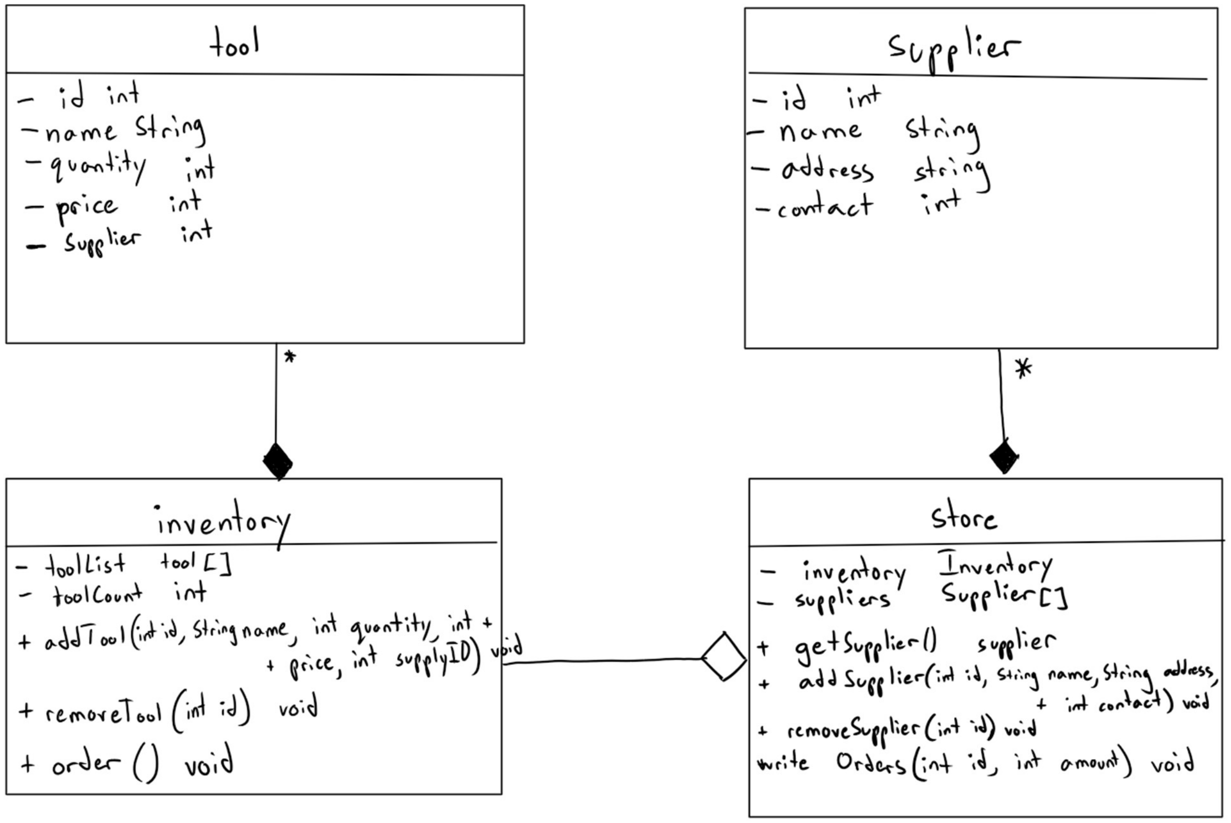
Hunter Coles

ENSF 409

30078896

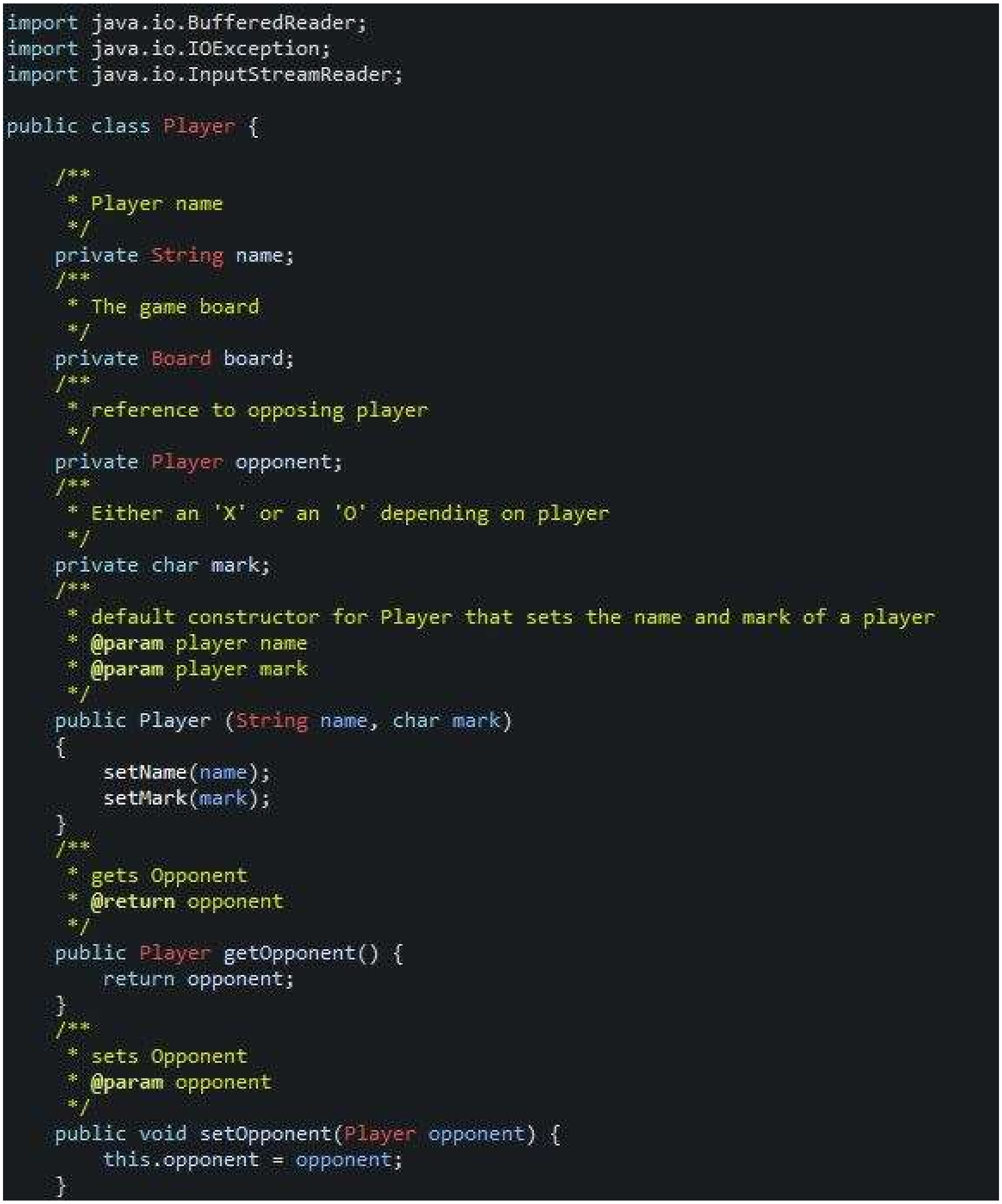
Exercise 3



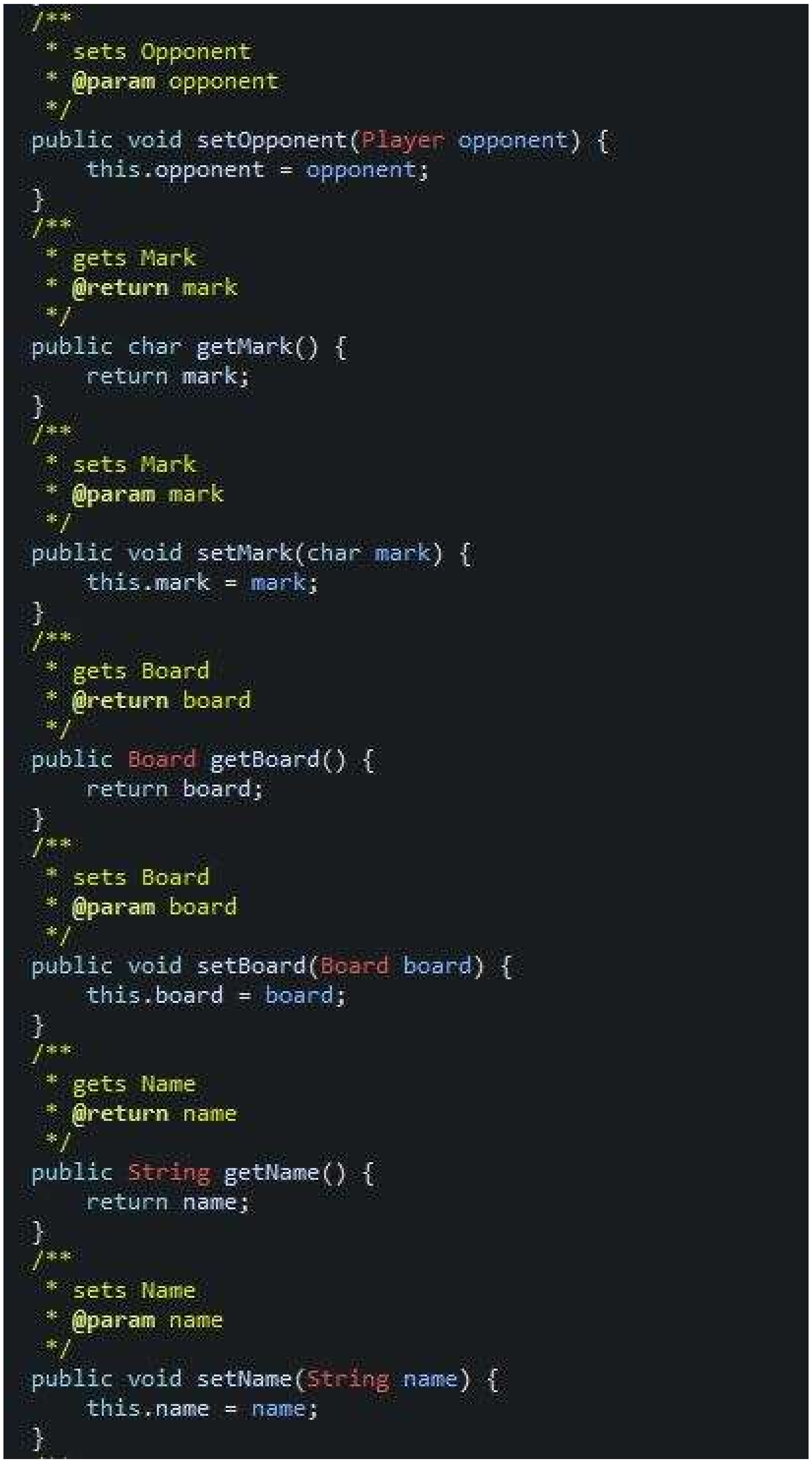
Exercise 4

Code: Game, Constants & Board were not modified so they were not submitted as part of the report

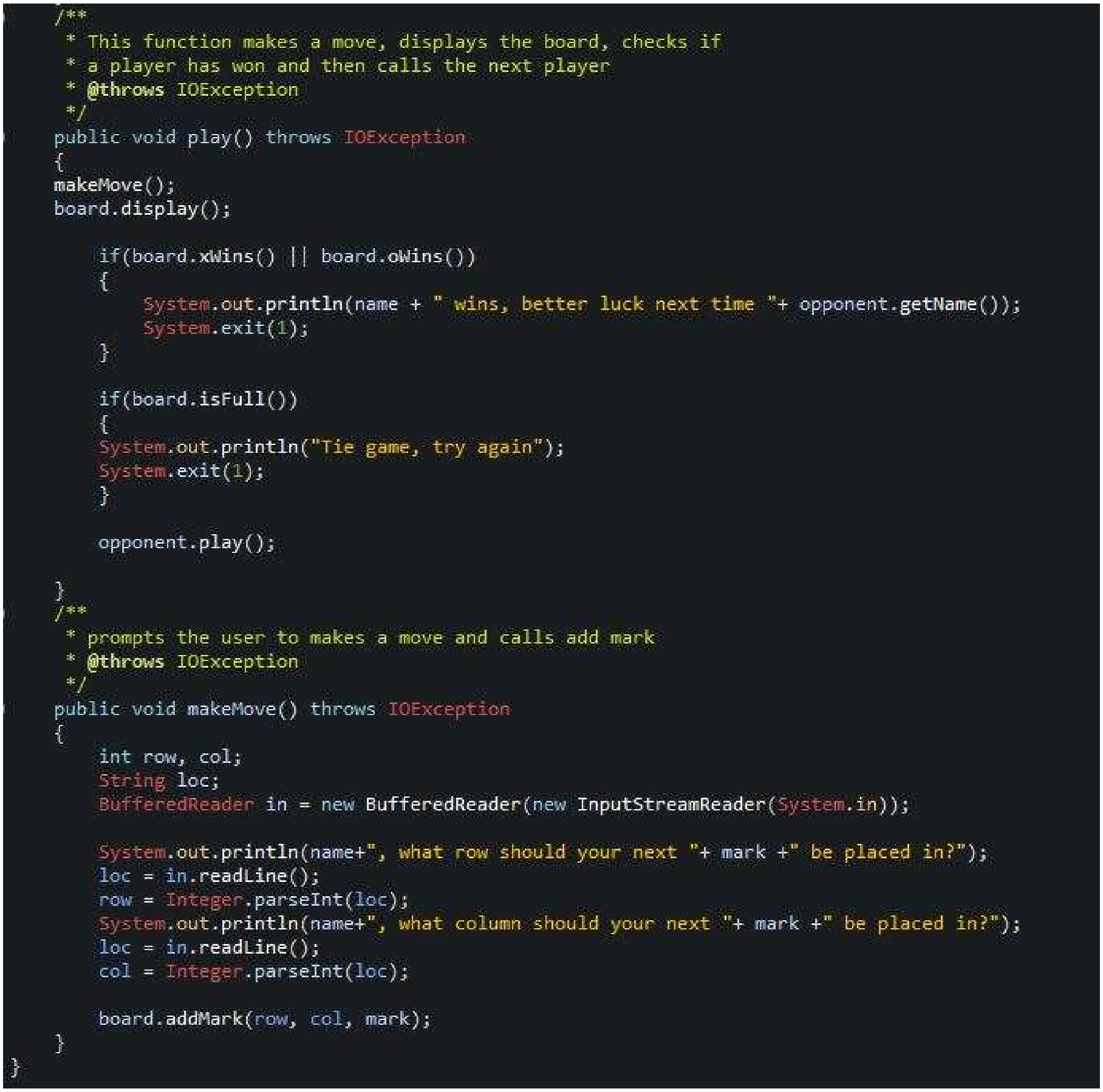
Player:



```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Player {

    /**
     * Player name
     */
    private String name;
    /**
     * The game board
     */
    private Board board;
    /**
     * reference to opposing player
     */
    private Player opponent;
    /**
     * Either an 'X' or an 'O' depending on player
     */
    private char mark;
    /**
     * default constructor for Player that sets the name and mark of a player
     * @param player name
     * @param player mark
     */
    public Player (String name, char mark)
    {
        setName(name);
        setMark(mark);
    }
    /**
     * gets Opponent
     * @return opponent
     */
    public Player getOpponent() {
        return opponent;
    }
    /**
     * sets Opponent
     * @param opponent
     */
    public void setOpponent(Player opponent) {
        this.opponent = opponent;
    }
```

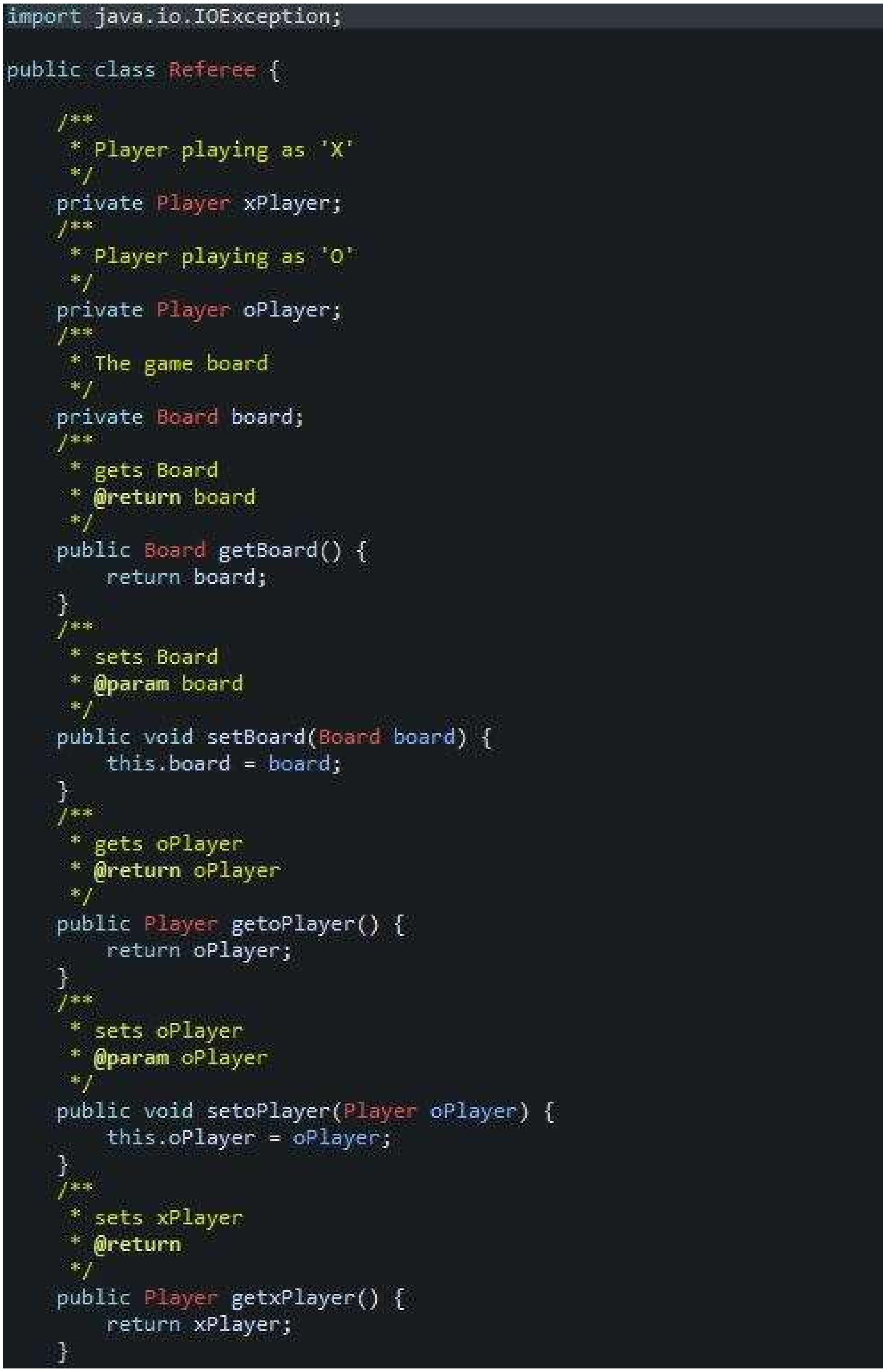


```
/**
 * sets Opponent
 * @param opponent
 */
public void setOpponent(Player opponent) {
    this.opponent = opponent;
}
/**
 * gets Mark
 * @return mark
 */
public char getMark() {
    return mark;
}
/**
 * sets Mark
 * @param mark
 */
public void setMark(char mark) {
    this.mark = mark;
}
/**
 * gets Board
 * @return board
 */
public Board getBoard() {
    return board;
}
/**
 * sets Board
 * @param board
 */
public void setBoard(Board board) {
    this.board = board;
}
/**
 * gets Name
 * @return name
 */
public String getName() {
    return name;
}
/**
 * sets Name
 * @param name
 */
public void setName(String name) {
    this.name = name;
}
```

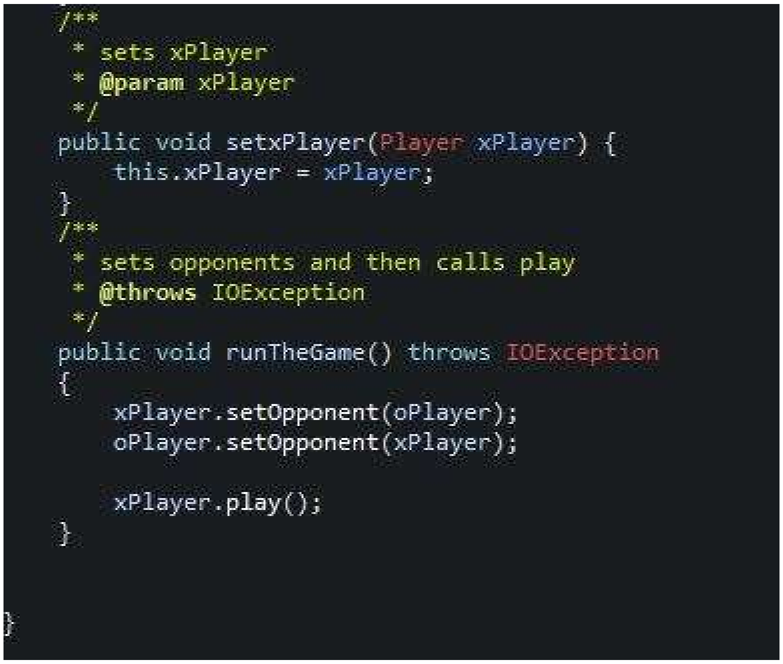


```
    /**
     * This function makes a move, displays the board, checks if
     * a player has won and then calls the next player
     * @throws IOException
     */
    public void play() throws IOException
    {
    makeMove();
    board.display();

        if(board.xWins() || board.oWins())
        {
            System.out.println(name + " wins, better luck next time "+ opponent.getName());
            System.exit(1);
        }

        if(board.isFull())
        {
        System.out.println("Tie game, try again");
        System.exit(1);
        }

        opponent.play();

    }
    /**
     * prompts the user to makes a move and calls add mark
     * @throws IOException
     */
    public void makeMove() throws IOException
    {
        int row, col;
        String loc;
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

        System.out.println(name+", what row should your next "+ mark +" be placed in?");
        loc = in.readLine();
        row = Integer.parseInt(loc);
        System.out.println(name+", what column should your next "+ mark +" be placed in?");
        loc = in.readLine();
        col = Integer.parseInt(loc);

        board.addMark(row, col, mark);
    }
}
```

Referee:

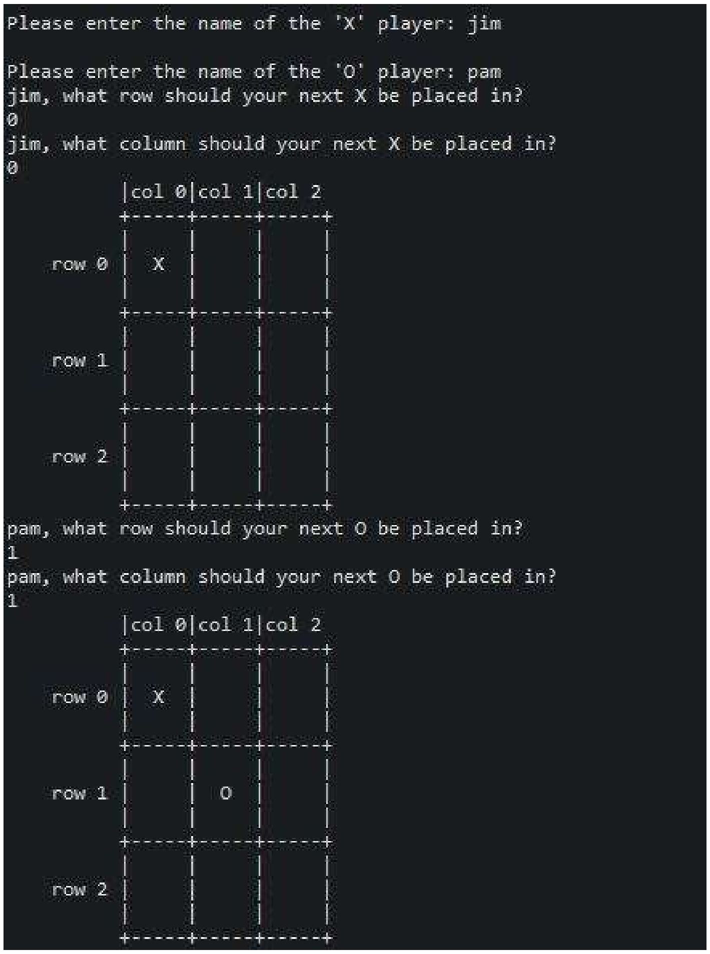


```java
import java.io.IOException;

public class Referee {

    /**
     * Player playing as 'X'
     */
    private Player xPlayer;
    /**
     * Player playing as 'O'
     */
    private Player oPlayer;
    /**
     * The game board
     */
    private Board board;
    /**
     * gets Board
     * @return board
     */
    public Board getBoard() {
        return board;
    }
    /**
     * sets Board
     * @param board
     */
    public void setBoard(Board board) {
        this.board = board;
    }
    /**
     * gets oPlayer
     * @return oPlayer
     */
    public Player getoPlayer() {
        return oPlayer;
    }
    /**
     * sets oPlayer
     * @param oPlayer
     */
    public void setoPlayer(Player oPlayer) {
        this.oPlayer = oPlayer;
    }
    /**
     * sets xPlayer
     * @return
     */
    public Player getxPlayer() {
        return xPlayer;
    }
```

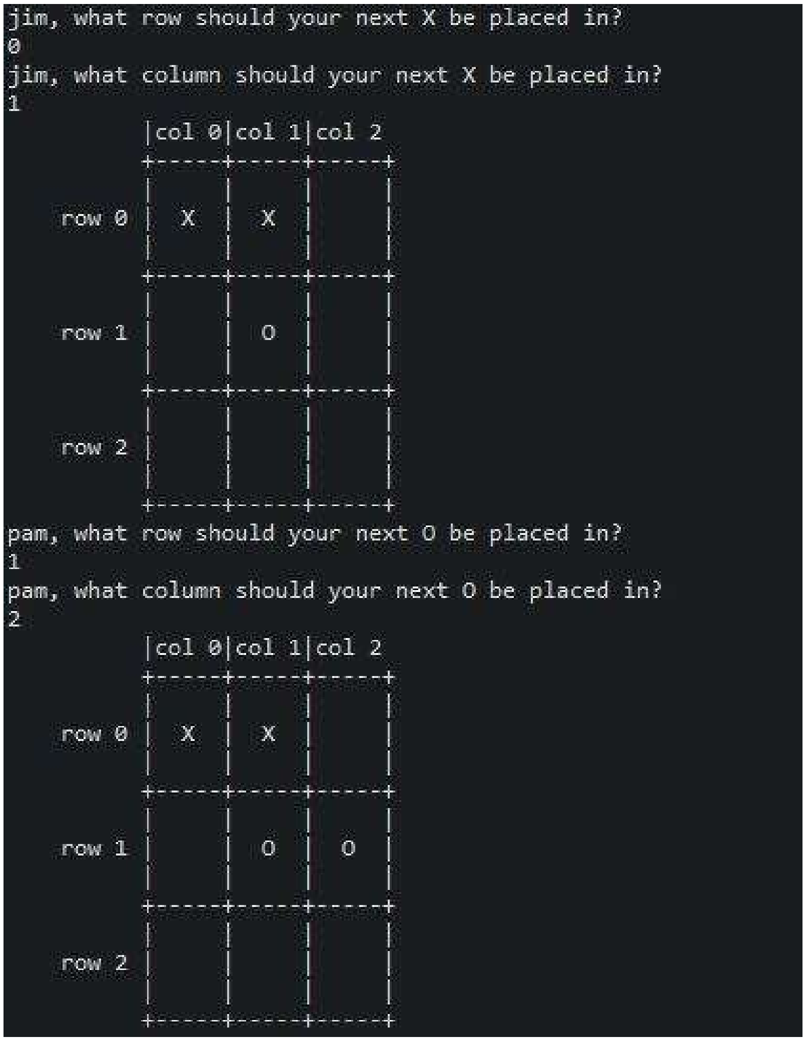


```
    /**
     * sets xPlayer
     * @param xPlayer
     */
    public void setxPlayer(Player xPlayer) {
        this.xPlayer = xPlayer;
    }
    /**
     * sets opponents and then calls play
     * @throws IOException
     */
    public void runTheGame() throws IOException
    {
        xPlayer.setOpponent(oPlayer);
        oPlayer.setOpponent(xPlayer);

        xPlayer.play();
    }


}
```

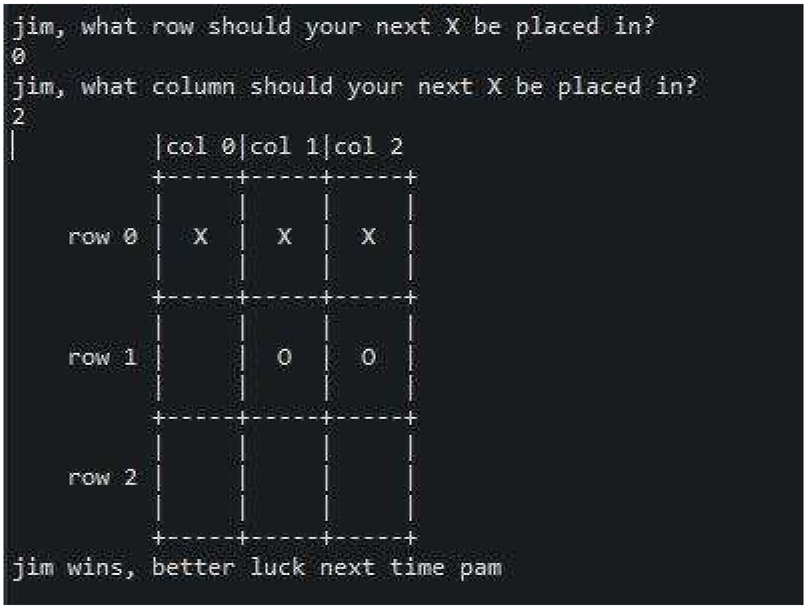
Output:



```
Please enter the name of the 'X' player: jim

Please enter the name of the 'O' player: pam
jim, what row should your next X be placed in?
0
jim, what column should your next X be placed in?
0
          |col 0|col 1|col 2
          +-----+-----+-----+
          |     |     |     |
    row 0 |  X  |     |     |
          |     |     |     |
          +-----+-----+-----+
          |     |     |     |
    row 1 |     |     |     |
          |     |     |     |
          +-----+-----+-----+
          |     |     |     |
    row 2 |     |     |     |
          |     |     |     |
          +-----+-----+-----+
pam, what row should your next O be placed in?
1
pam, what column should your next O be placed in?
1
          |col 0|col 1|col 2
          +-----+-----+-----+
          |     |     |     |
    row 0 |  X  |     |     |
          |     |     |     |
          +-----+-----+-----+
          |     |     |     |
    row 1 |     |  O  |     |
          |     |     |     |
          +-----+-----+-----+
          |     |     |     |
    row 2 |     |     |     |
          |     |     |     |
          +-----+-----+-----+
```



```
jim, what row should your next X be placed in?
0
jim, what column should your next X be placed in?
1
          |col 0|col 1|col 2
          +-----+-----+-----+
          |     |     |     |
    row 0 |  X  |  X  |     |
          |     |     |     |
          +-----+-----+-----+
          |     |     |     |
    row 1 |     |  O  |     |
          |     |     |     |
          +-----+-----+-----+
          |     |     |     |
    row 2 |     |     |     |
          |     |     |     |
          +-----+-----+-----+
pam, what row should your next O be placed in?
1
pam, what column should your next O be placed in?
2
          |col 0|col 1|col 2
          +-----+-----+-----+
          |     |     |     |
    row 0 |  X  |  X  |     |
          |     |     |     |
          +-----+-----+-----+
          |     |     |     |
    row 1 |     |  O  |  O  |
          |     |     |     |
          +-----+-----+-----+
          |     |     |     |
    row 2 |     |     |     |
          |     |     |     |
          +-----+-----+-----+
```



```
jim, what row should your next X be placed in?
0
jim, what column should your next X be placed in?
2
          |col 0|col 1|col 2
          +-----+-----+-----+
          |     |     |     |
    row 0 |  X  |  X  |  X  |
          |     |     |     |
          +-----+-----+-----+
          |     |     |     |
    row 1 |     |  O  |  O  |
          |     |     |     |
          +-----+-----+-----+
          |     |     |     |
    row 2 |     |     |     |
          |     |     |     |
          +-----+-----+-----+
jim wins, better luck next time pam
```