CS162 - Assignment 3: Unit Testing

This assignment is intended to give you practice with using JUnit. This assignment is worth 100 points. The due date is 26 August, by 23:59.

Assignment Overview

You will be using JUnit to implement unit tests to make sure that the BattleshipBoard class works as specified. You will be using the version of the BattleshipBoard class provided below (and not your own). There may be deliberate bugs in the BattleshipBoard class provided. **You may fix the bugs if you like but you are not required to fix them**. Note that a few things have changed in this implementation:

1. The types of exceptions thrown by the methods has changed. Pay close attention to this and make sure to test that the methods throw the right kind of exceptions.
2. The constructor now throws a BattleshipException if numCols or numRows is <= 0.
3. I explicitly require the constructor to set up the board to be empty initially (before ships are placed).

Specifically you will only write tests for the following three functions:

1. **public BattleshipBoard(int numRows, int numCols) throws BattleshipException**
   This is the constructor for the BattleshipBoard class. It creates a board which is initially empty. The constructor requires you to specify the number of rows and number of columns on the board. It throws a BattleshipException if the number of rows is <= 0. It throws a BattleshipException if the number of columns is <= 0.

2. **public void placeShip(int startCol, int startRow, int endCol, int endRow) throws BattleshipException, IndexOutOfBoundsException**
   This function places the battleship starting at position (startCol,startRow) and ending at position (endCol,endRow) inclusive. For example, if you place a ship at (startCol,startRow) = (2,3) to (endCol, endRow) = (4,3), the ship will be at (2,3), (3,3) and (4,3).
   This method throws an IndexOutOfBoundsException if any of the coordinates are out of bounds. A coordinate is in bounds if 0 <= x < (number of columns - 1) and 0 <= y < (number of rows - 1).
   This method throws a BattleshipException if:
   - The ship is placed diagonally. You are only allowed to place battleships vertically or horizontally.
   - The ship cannot be placed successfully because it intersects a board cell that is part of a battleship that has already been placed.
   - startCol > endCol or startRow > endRow

   Note that there is a special case where you could possibly throw either a BattleshipException or an IndexOutOfBoundsException if you have an end coordinate that is negative and less than the start coordinate. In this case, throw a BattleshipException.
3. **public boolean fireShot(int col, int row) throws IndexOutOfBoundsException**
   This function fires a shot at a Battleship located at position (col,row). It returns true if

you hit a battleship and false otherwise. If you fire at the same spot twice, it will return true both times if you hit a battleship. The function throws an IndexOutOfBoundsException if the coordinates are out of bounds.

You do NOT need to test any of the other functions. Please read the comments above each public function and make sure you test that the method behaves as specified. In particular:

- Make sure you test the code when it operates normally and
- Make sure you test the code when it hits error conditions
- Remember to try positive, boundary and negative input values for the parameters to the three methods above.

Please write a test method for each specific test condition you are checking eg. if you want to test that an Exception is thrown for an invalid column coordinate, write a single test method (eg. *testPlaceShipInvalidColCoordinate()*) for it. Do NOT write a test method (eg. *testStuff()*) that tests several things at the same time eg. having a single test method that tests for invalid coordinates AND tests for overlapping ships. This makes it difficult to pinpoint where the error is.

Then, add a brief 1-2 sentence comment above each test method describing exactly what condition it is testing for. You will lose points for forgetting to comment your tests.

Bonus

There is no bonus for this assignment.

Files you will need

- BattleshipBoard.java
- BattleshipException.java

What to Turn In

Please hand in the .java files in your assignment corresponding to the JUnit tests. You do not need to hand in the BattleshipBoard.java file. Please turn them in via Blackboard.

Grading Scheme

We will grade you on the thoroughness of your testing of the three functions.

- Constructor (25 points)
- placeShip (45 points)
- fireShot(25 points)
- Comments and coding conventions (5 points)