Tic-Tac-Toe

1. Der Roh-Entwurf des Spieles

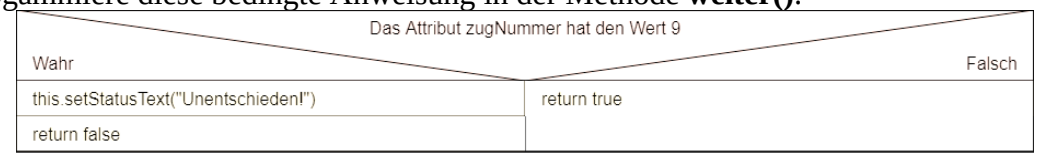
Öffne das Projekt TicTacToe und spiele das Spiel einmal durch.
Notiere zwei Fehler, die auftreten.

_____

_____

2. Das Spiel soll nach dem 9. Stein enden.

a) Deklariere ein privates Attribut **zugNummer** vom Typ **int**.
b) Initialisiere dieses Attribut in der 2. Zeile des Konstruktors mit dem Wert 0.
c) Erhöhe den Wert des Attributs in der Methode **neuerStein()** um eins.
d) Progammiere diese bedingte Anweisung in der Methode **weiter()**:



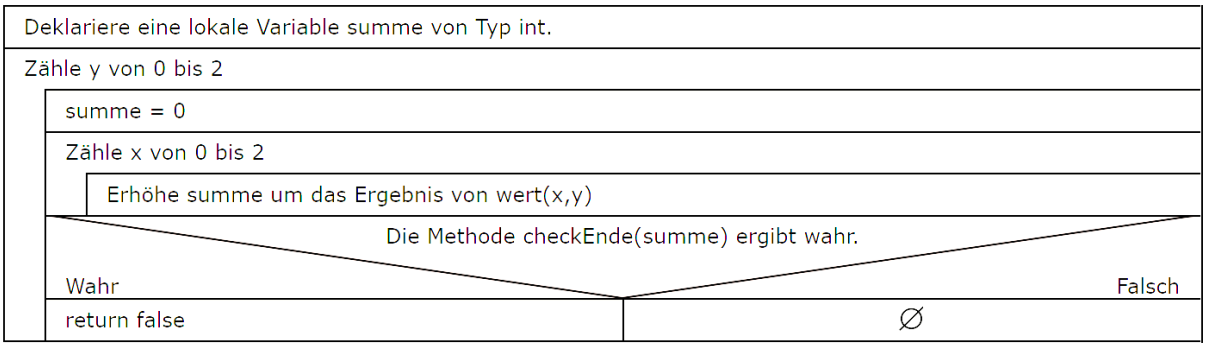
e) Teste dein Programm

3. Das Spiel endet, sobald ein Spieler eine waagrechte Reihe hat.

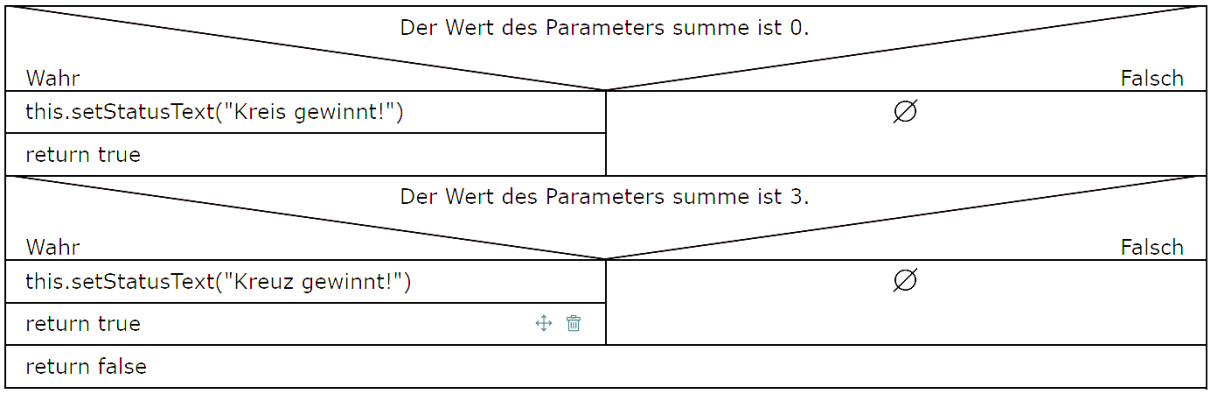
a) Die Methode wert(x,y) gibt den Wert des Feldes an der Position (x,y) zurück.
Trage die Rückgabe-Werte für folgende Fälle ein:

Inhalt des Feldes	leer	Kreis	Kreuz
Rückgabewert			

b) Programmiere im **else**-Zweig der Methode **weiter()** folgenden Code:



c) Die Methode **checkEnde(summe)** gibt derzeit immer false zurück.
Ersetze den Rumpf der Methode durch folgenden Code:



4. Das Programm endet, sobald ein Spieler eine senkrechte Reihe hat.

Kopiere die beiden FOR-Schleifen aus 3b) und ändere die Variablen so ab, dass das Programm jetzt prüft, ob der Spieler senkrechte Reihen vollständig belegt hat. Insgesamt sind nur 6 kleine Änderungen im kopierten Code nötig.

5. Eine weitere Sieg-Kombination

Ergänze den **else**-Zweig der Methode **weiter()** um folgende Code-Zeilen:

```
if (checkEnde(wert(0,0)+wert(1,1)+wert(2,2))){
    return false;
}
```

Gib an, welche Kombination für einen Sieg hiermit erkannt wird.

_____

_____

6. Die letzte Sieg-Kombination

Kopiere die 3 Code-Zeilen aus 5. und ändere sie so ab, dass die letzte Sieg-Kombination erkannt wird.

_____

_____

_____

7. Hinweis zum Return-Statement in Methoden

Ein Return-Statement unterbricht die Ausführung einer Methode. Alle Anweisungen, die darunter stehen, werden nicht mehr ausgeführt.

Aus diesem Grund sind diese beiden Methoden gleichwertig:

```
public boolean is99(int summe){
    if (summe==99){
        return true;
    } else {
        return false;
    }
}
```

```
public boolean is99(int summe){
    if (summe==99){
        return true;
    }
    return false;
}
```

Das zweite Return-Statement wird immer nur erreicht, wenn summe ungleich 99 ist.

Die Methode kann auch noch eleganter implementiert werden:

```
public boolean is99(int summe){
    return summe==99;
}
```