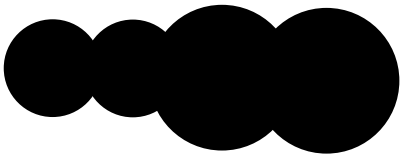
Assignment 10: Connect Four



Preparation

You are expected to be familiar with the following textbook sections and lectures *before* beginning the assignment. You are required to read the assignment in advance.You are required to submit external documentation with your code. You are also required to write doc comments (Javadoc) for each method and for the class itself.

Textbook sections

- 8.2-8.4

Most relevant lectures

- L11 (Javadoc), L17

Exercises

1. Read through the assignment.
2. Think about how to store the current state of a Connect Four game using a 2D array.
3. Figure out how to check if a win or a draw has occurred in the game, by checking your 2D array.
4. Make a top-down design diagram for `ConnectFour`.

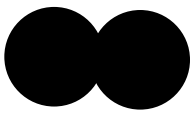
Introduction

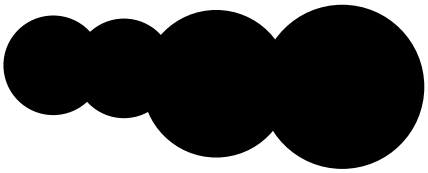
You may work in pairs. You will program a two player game called Connect Four. From Wikipedia:

> Connect Four [...] is a two-player connection game in which the players first choose a color and then take turns dropping colored discs from the top into a seven-column, six-row vertically suspended grid. The pieces fall straight down, occupying the next available space within the column. The objective of the game is to connect four of one's own discs of the same color next to each other vertically, horizontally, or diagonally before your opponent.

Specifications

- Write a program called `ConnectFour`.
- Your program should allow two (human) players to play the game Connect Four.
- The program should repeat the following steps until a win or a draw:
    1. Prompt the correct player to drop a disk into a column from $0-6$. Accept and validate user input.





2. Update the board.
3. Display the updated board.
4. Check the status of the game (continue, win, or draw).
   - It might be more efficient to check for a win immediately each time a valid move is made. Then you won't have to check the whole board, and just wins involving the most recent move.

- If there is a win, display a message saying which player won. If there is a draw, display an appropriate message.
- Your program must deal with invalid input. If the user enters an invalid column number, prompt again. Invalid entries include:
  - Incorrect type.
  - Incorrect values.
  - Already full columns.

Documentation

You are required to:

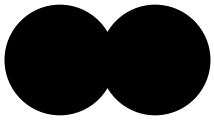
1. Write proper Javadoc comments for your methods and your class. Make sure to run `javadoc` yourself to see that the html page gets properly generated and lists all your methods and descriptions.
2. Write external documentation for the `ConnectFour` program, similar to the external documentation example posted to D2L, but also including your top-down design diagram. Submit a `.pdf` file. You do not need to include algorithms for all of your methods. Just include a general algorithm for your `main` or top-level methods.

Submission

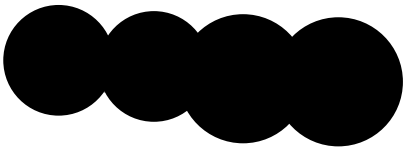
Recall that submission instructions are in the **Lab Guide**. Your group is required to submit **one** `.zip` folder (in one person's D2L dropbox) containing:

- the internally documented (including Javadoc comments for each method and class) and properly styled source code file, `ConnectFour.java`
- external documentation for `ConnectFour`, including top-down design diagram, in a `.pdf`

Do not submit the html files generated by `javadoc`! The submitted files should contain both partners' names in the headers. *Make sure both partners save copies of the finished code to their personal H: drive.*



Assignment 10: Connect Four

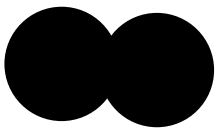


Sample Output

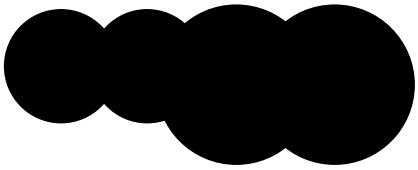
```
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
Select an empty column (0-6) to drop a yellow disk into:
2
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | |Y| | | | |
Select an empty column (0-6) to drop a yellow disk into:
t
Sorry, you have entered an invalid column number.
Select an empty column (0-6) to drop a red disk into:
3
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | |Y|R| | | |
Select an empty column (0-6) to drop a yellow disk into:
8
Sorry, you have entered an invalid column number.
Select an empty column (0-6) to drop a yellow disk into:
0
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
|Y| |Y|R| | | |

⋮

| | | | | | | |
| | | | | | | |
| | | | | | | |
| | |R| | | | |
|R|Y|Y|Y| | | |
|Y|R|Y|R|R| | |
```



Assignment 10: Connect Four



```
Select an empty column (0-6) to drop a yellow disk into:
4
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | |R| | | | |
|R|$|$|$|$| | |
|Y|R|Y|R|R| | |
Congratulations, the yellow player wins!!!
```