

1. Vorlesung

Organisatorisches, Wiederholung & Aufzählungstypen

Organisatorisches

- für einige Themen gibt es ausführliche Skripte (<https://vrpntngr.github.io/programmieren2>)
- für die weiteren Themen wird es Folien und Links geben
- die Veranstaltung findet in Präsenz statt
- Skripte werden nach der Veranstaltung hochgeladen
- Es wird voraussichtlich 12 Aufgaben geben. Bitte geben Sie davon 8 in Moodle ab. (Ich rate ihnen alle zu machen.) Die Abgabe ist jeweils freitags in der darauffolgenden Woche.
- In der Übung bearbeiten wir Aufgaben zu den Themen aus der Vorwoche
- Am Ende schreiben wir eine Klausur (am Laborrechner)

Empfehlung: lernen Sie gemeinsam

- Aufgaben im Pair lösen (aktiv zusammen)
- besprechen Sie Ihre Lösungen miteinander
- reviewn Sie Aufgaben untereinander (stellen Sie sich dabei Fragen: Warum haben Sie was wie gelöst? Was würde man selbst anders machen)

Klausur

- am 23.7.
- 90 Minuten Bearbeitungszeitraum
- keine große Aufgabe, sondern mehrere kleine Aufgabe
- an Laborrechnern
- keine Kommunikation (mit Menschen oder KI)

enum / Aufzählungstypen

Motivation

- eine Variable beschreibt eine festgelegte Menge an Zuständen.

Beispiel Wochentage:

```
String tag = "MONTAG"    // kann auch Werte "Dienstag" usw. annehmen
int tag = 0;             // Magic Number für "Montag"
```

Zu welchen Problemen kann dies führen?

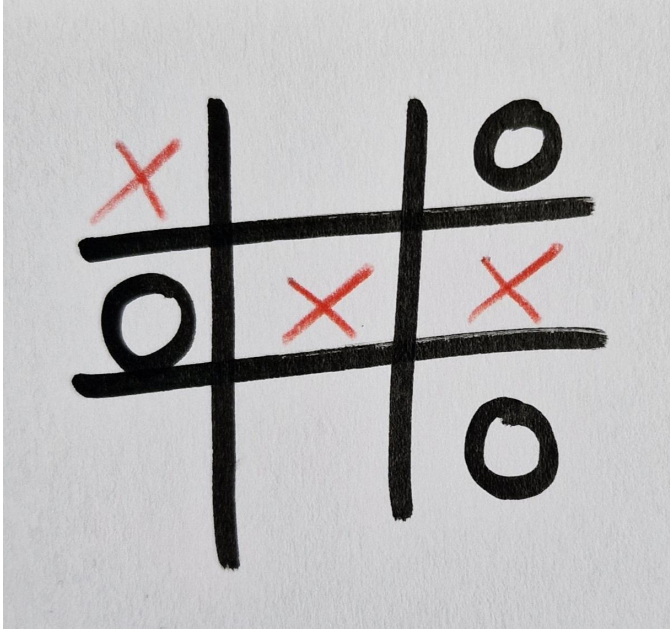
Probleme

Variablen können beliebigen Werten aus dem Wertebereich annehmen:

```
String tag = "hallo" oder int tag = 123
```

Magic Numbers sollten vermieden werden → schlechte Lesbarkeit

Beispiel TicTac Toe



Probleme

- Zustände `EMPTY (0)`, `RED (1)`, `BLACK (2)` verschlüsselt → magic numbers
- `field[i][j]` könnte auch beliebige andere `int`-Werte annehmen
- Code schlecht lesbar

Erste Verbesserung: Konstanten

- etwas besser, aber immer noch beliebige Werte für `field[i][j]` möglich

beliebige Werte vertragen sich nicht mit Typsicherheit

Anforderungen

- eigener Datentyp
- endliche Anzahl an Zuständen bzw. Werten
- leserliche Bezeichnung der Werte

Enumerations (Aufzählungstyp)

Schlüsselwort `enum`

Syntax:

```
enum TypName {WERT1, WERT2, WERT3};
```

Enum

- typsicher
- andere Zustände nicht möglich

Details:¶

- alle `enum` erben implizit von `java.lang.Enum`
- `enum` sind Referenztypen
- die Konstanten (Werte) in `enum` sind automatisch `static` und `final`
- `==` kann verwendet werden (auch `switch()`); `equals()` gibt es aber auch

- auch Definition von Methoden möglich
- `toString()`, `equals()` usw. aus `Object` können überschrieben werden
- Konstanten können mit Attributen versehen werden (dann noch privater Konstruktor notwendig)
- Zugriff auf das Array von Konstanten mithilfe von `values()`