

TikTacToe

Überblick

Im Rahmen einer Projektausarbeitung, welche mit Schulnoten bewertet wird um dem Teilnehmer/der Teilnehmerin ein besseres Verständnis über seine/ihre Leistungen zu schaffen, soll eine Art „TikTacToe – Spiel“ (konsolenbasiert) programmiert werden.

Beurteilungsschema

Beurteilung	Anforderungen
Sehr Gut	Grundlegende Teilaufgaben vollständig erfüllt und erweiterte Teilaufgaben vollständig erfüllt.
Gut	Grundlegende Teilaufgaben vollständig erfüllt und erweiterte Teilaufgaben überwiegend erfüllt.
Befriedigend	Grundlegende Teilaufgaben vollständig erfüllt oder grundlegende Teilaufgaben überwiegend erfüllt und zumindest Ansätze bei allen erweiterten Teilaufgaben.
Genügend	Grundlegende Teilaufgaben überwiegend erfüllt.
Nicht genügend	Grundlegende Teilaufgaben nicht überwiegend erfüllt.

Hilfsmittel

Erlaubt sind:

- Internet (Google, Foren, Teachin-Sites, ...)
- Alte, selbstprogrammierte Projekte

Prüfungsstoff

Es wurde eine Aufgabenstellung aus folgenden Kapiteln gewählt:

- Java OOP
- Java Kommentare
- Java If...Else
- Java Variablen und Datentypen
- Java Array
- Java Random
- Java Mehrdimensionale Arrays
- Java While-Schleife
- Java For-Schleife
- Java Switch
- Java Methoden
- Java Objekterstellung
- Java Konstruktor
- Java Mehrere Klassen mit Methoden und Objekten

Aufgabenstellungen

Aufgabe 1:

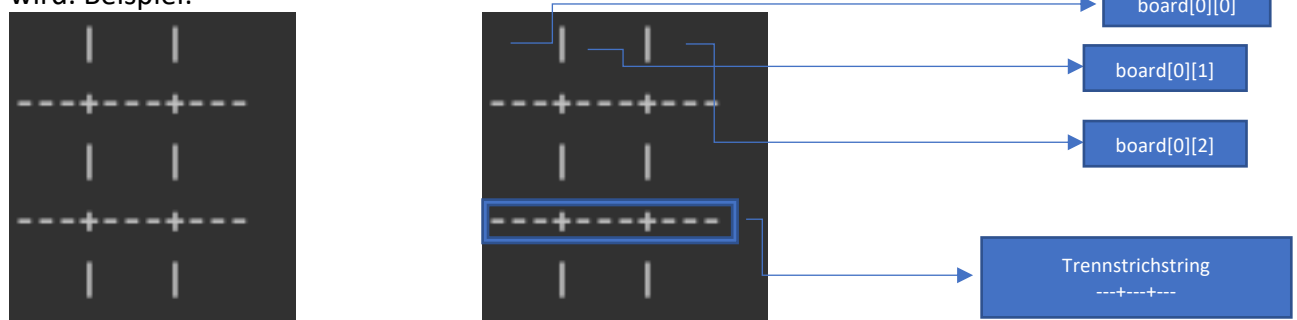
Erstelle eine Klasse TicTacToe. Das Spielfeld soll mittels einem 2-D-Arrays realisiert werden. Wähle dazu einen geeigneten Datentypen und lege das Array **board** an. Die Anzahl der Zeilen und Spalten des Arrays werden mittels Konstanten definiert (3x3 ist eine gültige Größe). Im Konstruktor soll nun das Spielfeld (Array) mithilfe der Konstanten instanziiert werden.

Aufgabe 2:

Erstelle eine Methode **initBoard** um das Spielfeld zu initialisieren. Diese Methode soll aufgerufen werden um das Spielfeld auf vorgegebene Standardwerte zu setzen.

Aufgabe 3:

Erstelle eine Methode **printBoard** mit der das Spielfeld gänzlich auf der Konsole ausgegeben wird. Beispiel:



Aufgabe 4:

Erstelle eine neue Klasse **Main**. Erstelle in ihr eine Main-Methode und erzeuge darin ein **Objekt** der Klasse TicTacToe mit dem Namen **game**. Initialisiere nun dein Spielfeld und gib es in der Main Methode aus.

Aufgabe 5:

Erstelle in deiner Main-Klasse zwei neue Methoden (`getRow` und `getColumn`), welche mittels Scanner vom Benutzer die Zeile und die Spalte einlesen in welcher der Spieler seine Figur platzieren möchte.

Die gültigen Eingaben hängen von der Größe deines Spielfelds ab (Bei 3x3 gibt es 0,1,2 für die Zeile und 0,1,2 für die Spalte.) Überprüfe die Benutzereingaben auf **Gültigkeit** (**Benutzereingabe ist eine ganze Zahl und in den gegebenen Grenzen**) und achte darauf, dass keine **IndexOutOfBoundsException** auftritt. Fange diese Exception mittels **try/catch**.

Am Ende soll die jeweilige Methode den **Index** der Zeile oder Spalte **zurückgeben**.

Aufgabe 6:

In der TicTacToe-Klasse soll nun eine Methode zum tatsächlichen Platzieren eines Symbols erstellt werden. Nenne diese Methode `setPlay`. Ihr übergibt man einen Zeilenindex *i*, einen Spaltenindex *j* und das Symbol als String *player*.

Die Methode setzt auf der übergebenen Position den vorhandenen String „ „ auf „ X „ oder „ O „. Die Methode gibt `true` zurück, falls die Position gesetzt wurde und `false`, wenn es nicht funktioniert hat.

ACHTUNG: Man darf nur in Felder setzen wo noch keine Figur gesetzt wurde!

Aufgabe 7:

Erstelle in deiner Main-Klasse eine Methode `getPlayer`. Abhängig davon welcher Spieler gerade an der Reihe war soll dementsprechend der Andere zum Zug kommen. Die Methode gibt dann den gewählten Spieler String *player* zurück.

Aufgabe 8:

In der TicTacToe Klasse soll nun eine Methode (`isGameOver`) zum Überprüfen des Spielstandes erzeugt werden, um herauszufinden ob das Spiel bereits Vorbei ist oder nicht. Dazu müssen alle Zeilen, Spalten und die beiden Diagonalen auf gleiche Symbole überprüft werden.

Eine Win-Condition wäre:

```
!board[0][0].equals(" ") && board[0][0].equals(board[0][1]) && board[0][1].equals(board[0][2])
```

Übersetzt bedeutet das:

Board[0][0] ist nicht der Standardstring und stimmt mit dem 2 überein und der zweite mit dem 3.

Aufgabe 9:

Kommentiere deinen Code sinngemäß.

Aufgabe 10:

Halte dich an die Regeln für Coding Style.