

Aufgabe 6 – (OOP Teil 3 von TIC TAC TOE)

Es sollte das bekannte Spiel **TIC TAC TOE** als Python-Programm erstellt werden. Anders als im HWS 2020 sollte diesmal das Spiel **Objekt-orientiert programmiert (=OOP)** werden. Zunächst haben wir für die **Klasse Spielfeld** die **spielfeld_anzeigen()** – Methode (=Funktion) und nutzen eine Liste, um die Positionen des TIC TAC TOE Spielfelds als Speicherplätze festzuhalten. Im nächsten Schritt haben wir die **Klasse Spieler** eine **spieler_eingabe()** - Methode programmiert, um die Eingabe einer Ziffer von 1 bis 9 vom **Spieler 1** „X“ und vom **Spieler 2** „O“ in die Liste **lsp** zu speichern.



Wir werden in der letzten Aufgabe die zwei weiteren Methoden **spiel_gewonnen()** und **spiel_unentschieden()** entwickeln:

- a.) Spieler hat gewonnen – prüfen wir nicht nach jedem Spielzug, sondern ab dem dritten Spielzug, dabei soll die **spiel_gewonnen()**- Methode aufgerufen werden.

Der Spieler hat gewonnen, wenn alle 3 Felder gleich sind – horizontal, vertikal oder diagonal

Horizontale-Möglichkeiten	<pre> 1 2 3 4 5 6 7 8 9 </pre>	<pre> 1 2 3 4 5 6 7 8 9 </pre>	<pre> 1 2 3 4 5 6 7 8 9 </pre>
Vertikale-Möglichkeiten	<pre> 1 2 3 4 5 6 7 8 9 </pre>	<pre> 1 2 3 4 5 6 7 8 9 </pre>	<pre> 1 2 3 4 5 6 7 8 9 </pre>
Diagonale-Möglichkeiten	<pre> 1 2 3 4 5 6 7 8 9 </pre>	<pre> 1 2 3 4 5 6 7 8 9 </pre>	

Die Funktion **spiel_gewonnen()** soll aus dem Hauptprogramm aufgerufen werden. Überlegen Sie sich einen **Rückgabewert** für die Funktion. Der Gewinner soll angezeigt werden, bevor das TIC TAC TOE-Spiel beendet wird.

- b.) Das Spiel geht unentschieden aus, sobald der **Spieler 1** „X“ seinen 9-ten Spielzug gemacht hat – er entscheidet zu einem letzten Mal, ob er das Spiel gewonnen hat oder es unentschieden ausgeht.

Falls der **Rückgabewert** der Funktion **spiel_gewonnen()** = **Spieler 1** „X“ seinen 9-ten Spielzug gemacht und hat nicht gewonnen → somit geht das Spiel unentschieden aus.