Lab Assignment 3

Start Assignment

Due Tomorrow by 23:59 **Points** 20 **Submitting** a file upload **File types** zip **Available** after 25 Mar at 0:00

Tic-Tac-Toe

You will be provided with some skeleton code for an implementation of the tic-tac-toe game in Java. See **here** **(https://en.wikipedia.org/wiki/Tic-tac-toe)** if you are unfamiliar with the game and its rules.

Resources

You will be provided with the following classes:

- **TicTacToeGame**: A class that implements a game of Tic-Tac-Toe. It uses the `TicTacToeBoard` and `TicTacToePlayer` classes.
- **TicTacToePlayer**: A class representing a player in the game of Tic-Tac-Toe.
- **TicTacToeBoard**: A class representing the board for a game of Tic-Tac-Toe

Instructions

This is an out-of-class **individual** lab activity. You are allowed to discuss problems in your study groups for this lab.
That said, **submit individual work**, and state in a block comment at the top of your .java files who you may have discussed the problem with.

The TicTacToeBoard class will require a declaration and initialization and/or instantiation of a 3x3 array of characters to represent the game board.

All other classes declare methods with empty or incomplete bodies or return a sentinel value. Your task will is to implement these methods in order to end up with a fully functional game.

You will have to implement/complete the following methods

- `TicTacToeGame` class:
  - `play()`
- `TicTacToeBoard` class:
  - `TicTacToeBoard()`
  - `isBoardFull()`
  - `isEmpty()`
  - `play()`
  - `hasWinner()`
  - `getWinningSymbol()`
  - `detectWin()`

Submission Instructions

What to submit:

- Your .java files

It is very important that you follow the submission instructions below:

1. Create a folder on your computer called **Firstname.Surname-Lab3** (where *Firstname* and *Surname* are your first name and surname, respectively). For example, a student called 'Ian Joseph Akotey', will create a file called 'Ian.Akotey-Assign3.zip'
2. Copy your code (the .java files, not the .class file or the .java~ file) into the folder you have created.

3. Right-click on the folder containing your code and document, and select “zip” or “compress” (or the appropriate command on your computer) to zip it up. This will result in a file called **Firstname.Surname-Lab3.zip**.

4. Upload your zip file to Canvas.

Code

TicTacToeGame

```
import java.util.Scanner;

/**
 * This program implements a game of Tic Tac Toe.
 * It uses the TicTacToeBoard and TicTacToePlayer classes.
 *
 * @author G. Ayorkor Korsah
 */
public class TicTacToeGame {

    /**
     * The function asks the user for a row and column,
     * and then attempts to play the player's symbol
     * at that position.
     * If the play is successful, the board is printed.
     * If the play is not * successful, the user is asked to try again
     *
     * @param board The board object that the game is being played on.
     * @param player The player who is making the move.
     */
    private static void play(TicTacToeBoard board, TicTacToePlayer player) {
        Scanner input = new Scanner(System.in);
        int row, col;
        boolean playSuccessful;
```

```java
        // TODO: Complete the play method
        // Allow the player to play on the board

        System.out.println("The board now looks like this: ");
        board.printBoard();
    }

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        TicTacToeBoard board;
        TicTacToePlayer player1, player2; // The two players of the game
        TicTacToePlayer curPlayer; // a reference to the current player
        int whoseTurn; // a number to keep track of the player turn

        board = new TicTacToeBoard();
        player1 = new TicTacToePlayer();
        player2 = new TicTacToePlayer();

        System.out.println("Player 1, please enter your information: ");
        player1.getPlayerInfo();

        System.out.println("Player 2, please enter your information: ");

        do {
            player2.getPlayerInfo();
            if (player2.getName().equalsIgnoreCase(player1.getName()))
                System.out.println("Sorry, that name is being used by Player 1.");
            if (player2.getSymbol() == player1.getSymbol())
                System.out.println("Sorry, that symbol is being used by Player 1.");
        } while (player2.getName().equalsIgnoreCase(player1.getName()) ||
                player2.getSymbol() == player1.getSymbol());

        System.out.println("Okay, " + player1.getName() +
                " is player 1 and will use symbol " + player1.getSymbol());
        System.out.println(player2.getName() + " is player 2 and will use symbol " +
```

```
                player2.getSymbol());

        System.out.print("Who will go first? (Enter 1 or 2) ");
        whoseTurn = input.nextInt();
        if (whoseTurn == 1)
            System.out.println(player1.getName() + " will go first.");
        else
            System.out.println(player2.getName() + " will go first.");

        System.out.println("Initially, the board looks like: ");
        board.printBoard();

        do {
            if (whoseTurn == 1)
                curPlayer = player1;
            else
                curPlayer = player2;

            play(board, curPlayer);
            whoseTurn = whoseTurn % 2 + 1;

        } while (!board.isBoardFull() && !board.hasWinner());

        if (board.hasWinner()) {
            if (board.getWinningSymbol() == player1.getSymbol())
                System.out.println(player1.getName() + " wins!");
            else
                System.out.println(player2.getName() + " wins!");
        } else
            System.out.println("There is no winner.");

        input.close();
    }
}
```

TicTacToePlayer

```java
import java.util.Scanner;

/**
 * This class represents a player in the game of Tic-Tac-Toe.
 *
 * @author G. Ayorkor Korsah
 */
public class TicTacToePlayer
{
  private String name;
  private String symbol;

  /**
   * This method reads in the player's name and desired symbol.
   * It allows only non-digit single-character symbols.
   */
  public void getPlayerInfo() {
    Scanner input = new Scanner(System.in);
    String pattern = "[^\\d]";
    String answer;

    System.out.print("What is your name? ");
    name = input.next();

    do {
      System.out.print("What symbol would you like to use? ");
      answer = input.next();
      if (!answer.matches(pattern)){
        System.out.println("Your symbol must be exactly one character, " +
                           "and cannot be a digit");
      } else
        symbol = answer;
```

```
    } while (!answer.matches(pattern));
  }

  /**
   * Retrieves the name of the player.
   */
  public String getName(){
    return name;
  }

  /**
   * Retrieves the symbol of the player.
   */
  public String getSymbol(){
    return symbol;
  }
}
```

TicTacToeBoard

```
import java.util.Scanner;

/**
 * This class represents the board for a game of
 * TicTacToe
 *
 * @author G. Ayorkor Korsah
 */
public class TicTacToeBoard {

    private String[][] board; // the board
    private int numEmpty; // The number of empty slots
```

```java
    private boolean detectedWin; // Whether or not a winner has been seen
    private String winningSymbol; // The symbol of the winner
    public static final int SIZE = 3;
    public static final String EMPTY = ".";

    /**
     * The constructor for the class.
     * It instantiates the 3x3 board and initializes the board to be empty.
     * You should also initialize the number of empty slots
     *
     */
    public TicTacToeBoard() {
        // TODO: Complete the method

        // This part below has been done for you, for free 😄
        detectedWin = false;
        winningSymbol = "";
    }

    /**
     * It prints the board
     */
    public void printBoard() {
        System.out.print("  ");
        for (int i = 0; i < board[0].length; i++) {
            System.out.print(i + " ");
        }
        System.out.println();
        for (int i = 0; i < board.length; i++) {
            System.out.print(i + " ");
            for (int j = 0; j < board[i].length; j++) {
                System.out.print(board[i][j] + " ");
            }
            System.out.println();
        }
    }
```

```java
    /**
     * If the number of empty spaces is 0, then the board is full
     *
     * @return whether or not the board is full.
     */
    public boolean isBoardFull() {
        return false; // TODO: Correct the method
    }

    /**
     * Return true if a given location on the board,
     * specified by a given row index and column index, is empty
     *
     * @param row The row of the board.
     * @param col the column of the board
     * @return Whether or not the location is empty.
     */
    public boolean isEmpty(int row, int col) {
        return false; // TODO: Correct the method
    }

    /**
     * check the game for a win
     * If all the elements a row, column, or diagonal are the same,
     * then a win has been detected.
     *
     * If a win is detected, the detectedWin instance variable should be set to true,
     * and the winningSymbol instance variable should be set to the symbol of the winner.
     *
     * Hint: There are 8 ways to win a 3x3 game of tic-tac-toe
     */
    private void detectWin() {
        // TODO: Correct the method
    }

    /**
     * If the board is empty at the given row and column,
```

```java
     * then place the symbol on the board,
     * decrement the number of empty spaces,
     * and check for a win
     *
     * If the board is not empty at the given row and column,
     * then the play is not successful and the method returns {@code false}.
     *
     *
     * @param row the row of the board (0 indexed)
     * @param col The column number of the board (0 indexed).
     * @param symbol the symbol to be placed on the board
     * @return Whether a move was successful or not.
     */
    public boolean play(int row, int col, String symbol) {
        // TODO: Implement the method
        return false; // TODO: Correct the method
    }

    /**
     * If the game is over,and has a winner, return true. Otherwise, return false
     *
     * @return The boolean value of detectedWin.
     */
    public boolean hasWinner() {

        return false; // TODO: Correct the method
    }

    /**
     * This function returns the winning symbol.
     * It is only valid to call this method if hasWinner returns true.
     *
     * @return The winningSymbol.
     */
    public String getWinningSymbol() {
        return EMPTY; // TODO: Correct the method
    }
```

```java
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        TicTacToeBoard board = new TicTacToeBoard();
        board.printBoard();

        int row, col;
        boolean xTurn = true;
        while (!board.isBoardFull() && !board.hasWinner()) {
            System.out.print("Enter row & col to play: ");
            row = input.nextInt();
            col = input.nextInt();
            board.play(row, col, xTurn ? "X" : "O");
            board.printBoard();
            if (board.isBoardFull())
                System.out.println("Board is full.");
            if (board.hasWinner())
                System.out.println("Has winner: " + board.getWinningSymbol());
            xTurn = !xTurn;
        }
        System.out.println("Goodbye!");
        input.close();
    }
}
```