Java Progammierkurs
TicTacToe (Games)

Institut: Beuth Hochschule für Technik Berlin
Dozent: Prof. Dr. Christian Forler
Url: `https://lms.beuth-hochschule.de/`
Email: `cforler(at)beuth-hochschule.de`

Aufgabe 1 (4 Punkte) TicTacToe die Klasse
Heute wollen wir das berühmte Spiel TicTacToe implementieren. Entwerfen dazu eine Klasse `TicTacToe`, die eine Objektvariable `board` vom Typ `char[][]` und die Objektvariable `currentPlayer` vom Typ `char` beinhaltet. Desweitern sollet noch die folgenden vier Klassenvariablen Teil der Klasse sein.

```
public static final int  BOARD_SIZE = 3;
public static final char PLAYER_ONE = 'X';
public static final char PLAYER_TWO = 'O';
public static final char EMPTY_CELL = '␣';
```

Implementieren Sie unter anderem die folgenden Objekt-Methoden. Implementieren Sie ggf. noch private Helfermethoden die Ihnen das Leben erleichtern.

- Einen Konstruktor ohne Parameter der die Objektvariablen initalisiert.
- Getter-Methoden.
- `public void initializeBoard()`
  Diese Methode resettet das Spielfeld.
- `toString()`
  Diese Methode soll das Spielfeld als String-Representation von `board` zurückliefern.
- `public void printBoard()`
  Dies Methode gibt das Spielfeld als ASCII-Arts aus.
- `public boolean isBoardFull()`
  Gibt `true` zurück wenn das Board voll ist, ansonsten `false`.
- `boolean checkForWin()`
  Gibt `true` zurück wenn es einen Gewinner gibt, ansonsten `false`.
- `public void switchPlayer()`
  Wechselt den Spieler der gerade am Zug ist.
- `public boolean placeMark(int col, int row)`
  Gibt `true` zurück falls der derzeitige Spieler einen validen Zug macht und weschselt den Spieler, ansonsten gibt die Methode nur `false` zurück.

Aufgabe 2 (4 Punkte) TicTacToe das Spiel
Schreiben Sie eine Main-Klasse die es zwei Spielern ermöglicht gegeneinander Tic-Tac-Toe zu Spielen. Verwenden Sie die Klasse `TicTacToe` aus Aufgabe 1.

```
------------
|   |   |   |
-------------
|   |   |   |
-------------
|   |   |   |
-------------
Player X. Please enter your coordinates (x y): 2 1

-------------
|   |   |   |
-------------
|   |   | X |
-------------
|   |   |   |
-------------
Player O. Please enter your coordinates (x y): 0 0
.
.
.
-------------
| O |   | X |
-------------
|   | O | X |
-------------
|   |   |   |
-------------
Player X. Please enter your coordinates (x y): 2 2
-------------
| O |   | X |
-------------
|   | O | X |
-------------
|   |   | X |
-------------
Sorry! Player O. You lost. :(
Player X. Congratulations! You won. :)
```

Aufgabe 3 (4 Punkte) Der Tic-Tac-Toe Bot
Jetzt wollen wir einen Bot für unser TicTacToe Spiel implementieren. Diese Klasse soll über die folgenden drei Variablen verfügen.

```
private TicTacToe ttt;
private char botMark;
private char playerMark;
```

Implementieren Sie unter anderem die folgenden Objekt-Methoden. Implementieren Sie ggf. noch zusätzliche private Helfermethoden die Ihnen das Leben

erleichtern.

- `public Bot(TicTacToe ttt, char botMark)` Konstruktor der die Membervariablen initalisiert.
- `public char getBotMark()` Getter-Methode für die Objektvariable `botMark`.
- `private boolean winGame()` Der Bot soll erkennen, ob er in diesem Zug gewinnen kann. Falls ja, soll er den entsprechenden Feld makieren und `true` zurück geben, anonsten `false`. Vielleicht klappt es ja das nächste mal. :)
- `private boolean notLoseGame()` Der Bot soll erkennen, ob er in diesem Zug eine Niederlage abwenden kann. Falls ja, soll er den entsprechenden Feld makieren und `true` zurück geben, anonsten `false`.
- `public void nextTurn()` Der Bot makiert ein Feld. Überlegen Sie sich eine gute Gewinnstrategie. Verwenden Sie dafür die privaten Methoden `winGame` und `notLoseGame`.

Aufgabe 4 (4 Punkte) Das Tic-Tac-Toe Spiel
In dieser Klasse `SinglePlayer` verbirgt sich die singleplayer Variante des Spiels Tic-Tac-Toe. Dei Klasse soll sich auch um die Benutzerinteraktion kümmern. Verwenden Sie die folgende Variablen.

```
private static java.util.Scanner reader = new java.util.Scanner(
    System.in);
private TicTacToe ttt;
private Bot bot;
```

Implementieren Sie unter anderem die folgenden Objekt-Methoden. Implementieren Sie ggf. noch zusätzliche private Helfermethoden die Ihnen das Leben erleichtern.

- `public SinglePlayer()`
  Singleplayer Spiel bei dem der Bot als Spieler 2 spielt.
- `public SinglePlayer(char playerBot)`
  Singleplayer Spiel bei dem der Bot als `playerBot` spielt.
- `public void printReadAndPlayNextTurn`
  Der Spieler darf einen Zug machen.
- `public boolean isOver()`
  Gibt `false` zurück falls das Spiel vorbei ist, ansonsten `true`.
- `public void printWin()`
  Falls der Spieler verloren hat wird `"You Lost!"`, wenn er gewonnen hat `"Congratulations! You won."` und bei einem Unentschieden wird `"Draw."` auf der Kommandozeile ausgegeben.
- `public void play()`
  Es wird solange gespielt bis das Spiel zu Ende ist.
- `public void resetGame()`
  Resetet das Spielfeld.