Recitation 2: Arrays, Stacks, and Queues

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. (Java is backwards compatibile so if it compiles under JDK 7 it *should* compile under JDK 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.

Stacks

A stack is a data structure that follows the "LIFO, Last In First Out" set of rules. You can think of it as a stack of papers (or maybe tickets). If you lay three sheets of paper on top of each other and then want a sheet of paper, you will take the top sheet. The last one that you added to the stack is therefore the first one that leaves it. In your implementation, you will be required to implement push and pop methods using an array as the backing store. See the Javadoc in `ArrayStack.java` for more instructions.

Queues

A queue follows the "FIFO, First In First Out" set of rules. Think of it as a line (maybe at an amusement park). The first person in line expects to be the first one to get his or her ticket. For this homework you will implement an enqueue and dequeue method using two stacks as your backing store. See the Javadoc in `StacksQueue.java` for more instructions.

Javadoc

Javadoc any helper methods you create in a style similar to the Javadoc for the methods in the interface.

JUnits

The JUnits provided will be used to grade your assignment. If you pass all the JUnit tests given, then you have completed the assignment.

Provided

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

1. `StackInterface.java` This is the interface for the stack you will implement. All instructions for what the methods should do are in the Javadoc. **Do not alter this file.**
2. `ArrayStack.java` This is the class that will implement your stack. Look at the Javadoc for the given methods for information on how to implement them. Feel free to add private helpers but **do not add any new public methods.**
3. `QueueInterface.java` This is the interface for the queue you will implement. All instructions for what the methods should do are in the Javadoc. **Do not alter this file.**
4. `StacksQueue.java` This is the queue implementation. Check the Javadoc for instructions and feel free to add additional methods (with Javadoc!). Feel free to add private helpers but **do not add any new public methods.**
5. `StacksAndQueuesTests.java` These are the JUnits 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. `ArrayStack.java`
2. `StacksQueue.java`

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