Recitation 3: Iterators

Due: See T-Square

Important

There are general submission 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.
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. All methods must be efficient, even if an expected runtime is not specified.
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.

Overview

You are to fill out the methods of two classes: `SquareMatrix` and `FibonacciIterator`.

FibonacciIterator

The Fibonacci sequence is found many places in nature and now also in your assignments. The Fibonacci sequence is 0, 1, 1, 2, 3, ... Each number in the sequence $a_n = a_{n-1} + a_{n-2}$. The `FibonacciIterator` has two methods: `next()` and `hasNext()`. `hasNext()` should return a boolean representing if there is a next value. `next()` should return the next item and increment the iterator to be pointing at the next item. If `next()` is called when there is no next item a `NoSuchElementException` should be thrown. The integer passed in in the constructor (n) tells you how many numbers from the Fibonacci sequence the iterator should return. Note that n can be any non-negative number.

SquareMatrix

The `SquareMatrix` is already completed for you. You need to fill out the iterator. The `SquareMatrix` is backed by a 2D array, matrix. This array stores the data in the form `matrix[row][column]`. The iterator should iterate through the `SquareMatrix` in column major form. This means that it should go through the first column completely, then the second and so on.

ex.
matrix[0][0] = a
matrix[0][1] = c
matrix[1][0] = b
matrix[1][1] = d

The iterator would iterate in the order a, b, c, d.

Javadocs

You do not need to write any Javadocs.

CheckStyle

CheckStyle does not need to be run on this assignment.

Forbidden Statements

You may not use these in your code at any time in CS 1332.

- `break` may only be used in switch-case statements
- `continue`
- `package`
- `System.arrayCopy()`
- `clone()`
- `assert()`
- `Arrays` class
- `Array` class
- `Collections` class
- Reflection APIs

Debug print statements are fine, but should not print anything when we run them. We expect clean runs - printing to the console when we're grading will result in a penalty. If you use these, we will take off points.

Provided

The following file(s) have been provided to you. There are several, but you will only edit two of them.

1. `FibonacciIterator.java` This is the iterator you'll implement that will return the numbers in the Fibonacci sequence. **Do not add any public methods to this file.**
2. `SquareMatrix.java` This is the class whose iterator will go through the matrix. **Do not add any public methods to this file.**
3. `IteratorTests.java` This is the test class that will be used to grade your assignment.

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. `FibonacciIterator.java`
2. `SquareMatrix.java`

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