

Lab 05: Connect-Four

Overview

This lab is designed to introduce students to 2-D arrays by recreating one of everyone's favorite childhood games: Connect-Four. You will loop through arrays, and manipulate them. Your end product should be robust enough to not have a single `ArrayIndexOutOfBoundsException` Exception!

Specification

You will first start by asking the user for what they wish the height and length of the board to be:

```
What would you like the height of the board to be? 4
What would you like the length of the board to be? 5
```

Then you will print the empty board:

```
- - - - -
- - - - -
- - - - -
- - - - -
```

And tell the players what their tokens are:

```
Player 1: x
Player 2: o
```

The players will take turns placing their tokens by choosing columns...

```
Player 1: Which column would you like to choose? 0
```

```
- - - - -
- - - - -
- - - - -
x - - - -
```

```
Player 2: Which column would you like to choose? 3
```

```
- - - - -
- - - - -
- - - - -
x - - o -
```

...until one of them wins!

```
Player 1: Which column would you like to choose? 0
```

```
x - - - -
x - - - -
x - o - -
```

x o x o o

Player 1 won the game!

Or until there is a tie!

Player 2: Which column would you like to choose? 2

o x o x o
x o o x x
x o o o x
x o x o x

Draw. Nobody wins.

Elements in the array should be accessible via row-major indexing (**board[row][column]**). In addition, the data should be stored so that **row zero is the bottom of the board**, i.e.:

Row 3	x - - - -
Row 2	x - - - -
Row 1	x - o - -
Row 0	x o x o o

Make sure you test this along the way! Otherwise, your method tests in ZyBooks will fail!

Assumptions

Students can assume that:

- the user will choose for the board dimensions to be 4x4 or greater.
- the user will input a valid column number (from 0 to length-1).
- the column that the user choses to place their token into has space (it is not filled already by other tokens).
- players can only win vertically or horizontally, but not diagonally.

Required Methods

public static void printBoard(*char*[][] array)

This will print the board.

public static void initializeBoard(*char*[][] array)

This will set each spot in the array to “-”.

public static int insertChip(*char*[][] array, *int* col, *char* chipType)

Places the token in the column that the user has chosen. Will find the next available spot in that column if there are already tokens there. The row that the token is placed in is returned.

public static boolean checkIfWinner(*char*[][] array, *int* col, *int* row, *char* chipType)

After a token is added, checks whether the token in this location, of the specified chip type, creates four in a row. Will return true if someone won, and false otherwise.

Hint: Implement the methods in this order.

Submission

NOTE: Your output must match the example output **exactly**. If it does not, *you will not receive full credit for your submission!*

Files: ConnectFour.java
Method: Submit on ZyLabs

Sample Output

What would you like the height of the board to be? 4
What would you like the length of the board to be? 5

```
- - - - -  
- - - - -  
- - - - -  
- - - - -
```

Player 1: x
Player 2: o

Player 1: Which column would you like to choose? 0

```
- - - - -  
- - - - -  
- - - - -  
x - - - -
```

Player 2: Which column would you like to choose? 3

```
- - - - -  
- - - - -  
- - - - -  
x - - o -
```

Player 1: Which column would you like to choose? 0

```
- - - - -  
- - - - -  
x - - - -  
x - - o -
```

Player 2: Which column would you like to choose? 1

```
- - - - -  
- - - - -  
x - - - -  
x o - o -
```

Player 1: Which column would you like to choose? 0

```
- - - - -  
x - - - -  
x - - - -  
x o - o -
```

Player 2: Which column would you like to choose? 4

```
- - - - -  
x - - - -
```

```
x - - - -  
x o - o o
```

Player 1: Which column would you like to choose? 2

```
- - - - -  
x - - - -  
x - - - -  
x o x o o
```

Player 2: Which column would you like to choose? 2

```
- - - - -  
x - - - -  
x - o - -  
x o x o o
```

Player 1: Which column would you like to choose? 0

```
x - - - -  
x - - - -  
x - o - -  
x o x o o
```

Player 1 won the game!

Process finished with exit code 0