Subtyping

Wie wir gelernt haben, definiert jede Klasse in Java auch einen Variablentyp - in diesem Sinne definiert jede Subklasse einen **Subtyp**.

Um Beispielsweise einen Fuhrpark zu modellieren, kann die folgende Klassenhierarchie zum Einsatz kommen:

Wir definieren zunächst eine Basisklasse `Fahrzeug`, die als Oberklasse für die anderen Klassen dienen soll

- Ein `Zweirad` ist ein Fahrzeug mit 2 Rädern.
- Ein `Tretroller` ist ein Zweirad mit Höchstgeschwindigkeit 12km/h
- Ein `Fahrrad` ist ein Zweirad, mit Maximalgeschwindigkeit 30 km/h.
- Ein `Auto` ist ein Fahrzeug mit 4 Rädern und Maximalgeschwindigkeit 140 km/h.
- Ein `Rennwagen` ist ein Auto mit Maximalgeschwindigkeit 220 km/h.
- Ein `Lastwagen` ist ein Auto mit dem zusätzlichen Attribut Ladung.

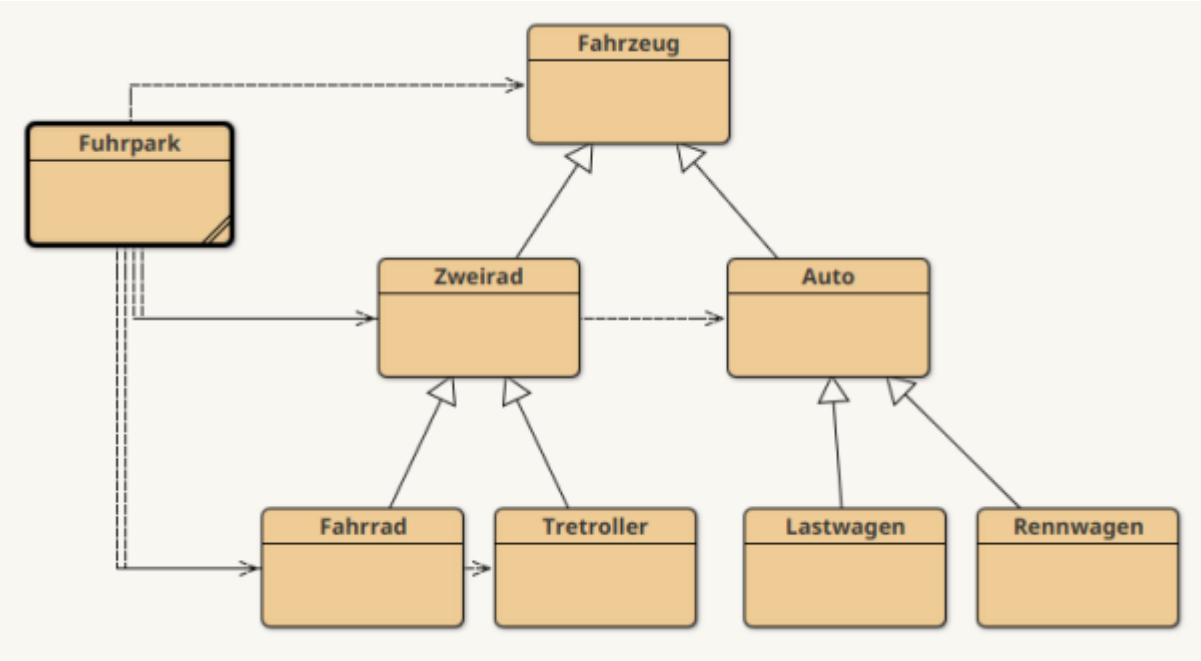
Polymorphie

Unter dem Begriff **Polymorphie** versteht man bei Variablen, dass eine Variable vom Typ einer Oberklasse auch Werte von Typ ihrer Subklassen halten kann. Man spricht von Variablenpolymorphie.

Polymorphie bedeutet "Vielgestaltigkeit" und tritt in der OOP noch in anderen Zusammenhängen auf, wie wir sehen werden.

Beispiel:

Für die Klassenhierarchie wie sie im Bild dargestellt ist, funktioniert der folgende Code:



```
public Fuhrpark()
    {
        Fahrzeug f1 = new Fahrzeug("FID001");
        Zweirad t1 = new Tretroller("Roller1");
        Zweirad t2 = new Fahrrad("Radl001");

        f1 = new Auto("EinAuto");

        System.out.println("Fuhrpark wurde erzeugt!");
    }
```

Man erkennt, man kann in einer Variablen vom Typ `Fahrzeug` eben auch beliebige Objekte vom Typ einer Unterklasse speichern.

(A1)

Welche Typen können Werte haben, die in den folgenden Variablen gespeichert werden?

```
Fahrzeug f;
Tretroller r;
Zweirad v;
Auto a;
```

fuhrpark.png	24.6 KiB	19.03.2024 08:51
kap10_04subtyping.odp	1.4 MiB	22.11.2021 20:25
kap10_04subtyping.pdf	224.5 KiB	22.11.2021 20:25

From:
https://info-bw.de/ -

Permanent link:
https://info-bw.de/faecher:informatik:oberstufe:modellierung:vererbung:subtyping:start?rev=1710838777

Last update: **19.03.2024 08:59**