Focus

Your names:

This programming project is to be done by a pair of students. (If your class has an odd number of students, you may have been assigned to a three-person "pair".) Work together, discussing your current problem and how to solve it. Regularly trade off who is driving (actually editing the code) and who is navigating (making suggestions and consulting documentation).

By the time you are done with this activity, you and your pair should be able to:

- implement a double-ended queue using a linked data structure.
- implement the Iterable interface.
- use generic types.

Legal notice

After you complete this activity, please fill out the short survey at

http://goo.gl/forms/HXjyuUb2ou

to improve this project for future users.

Playing the game

Included in the Focus project is a file Focus.jar. This is a compiled version of the working game. To play it, use a terminal to navigate to the directory containing the file and type this on the command line:

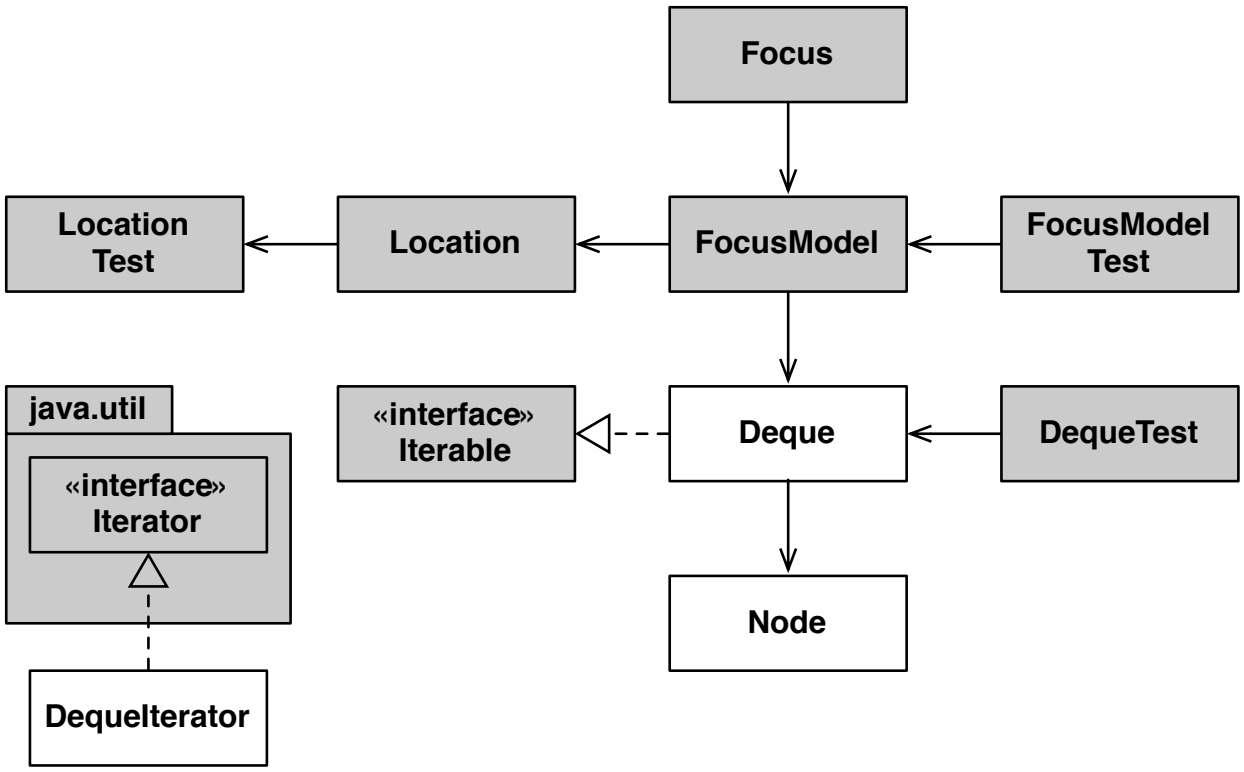
```
java -jar Focus.jar
```

This is a two-player game. Your pair should play a couple of times to get a sense of how the game works. As the rules are somewhat complex, here are some questions to ask yourself as you play:

- Which piles are you allowed to move from?
- What determines the legal destination squares?
- Can you move to the four squares along each side that were initially empty?
- Can you move to the three "missing" squares in each corner of the board?
- Sometimes only part of a pile is moved. What determines how much of the pile is moved?
- Are pieces *taken from* the top or bottom of a pile?
- Are pieces *added* to the top or bottom of a pile?
- What happens if a pile would have more than five pieces?
- How do pieces appear in your reserves?
- When are you allowed to move from reserves?
- To where are you allowed to move from reserves?

Overview

The UML class diagram below shows the relationships between classes. The shaded classes have been provided for you; you should not alter them. Note that Iterable and Iterator are built into Java, the latter in the java.util package. You must write the others from scratch.



Deque

A Deque (pronounced "deck") is a double-ended queue: it's like a queue, but you can add or remove elements from either end. Your primary responsibility is to implement the Deque class. To do this, you will need two other classes, which might reasonably be named DequeIterator and Node.

FocusModel uses Deques to represent piles of pieces. To draw a pile of pieces, Focus iterates through a Deque. Because the piece that is conceptually at the top of the pile has to be drawn last, and iteration goes front to back within the Deque, the back of the Deque corresponds to the top of the pile.

The `move` method in FocusModel uses three Deques: one for the source pile, one for the destination pile, and one for the pieces in the (initially empty) "hand". When *n* pieces are moved:

1. *n* pieces are removed from the back (top) of the source pile. Each one is added to the front (bottom) of the hand.

2. *n* pieces are removed from the front (bottom) of the hand. Each one is added to the back (top) of the destination pile.

3. If the destination pile is now too tall, excess pieces are removed from the front (bottom).

To implement Deque, work through the tests in DequeTest.java in the order in which they appear in that file.

Your implementation of Deque must be linked, not array-based. You may want to start by examining a linked implementation of a regular FIFO queue.

Of the four methods `addBack`, `addFront`, `removeBack`, and `removeFront`, three must run in constant time. The other will require linear time. As you program, you will realize why one of these methods is more difficult than the other three.

Take notes as you work. (This is good work for the navigator.) How well are you working as a pair? What bugs and conceptual difficulties did you encounter? How did you overcome them? What did you learn?

Be sure to play the game (by running Focus.java) a couple of times after you're done to make sure the entire system is working correctly.

After you're done, please fill out the survey at http://goo.gl/forms/HXjyuUb2ou.