file(s). Please make sure the filename matches the filename(s) below. Be sure you receive the confirmation email from T-Square, and then download your uploaded files to a new folder, copy over the interfaces, recompile, and run. It is your responsibility to re-test your submission and discover editing oddities, upload issues, etc.

1. `CircularLinkedList.java`
2. `FibonacciIterator.java`

You may attach each file individually, or submit them in a zip archive.