**Dokumentation**

1. **Einleitung (Kontext, Motivation)**
2. **Funktionalitäten**
3. **Code-Dokumentation**
4. **Fazit / Quellen/ Anhang**

**1. Einleitung**

Dieses Programm implementiert das klassische Spiel Tic Tac Toe für zwei Spieler auf der Kommandozeile…

**2. Funktionalitäten**

Die beiden Spieler können

**3. Code-Dokumentation**

**Globale Variablen**

* fields: Ein zweidimensionaler Array (3x3) zur Darstellung der Inhalte des Spielfelds
* spieler: Variable zur Verfolgung der aktuellen Spielers (1 oder 2)
* win: Boolean zur Überprüfung, ob jemand gewonnen hat
* draw: Boolean zur Überprüfung auf ein Unentschieden
* zahl: Hilfsvariable zur Initialisierung der Spielfeldwerte

**Initialisierung des Spielfelds**

Das Spielfeld wird als zweidimensionaler Array mithilfe einer for-Schleife initialisiert und mit Zahlen von 1 bis 9 befüllt. Diese Zahlen repräsentieren die Positionen auf dem Spielfeld und sollen den Spielern bei der Orientierung helfen.

**Spielablauf**

Durch eine While-Schleife läuft das Spiel so lange, bis einer der beiden Spieler gewonnen hat oder das Spiel unentschieden ist.

Zunächst wird über die Methode „SpielfeldAusgeben“ der aktuelle Stand des Spielfelds ausgegeben. Anschließend überprüft das Programm, welcher Spieler gerade an der Reihe ist und gibt eine entsprechende Meldung aus. In dieser Meldung sehen die Spieler auch, welches Zeichen (X oder O) der momentane Spieler hat.

**Nutzereingabe und Fehlerbehandlung**

Nun wird der Spieler gefragt, wo er setzen möchte. Durch den Scanner wartet das Programm nun auf eine Eingabe des Spielers und speichert diese in der Variable „eingabe“.  
Handelt es sich bei diesem Wert nicht um einen numerischen Wert, wird über die try catch Funktion eine Fehlermeldung ausgegeben. Diese weist den Nutzer darauf hin, dass eine Zahl zwischen 1 und 9 eingegeben werden muss. Die falsche Eingabe wird verworfen und die Schleife wird erneut durchlaufen, sodass derselbe Spieler nochmals ein Feld wählen kann.

In einer If-Schleife wird anschließend geprüft, ob die eingegebene Zahl zwischen 1 und 9 ist. Ist dem nicht so, wird eine entsprechende Fehlermeldung ausgegeben und die Schleife wird durch den Befehl continue erneut durchlaufen, sodass derselbe Spieler nochmals an der Reihe ist.

**Berechnung Spielfeldposition**

Ist die Eingabe korrekt, wird die Position im Array berechnet, der die Eingabe entspricht. Hierbei werden die Variablen i für den ersten Index und j für den zweiten Index verwendet:

* Für die Berechnung von i wird zunächst von der Nutzereingabe 1 subtrahiert, um eine Bereichsverlagerung zu den Stellen des Arrays zu bewerkstelligen. Anschließend wird der Wert durch 3 geteilt. Da Java nicht rundet, sondern die Stellen nach dem Komma abschneidet, kommt hierbei der richtige Array-Index zwischen 0 und 2 raus.
* Für die Berechnung von j wird ebenfalls 1 von der Nutzereingabe subtrahiert. Anschließend wird durch die Modulo-Operation „%3“ der Spaltenindex im Array berechnet. Da die Modulo-Operation den Rest einer Teilung ausgibt, kommt auch hier immer der richtige Spalten-Index zwischen 0 und 2 raus.

**Zeichenplatzierung**

Anschließend prüft eine if-Schleife, ob in dem Feld bereits ein „X“ oder ein „O“ steht. Stimmt die, wird auch hier eine Fehlermeldung ausgegeben und der Spieler darf erneut ein Feld wählen.  
Ist das Feld noch frei, wird je nachdem, welcher Spieler gerade dran ist, ein „X“ oder ein „O“ an der eben berechneten Position in den Array geschrieben.

**Überprüfung auf Gewinn oder Unentschieden**

Im Anschluss prüft das Programm, ob einer der Spieler durch den vorherigen Spielzug gewonnen hat oder ob das Spiel unentschieden ist:

Zunächst wird überprüft, ob eine Reihe under Spalte dieselben Einträge hat. Hierfür werden durch eine for-Schleife die verschiedenen Kombinationsmöglichkeiten von Reihen und Spalten miteinander verglichen. Falls die 3 Werte einer Reihe oder Spalte gleich sind, wird win auf true gesetzt und die Schleife wird abgebrochen.

Anschließend wird getestet, ob eine der beiden Diagonalen die gleichen Werte enthält. Falls dem so ist, wird win auf true gesetzt.

Nun wird noch überprüft, ob das Spiel unentschieden ist. Hier bedeutet unentschieden, dass alle Felder belegt sind, aber keiner gewonnen hat. Für die Überprüfung wird draw zunächst auf true gesetzt, denn die Schleife prüft mit der Annahme, dass das Spiel unentschieden ist, bist ein leeres Feld gefunden wird. Mit zwei ineinander verschachtelten for-Schleifen wird jedes Feld des Arrays darauf überprüft, ob es leer ist. Wird ein leeres Feld gefunden, wird draw wieder auf false gesetzt und die Schleife wird durch einen break verlassen.

**Spielende und Ausgabe von Ergebnissen**

Falls win true ist, wird das Spielfeld ein letztes Mal ausgegeben. Außerdem erscheint in der Kommandozeile, welcher Spieler gewonnen hat. Ist das Spiel unentschieden, wird auch hier zunächst das finale Spielfeld ausgegeben. Anschließend erscheint eine Meldung, dass das Spiel unentschieden ist.  
Da die while-Schleife nur solange durchgeführt wird, bis einer der beiden Spieler gewonnen hat oder das Spiel unentschieden ist, wird die Schleife nicht erneut durchlaufen und das Spiel ist nach diesem Durchlauf beendet.

**Spielerwechsel**

Als letzter Schritt eines Spielzuges wird nun noch der Spieler gewechselt. War Spieler 1 an der Reihe, ist nun Spieler 2 dran und umgekehrt.

**Methode SpielfeldAusgeben**

Die Methode SpielfeldAusgeben ist verantwortlich für das Ausgeben des aktuellen Spielfelds auf der Konsole. Das übergebene zweidimensionale Array „arr“ enthält die Werte der einzelnen Felder im Spielfeld. Durch senkrechte und waagerechte Striche wird das Spielfeld dargestellt und mit den aktuellen Werten im übergebenen Array ausgegeben.