CS 122 – Homework #6 – Connect Four – due March 27, 2019

Being able to write a game in a programming language is a lot of fun, and it's also a good milestone for learning about problem solving in general. This is because you need to be able to resolve several issues – how to represent the game (board) information both internally and in the output, how to detect and perform a legal move, and how to detect a winner.

The game Connect Four is a two-player game that has a board with 6 rows and 7 columns. The board is usually kept vertical so that players insert their tokens by selecting a particular column and dropping their piece in the next empty slot in that column. To win the game, you must get 4 of your tokens to line up in any row, column or diagonal, and you must prevent your opponent from doing the same. Your program will be a simulation of this game – we will call the players by single letters (B and Y standing for possible token colors).

The structure of your program should have 2 files. The main() method in Driver.java should involve only the high-level algorithm of whose turn it is, asking the player for which column to go into, and announcing the winner. The nuts and bolts of the implementation should be in a separate file (e.g. Board.java) where a Board class is defined and implemented. I recommend your Board class include the following functions:

- Board() – default constructor that initializes the board. I recommend that you store a period in each board position so that blank positions will be visible when printed out.

- toString() – Return a string representation of the board. Because a loop is needed to create this string, I recommend that you use StringBuilder to help you.

- insert(char player, int column) – given a player and a column number, attempt to place this player's token into the board. The players are not Java programmers, so we will allow the numbers 1-7 to be the legal column numbers. Subtract 1 from this input value to access the proper column of your board. If the column number is out of range or the chosen column is full, return false so that main() can print an error message and ask the player to try again. Hence, this method should return boolean. If you are able to place the player's token in the board, remember to place it in the lowest available space in the column. This is unlike many other games where the user is able to specify both a row and a column.

- boardIsFull() – returns boolean: Determine if every column is full. This is useful when you want to know that the game has ended in a tie.

- findWinner() – see if the player who just went has won the game. For example, this method could return char. The return value could be 'B' if B won, or 'Y' if Y won, or ' ' if there is no winner yet.

The main() method in the Driver class should call the Board constructor to create an instance of the game. When the game starts, player B goes first. With error checking, ask the current player to make a move. Once the move is made, print out the board. Then determine if there is a winner, or if the game has just ended in a tie. If the game is over, print a message announcing the final outcome. Otherwise, switch to the other player and go to the next turn of the game.

Your program's output should look like this example I/O:

```
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .


Player B, which column do you want to try? (1-7)  4
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . B . . .


Player Y, which column do you want to try? (1-7)  7
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . B . . Y


Player B, which column do you want to try? (1-7)  3
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . B B . . Y


Player Y, which column do you want to try? (1-7)  8
Sorry, the column you chose is not a valid choice; try again.

Player Y, which column do you want to try? (1-7)  4
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . Y . . .
. . B B . . Y


Player B, which column do you want to try? (1-7)  5
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . Y . . .
. . B B B . Y
```

```
Player Y, which column do you want to try? (1-7)  3
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . Y Y . . .
. . B B B . Y


Player B, which column do you want to try? (1-7)  0
Sorry, the column you chose is not a valid choice; try again.

Player B, which column do you want to try? (1-7)  6
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . Y Y . . .
. . B B B B Y


The winner is B!
```