Project 2: Syeda

DUE: Sunday, October 6th at 11:59pm

Extra Credit Available for Early Submissions!

Basic Procedures

You must:

- Fill out a `readme.txt` file with your information (goes in your user folder, an example is provided)
- Have a style (indentation, good variable names, etc.)
- Comment your code well in JavaDoc style (no need to overdo it, just do it well)
- Have code that compiles with the command: `javac *.java` in your user directory
- Have code that runs with the command: `java Simulate [numPlates > 0]`

You may:

- Add additional methods and variables **where indicated** (each class has its own restrictions). If it doesn't say you *may* add something, then you may not add them.

You may NOT:

- Add any additional methods or instance variables where
- Make your program part of a package.
- Add additional public methods or variables.
- Use any built in Java Collections Framework classes anywhere in your program (e.g. no `ArrayList`, `LinkedList`, `HashSet`, etc.).
- Use any arrays anywhere in your program.
- Alter any method signatures defined in this document of the template code. Note: "throws" is part of the method signature in Java, don't add/remove these.
- Alter provided classes/methods that are complete (`Stack`, `Queue`, `Simulate`, etc.).
- Add any additional import statements (or use the "fully qualified name" to get around import statements).
- Add any additional libraries/packages which require downloading from the internet.

Setup

- Download the `project2.zip` and unzip it. This will create a folder `section-yourGMUUserName-p2`
- Rename the folder replacing `section` with the `001`, `003`, `004` etc. based on the lecture section you are in
- Rename the folder replacing `yourGMUUserName` with the first part of your GMU email address
- After renaming, your folder should be named something like: `001-krusselc-p2`
- Complete the `readme.txt` file (an example file is included: `exampleReadmeFile.txt`)

Submission Instructions

- Make a backup copy of your user folder!
- Remove all test files, jar files, class files, etc.
- You should just submit your java files and your readme.txt
- Zip your user folder (not just the files) and name the zip `section-username-p2.zip` (no other type of archive) following the same rules for `section` and `username` as described above.
  - The submitted file should look something like this:
```
001-krusselc-p2.zip --> 001-krusselc-p2 --> JavaFile1.java
                                            JavaFile2.java
                                            ...
```
- Submit to blackboard.

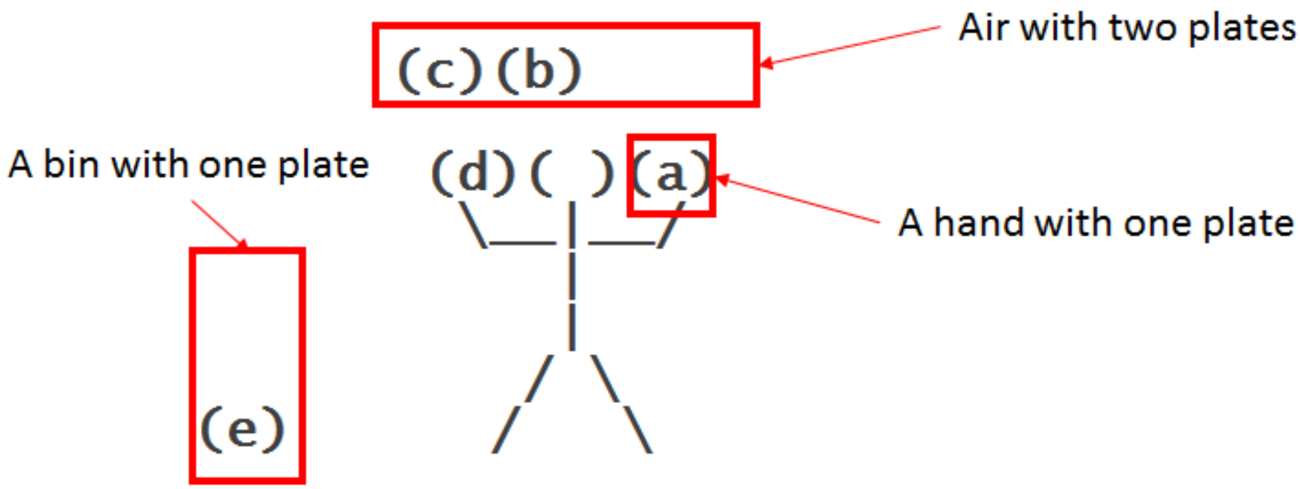
Grading Rubric

Due to the complexity of this assignment, an accompanying grading rubric pdf has been included with this assignment. Please refer to this document for a complete explanation of the grading.

Overview: What Goes Up Must Come Down (Spinning Plates)

You are going to build a small game using linked lists. You will have a plate spinner (Syeda) who is using three data structures to spin plates (her hands, a bin, and the air). Her hands will be modeled with a simple data structure that holds a single plate at a time. The bin will be modeled as a stack data structure which can hold any number of items. The air will be modeled as a queue data structure which can hold a specific number of items. An outline has been provided for these classes in the zip file with this document.

What does the Syeda look like?



There are five plates in the picture above: 2 in the air, 1 in the bin, and 1 in each of Syeda's hands. The bin is a stack, new plates are placed on top and plates are taken off the top. The air is a queue, the front of the queue in this picture is plate (b) and the back is plate (c). Plates are thrown into the back of the queue and taken out of the front of the queue. This produces what is known as the "shower trick" in juggling.

This project is a little different from Project 1: you will not be writing the "driver" program for this, but instead focusing on your data structure implementations. If you complete all the required classes, then the menu, game interactions, and the ASCII part should just work. If it doesn't, there's something wrong with one of your implementations!

Requirements

An overview of the requirements are listed below, please see the grading rubric for more details.

- **Implementing the classes** - You will need to implement required classes and provided template files.
- **Big-O** - The template files provided to you contains instructions on the REQUIRED Big-O runtime for many methods. Your implementation of those methods should NOT have a higher Big-O.
- **Style** – You must follow the coding and conventions specified by the style checker.
- **JavaDocs** - You are required to write JavaDoc comments for all the required classes and methods.

Input Format(s)

No file I/O is required for this project.

Implementation/Classes

This project will be built using a number of classes representing the component pieces of the project. Here we provide a description of these classes. Template files are provided for each class in the project folder and these contain further comments and additional details.

Completed Classes and Interfaces

- **The `Plate` Class (see Plate.java)** This class is complete. It creates plates with letters.

- **The `Stack` Interface (see Stack.java)** This interface is done for you and outlines the required methods for the Bin class.

- **The `Queue` Interface (see Queue.java)** This interface is done for you and outlines the required methods for the Air class.

- **The `Simulate` Class (see Simulate.java)** This class is done for you and you can use it to simulate Syeda's movements. Enjoy the beautiful ASCII art!

Required Classes

- **The `AttachedList<T>` Class (see AttachedList.java)** This class, as outlined, is a linked list. It implements Java's List interface as outlined in the class, you'll also be adding a few new methods: `slice()`, `reverseCopy()`, `pack()`, and `flatten()`.

  - **Note:** Not all the methods in this class are needed by the simulation of Syeda, but they are all graded. The simulation is to give you some practice using these classes (and something fun to look at), but the project itself is about linked lists, stacks, and queues.

- **The `Bin` Class (see Bin.java)** This class implements the interface `Stack<Plate>`. It keeps an `AttachedList` to use as its internal storage.

- **The `Air` Class (see Air.java)** This class implements the interface `Queue<Plate>`. It keeps an `AttachedList` to use as its internal storage.

- **The `Hand` Class (see Spinner.java)** This class can store a single item (a plate). It's the simplest one.

- **The `Spinner` Class (see Spinner.java)** This class does the handoffs (no pun intended) between the bin, the air, and Syed's hands.

Testing

- Empty main methods for testing are given to you, so no test code is provided.[1]
- You can run the simulation of Syeda once everything is done.
  - Note: Testing via the simulation *only* is not a good idea. Many of the methods you need to write are not used by the simulation, but are still a graded part of the project.
- Make sure you're running the style checker and javadoc style checker.

[1] Why are we being so cruel?? So that you are prepared for the future... Think of it this way, if your parents never let you try feeding yourself as a kid and you didn't experience spilling food on the floor a bunch and being sad about it, you wouldn't be the refined diner you are today. Same thing for testing, if you don't learn to do it yourself, you're going to look like a slob when you go to your first job and not the ultra-cool new hotshot you know you are.

How To Handle a Multi-Week Project

While this project is given to you to work on over two and a half weeks, and you are unlikely to be able to complete this in one weekend if you don't plan ahead. We recommend the following schedule:

- Step 1 (Understanding the Code): Before the first weekend (before 9/20)
  - Read project specification, inspect the given template files and get yourself familiar with the sample runs. Make sure to investigate all the methods you'll be writing in each class and start planning which methods can be used to help when writing others.
  - You'll want to review iterators.
- Step 2 (`AttachedList`): First weekend (9/20-9/22)
  - You've learned about linked lists in class last week and had some review this week, so go ahead and complete `AttachedList` if you haven't done so before the weekend. These methods should be familiar to you, but `slice()`, `reverseCopy()`, `pack()`, and `flatten()` might take some additional thought.
  - Test as you go and write as you test. Rather than writing the class from "top to bottom", determine an order to implement the methods which would be easy to test. For example, it would be hard to test `remove()` until after you're sure you have a working `add()`.
  - Also, do the style and comment checking as you go, no one wants to sit around writing JavaDocs for an entire project all at once... don't do that to yourself.
  - If you're struggling... go to office hours as soon as you can!
- Step 3 (`Bin,Air`): Before the second weekend (9/23-9/26)
  - You've learned about stacks and queues in class and are reviewing that material this week, so make the `Bin` and `Air` classes which are stacks and queues! They're very simple classes...
  - Test as you go, do the style and comment checking too.
- Step 4 (`Hand,Spinner`): Second weekend (9/27-9/29)
  - Now you're almost set... finish up by writing the Hand and Spinner classes which connects all the parts together, and run the simulation ☺

This schedule will leave you with the third week and third weekend (9/30-10/6) to perform additional testing, get additional help, and otherwise handle the rest of life. ☺ Also, notice that if you get it done early in the week, you can get extra credit! Check our grading rubric PDF for details. Otherwise you'll have plenty of time to test and debug your implementation before the due date.

Sample Run

```
> java Simulate 5

Syeda, the plate spinner, is learning to do a
shower trick...
She has 5 plate(s) to work with




          ( )
(a)     \__|__/
(b)        |
(c)        |
(d)       / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 1




       (a)( )
        \__|__/
(b)        |
(c)        |
(d)       / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 1

Catching hand not empty
Syeda dropped everything!
Syeda wants to try again...





          ( )
(a)     \__|__/
(b)        |
(c)        |
(d)       / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

       (a)( )
        \__|__/
(b)        |
(c)        |
(d)       / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 2


      (a)

          ( )
        \__|__/
(b)        |
(c)        |
(d)       / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 1


      (a)

       (b)( )
        \__|__/
           |
(c)        |
(d)       / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit
```

```
What should she do? 2


      (b)(a)

          ( )
        \__|__/
           |
(c)        |
(d)       / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 1


      (b)(a)

       (c)( )
        \__|__/
           |
           |
(d)       / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 2


      (c)(b)(a)

          ( )
        \__|__/
           |
           |
(d)       / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit
```

```
What should she do? 1


      (c)(b)(a)

       (d)( )
        \__|__/
           |
           |
          / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 2


      (d)(c)(b)(a)

          ( )
        \__|__/
           |
           |
          / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 3


      (d)(c)(b)

          ( )(a)
        \__|__/
           |
           |
          / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit
```

```
What should she do? 4


      (d)(c)(b)

       (a)( )
        \__|__/
           |
           |
          / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 2


      (a)(d)(c)(b)

          ( )
        \__|__/
           |
           |
          / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 1


      (a)(d)(c)(b)

       (e)( )
        \__|__/
           |
           |
          / \
         /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit
```

```
What should she do? 5


      (a)(d)(c)(b)

          ( )
        \__|__/
           |
           |
          / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 5

Tossing hand was empty
Syeda dropped everything!
Syeda wants to try again...





          ( )
(a)     \__|__/
(b)        |
(c)        |
(d)       / \
(e)      /   \


Syeda can:
 1) Pick up a plate from the bin
 2) Spin a plate into the air
 3) Catch a plate from the air
 4) Pass a plate between hands
 5) Put a plate down on the bin
 6) Quit

What should she do? 6
```