TASK

In the game of tic-tac-toe, the game board consists of a grid of 3x3 entry elements. Players alternate in placing X or O on the board; the first player to have 3 characters in a line (row, column, or diagonal) wins the game. The game ends in a draw if there is no winner and the board is fully occupied.

In-class lab from Week 9 accomplished the printing of the board. In this lab we will continue to develop the game. First we attempt to do it as a monolithic program that is included in the project class' main method. Then we take advantage of modular design and use classes to organize the program more efficiently.

INSTRUCTIONS

Part 1 - procedural:

1) Lay out the program's logic for one round of game in pseudo-code:
   a) initialize the board to hold only spaces
   b) print the board
   c) player X's turn
      i) prompt for entering a position (1-9)
      ii) read keyboard input
      iii) check for valid input and repeat steps i-iii until a valid number is entered
      iv) check for valid playing position (must be empty) and repeat steps i-iv until a valid position is entered
      v) enter the player's character on the board
   d) print the board
   e) end the game if player won or there is a draw
   f) player O's turn
      same as for player X except a different playing character is used (O instead of X)
   g) print the board
   h) end the game if player won or there is a draw

2) Turn the pseudo-code into java code by reusing the Week 9 code for initialization and printing of the board. Reuse the code for player X to make player O. Skip the code for checking wins or draws for now.
3) Add an outside loop for making a one-turn game into an infinite-turn game, relying on the win/draw checking code to stop the program (use a "while(true)" loop).

   Now is a good time to realize that organizing code this way leads to inefficient and error-prone code. Let's re-factor the code and use an object-oriented approach with classes.

Part 2 - object-oriented:

1) Create a new project.
2) We'll use 2 helping classes (game and player) and one "driver" class with the main method - the project's class where the program execution begins.
3) Right-click on the package icon (under source packages in the project's window) and add a new class to the package called Game, and another one called Player. There should now be 3 java files in the package, and they should be open in the editor.
4) Decide on the responsibilities of the classes. There are numerous ways to organize the code, here is just one possibility:
   a) the driver class (with main) should provide declarations, initializations, and the main loop for the rounds:
      i) initialize the board and print it
      ii) while (true) {
          player X plays
          print board
          check for win or draw
          player O plays
          print board
          check for win or draw
          }

   b) Game class
      i) includes the board array (public)
      ii) includes initialization of the board method (public)
      iii) includes the printing of the board method (public)
      iv) includes the winning condition and draw checking method (public).
          When a win or draw occurs, program uses System.exit(0) to stop execution and terminate the infinite while(true) loop.
          Hint: you will need to pass a player's character symbol to this method so that the method can check for a win

   c) Player class
      i) includes the player's playing character (X or O) (public)
      ii) includes a play method (public) that takes care of user input and puts the character on the board
          Hint: you will need to pass the game's board to this method so that the method can check for a valid position (unoccupied) and to place the player's character on the board

5) Code the classes and the driver class. You can assume throughout the program that the game is always a 3x3 board. The code can be made general to take care of a board of an arbitrary size but that complexity is not necessary for this exercise.

6) Upload your **project file** to the Week 11 in-class lab dropbox. Use **File>Export Project>To Zip...**