Seth Gilbert

CS2020: Algorithms and Data Structures (Accelerated)

Exercise 2: Queues and Tic-Tac-Toe

Exercise 2: Queues and Tic-Tac-Toe

This exercise contains two parts: **Queues** and **Tic-Tac-Toe**. Both parts require using a Java ArrayList. Before you get started, to read the documentation for an ArrayList:

http://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html

Another related aspect of this problem set is using generics in Java: for example, your queue will be parameterized by the type of element to be stored in the queue. Notice that the queue interface itself is paramaterized.

The tic-tac-toe problem also includes an enumerated type. Read the Java documentation for enumerated types here:

http://docs.oracle.com/javase/tutorial/java/javaOO/enum.html

For each part, you need to complete the implementation for the class, and run the associated JUnit test. Once you pass all the tests (and only after you pass all the tests), submit the Java file.

(a) Queue

The problem is simple: implement a queue. Specifically, create a class Queue that properly implements the **IQueue** interface. Your queue should **not** be a fixed-sized queue (like we discussed in class), but instead should accomodate any number of elements. In order to accomplish this, use an **ArrayList** to store the elements in your queue. The **IQueue** interface is attached to this exercise.

(b) Tic-Tac-Toe

Everyone's favorite game, right? Tic-tac-toe! Attached to this exercise is **TacTacToeGame.java**,

which contains all the code you need to play the game. Your job is to implement the **ITicTacToeBoard** interface. Finish the code in the **TicTacToeBoard.java**, and run the **TicTacToeBoardTest.java** JUnit tests. The documentation for the **ITicTacToeBoard** interface can be found in the interface file itself.

Exp:	1
Open at:	30-01-2014 17:00
Close at:	07-02-2014 23:59

Problem Set files :

- Queues.zip
- TicTacToe.zip

Your answer:

```
1
```

Comments

Comment

Save Finalize Submission

Add files for submission