

# Lösung TicTacToe

Prototyp	Fachinhalte	Fehler
1. Spieler	Bedingte Wiederholung  lokale Variablen  Methodenaufrufe eines anderen Objekts	Fehlende Klammern für Aufruf von <code>spielerZug</code> .  Initialisierung des Spielfeldes fehlt  <code>== true</code> statt <code>== false</code> für Prüfung ob Feld frei ist
2. Computer	Zufall mit <code>Math.Random</code>	Klammer am Ende der Methode <code>computerZug()</code> fehlt.  <code>setzeZeichen1()</code> statt 2  <code>ueberpruefeFeldFrei()</code> ist Methode der Klasse <code>Spielbrett</code> .  Zuweisung <code>feld</code> für <code>Feld</code> fehlt.
3. Auswertung	Mehrfache Bedingungen mit <code>  </code> und <code>&amp;&amp;</code>  Methoden mit Rückgabewert	<code>gibtSpielendeAus()</code> wird nicht aufgerufen.  Datentyp bzw. Deklaration für <code>sieger</code> fehlt.

## Lösung Prototyp 1: Zeichen setzen

Fehlertyp	Fehler	Korrektur
Kompilierzeit	Fehlende Klammern für Aufruf von <code>spielerZug</code> . <pre>while (spielfeld.     spielerZug;</pre>	Klammern hinzufügen. <pre>while (spielfeld.     spielerZug());</pre>
Laufzeit	Initialisierung des Spielfeldes fehlt. <pre>class TicTacToe{     private Spielfeld spielfeld;</pre>	Attribut initialisieren indem neues Objekt erzeugt wird. <pre>TicTacToe{     private Spielfeld spielfeld = new Spielfeld();</pre>
logisch	<code>== true</code> statt <code>== false</code> für Prüfung ob Feld frei ist. <pre>feldFrei(feld) == true){     Feld ist bereits belegt! Ver</pre>	<code>== false</code> verwenden. <pre>.ueberpruefeFeldFrei(feld) == false). println("Das Feld ist bereits belegt</pre>

## Lösung Prototyp 2: Computer

Fehlertyp	Fehler	Korrektur
Kompilierzeit	Klammer am Ende der Methode computerZug() fehlt. <pre> } spielfeld.setzeZeichen2(feld); </pre>	Klammer hinzufügen. <pre> spielfeld.setzeZeichen2(feld); } </pre>
Kompilierzeit	ueberpruefeFeldFrei() ist Methode der Klasse Spielbrett. <pre> while(ueberpruefeFeldFrei(feld) == false){ </pre>	spielbrett.ueberpruefeFeldFrei(). <pre> (spielbrett.ueberpruefeFeldFrei(feld)) </pre>
logisch	setzeZeichen1() statt 2 <pre> spielfeld.setzeZeichen1(feld); </pre>	2 statt 1. <pre> spielfeld.setzeZeichen2(feld); </pre>
logisch	Zuweisung feld für Feld fehlt. <pre> le(spielfeld.ueberpruefeFeldFrei(feld)); System.out.println("Das Feld is"); spielfeld.liesEingabe(); </pre>	feld = hinzufügen. <pre> feld = spielfeld.liesEingabe(); le(spielfeld.ueberpruefeFeldFrei(feld)); System.out.println("Das Feld is"); </pre>

## Lösung Prototyp 3: Auswertung

Fehlertyp	Fehler	Korrektur
Kompilzeit	Datentyp bzw. Deklaration für sieger fehlt. <pre>lic void gibSpielend //Die Methode ermit sieger = spielfeld.</pre>	Datentyp String hinzufügen und Variable so korrekt deklarieren. <pre>ic void gibSpielendeAus(){ //Die Methode ermittleSieg String sieger = spielfeld.</pre>
logisch	gibtSpielendeAus() wird nicht aufgerufen. <pre>spielereu.zug(), } }</pre>	Methodenaufruf nach Spielschleife hinzufügen. <pre>// Spielschleife while (spielfeld.ue     spielfeld.zug() ) gibSpielendeAus();</pre>