Lernziele

Technische Berufsschule Zürich

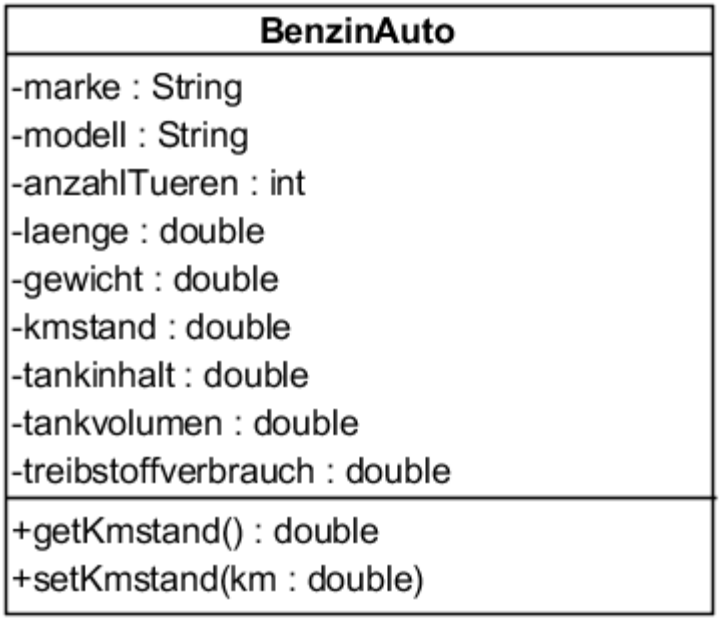
Die Lernenden…

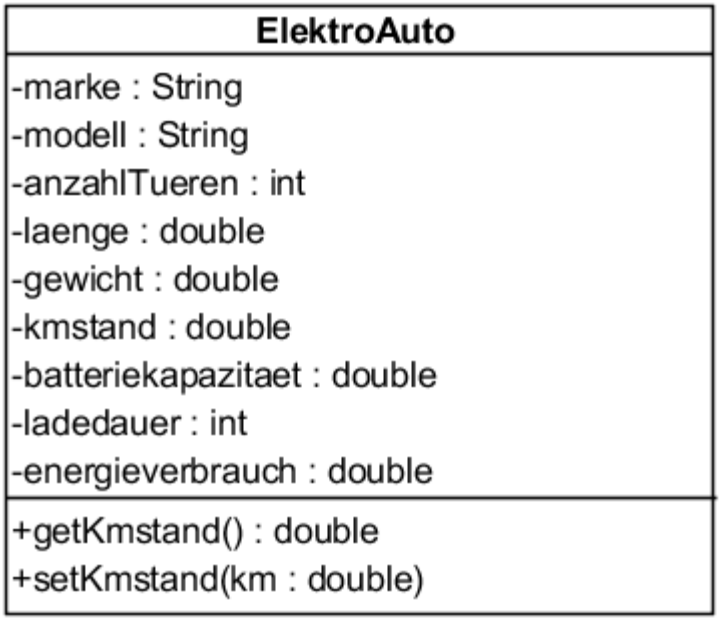
- … wissen was Vererbung ist und verstehen das Konzept dahinter
- … kennen Java-spezifische Schlüsselwörter, um Vererbung zu realisieren
- … können ein einfaches Vererbungsbeispiel lösen

Vererbung

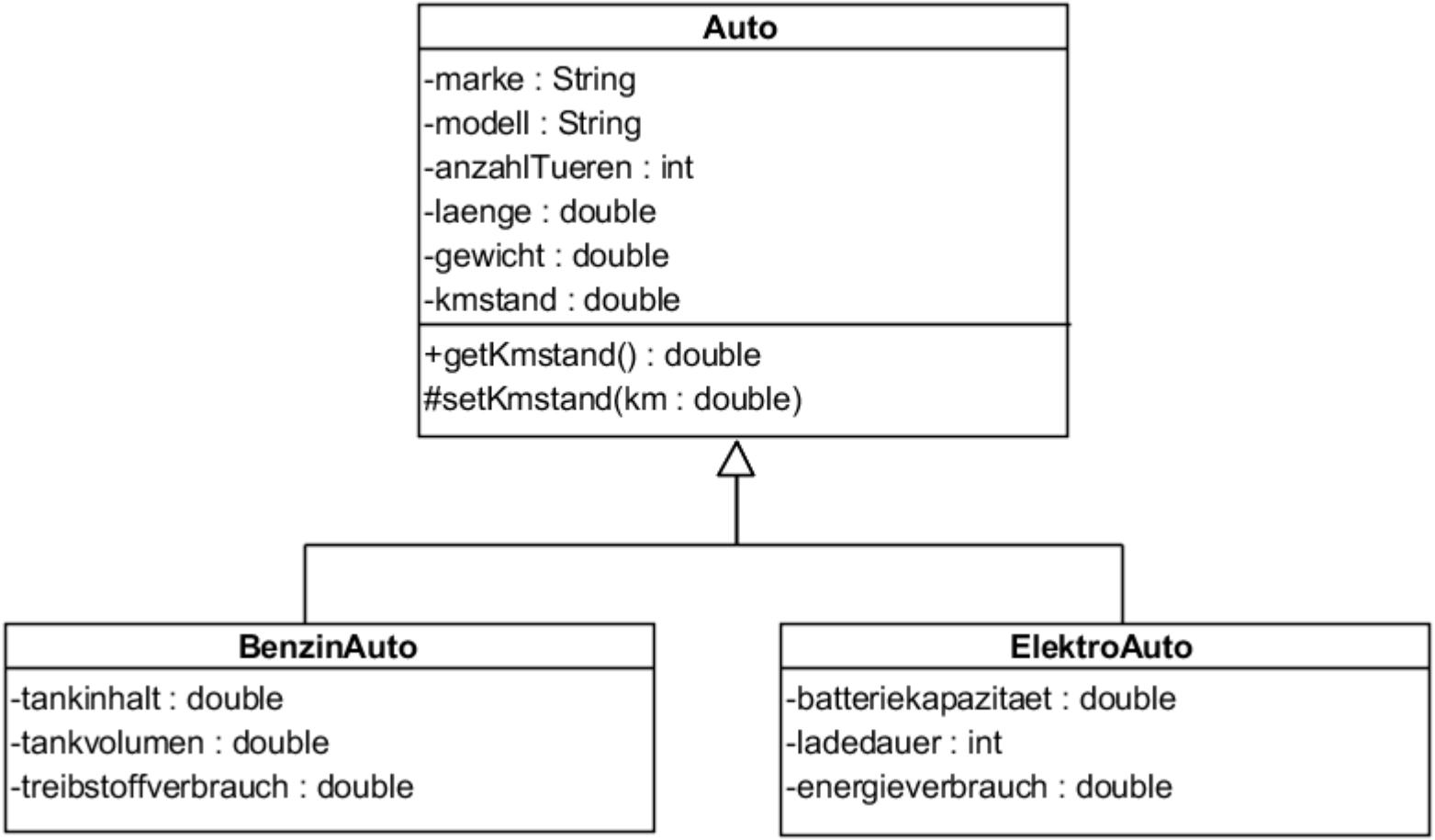
- Gemeinsamkeiten?
- Unterschiede?

Vererbung


BenzinAuto
-marke : String
-modell : String
-anzahlTueren : int
-laenge : double
-gewicht : double
-kmstand : double
-tankinhalt : double
-tankvolumen : double
-treibstoffverbrauch : double
+getKmstand() : double
+setKmstand(km : double)


ElektroAuto
-marke : String
-modell : String
-anzahlTueren : int
-laenge : double
-gewicht : double
-kmstand : double
-batteriekapazitaet : double
-ladedauer : int
-energieverbrauch : double
+getKmstand() : double
+setKmstand(km : double)

Vererbung

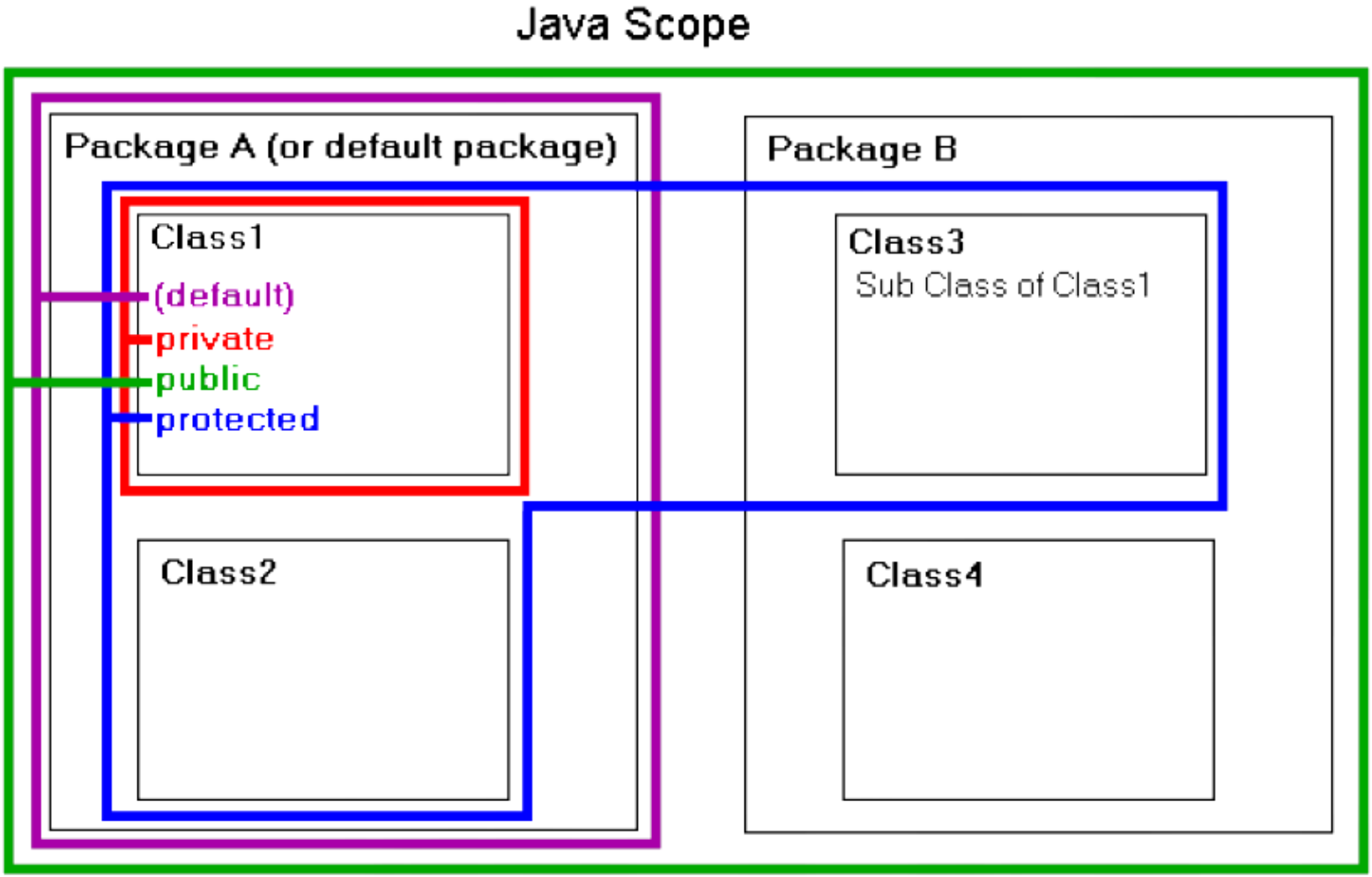

Auto
-marke : String
-modell : String
-anzahlTueren : int
-laenge : double
-gewicht : double
-kmstand : double
+getKmstand() : double
#setKmstand(km : double)
BenzinAuto
-tankinhalt : double
-tankvolumen : double
-treibstoffverbrauch : double
ElektroAuto
-batteriekapazitaet : double
-ladedauer : int
-energieverbrauch : double

Vererbung - Sichtbarkeiten

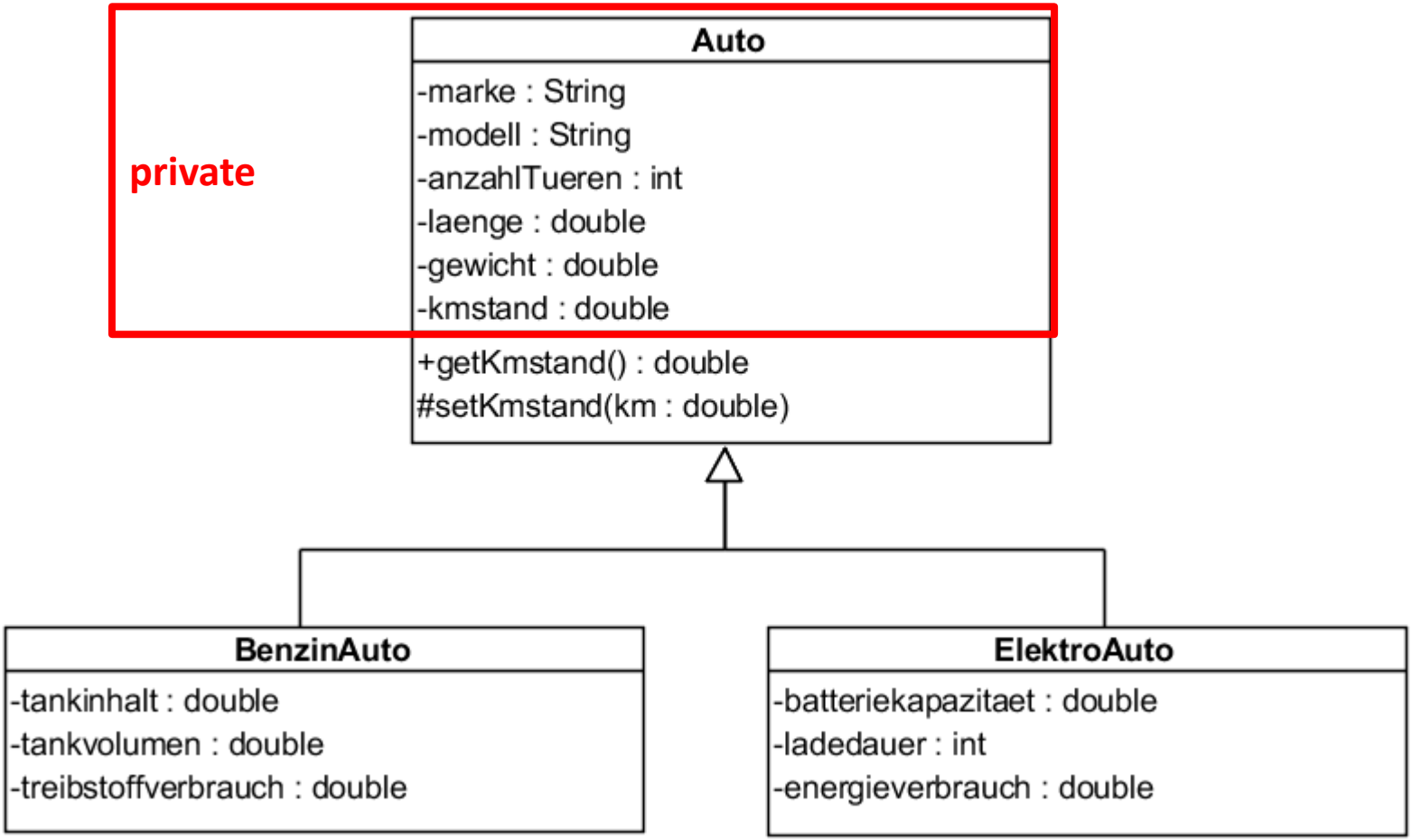
TBZ
Technische Berufsschule Zürich

Modifizierer	eigene Klasse	Klasse in gleichem Package	Unterklasse in anderem Package	Nicht Unterklasse in anderem Package
public	Ja	Ja	Ja	Ja
protected	**Ja**	**Ja**	**Ja**	**Nein**
- Standard bzw. package	Ja	Ja	Nein	Nein
private	Ja	Nein	Nein	Nein

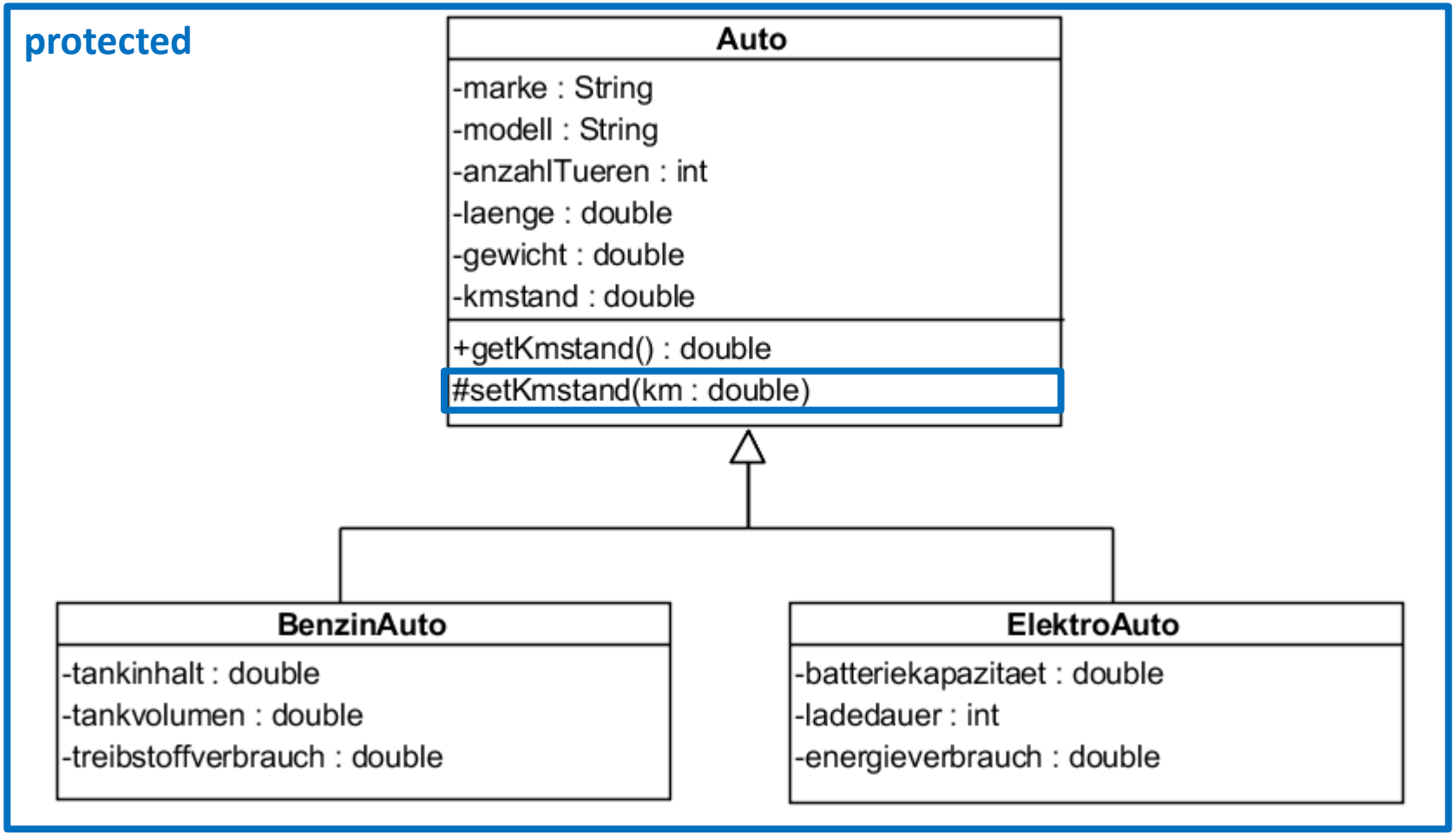
Vererbung - Sichtbarkeiten



Vererbung


private
Auto
-marke : String
-modell : String
-anzahlTueren : int
-laenge : double
-gewicht : double
-kmstand : double
+getKmstand() : double
#setKmstand(km : double)
BenzinAuto
-tankinhalt : double
-tankvolumen : double
-treibstoffverbrauch : double
ElektroAuto
-batteriekapazitaet : double
-ladedauer : int
-energieverbrauch : double

Vererbung


protected
Auto
-marke : String
-modell : String
-anzahlTueren : int
-laenge : double
-gewicht : double
-kmstand : double
+getKmstand() : double
#setKmstand(km : double)
BenzinAuto
-tankinhalt : double
-tankvolumen : double
-treibstoffverbrauch : double
ElektroAuto
-batteriekapazitaet : double
-ladedauer : int
-energieverbrauch : double

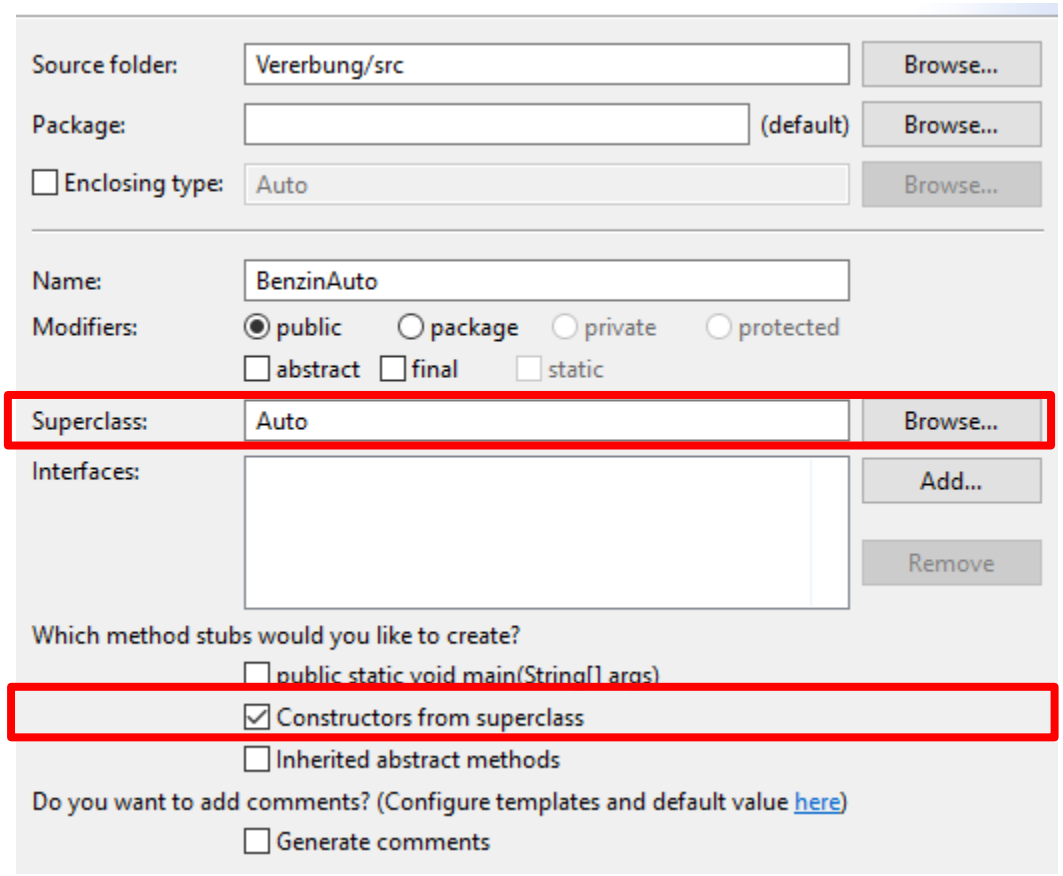
Vererbung

Technische Berufsschule Zürich

Auto.java

```
1
2  public class Auto {
3
4      private String marke;
5      private String modell;
6      private int anzahlTueren;
7      private double laenge;
8      private double gewicht;
9      private double kmstand;
10
11     public Auto(String marke, String modell) {
12         this.marke = marke;
13         this.modell = modell;
14     }
15     public double getKmStand() {
16         return kmstand;
17     }
18     protected void setKmStand(double kmstand) {
19         this.kmstand = kmstand;
20     }
21 }
```

Vererbung


Source folder:
Vererbung/src
Browse...
Package:
(default)
Browse...
Enclosing type:
Auto
Browse...
Name:
BenzinAuto
Modifiers:
public
package
private
protected
abstract
final
static
Superclass:
Auto
Browse...
Interfaces:
Add...
Remove
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments

Vererbung

Technische Berufsschule Zürich

Auto.java | BenzinAuto.java

```
1
2  public class BenzinAuto extends Auto {
3
4      private double tankinhalt;
5      private double tankvolumen;
6      private double treibstoffverbrauch;
7
8      public BenzinAuto(String marke, String modell) {
9          super(marke, modell);
0      }
1  }
2
```

Vererbung

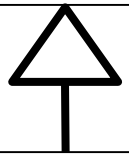
Technische Berufsschule Zürich

Auto.java | BenzinAuto.java | ElektroAuto.java

```
1
2  public class ElektroAuto extends Auto {
3
4      private double batteriekapazitaet;
5      private int ladedauer;
6      private double energieverbrauch;
7
8      public ElektroAuto(String marke, String modell) {
9          super(marke, modell);
0      }
1  }
```

super

```
public class Auto {
        …
        public Auto(String marke, String modell) {
                this.marke = marke;
                this.modell = modell;
        }
        …
}
```



```
public class BenzinAuto extends Auto {
        …
        public BenzinAuto(String marke, String modell) {
                super(marke, modell);
        }
        …
}
```

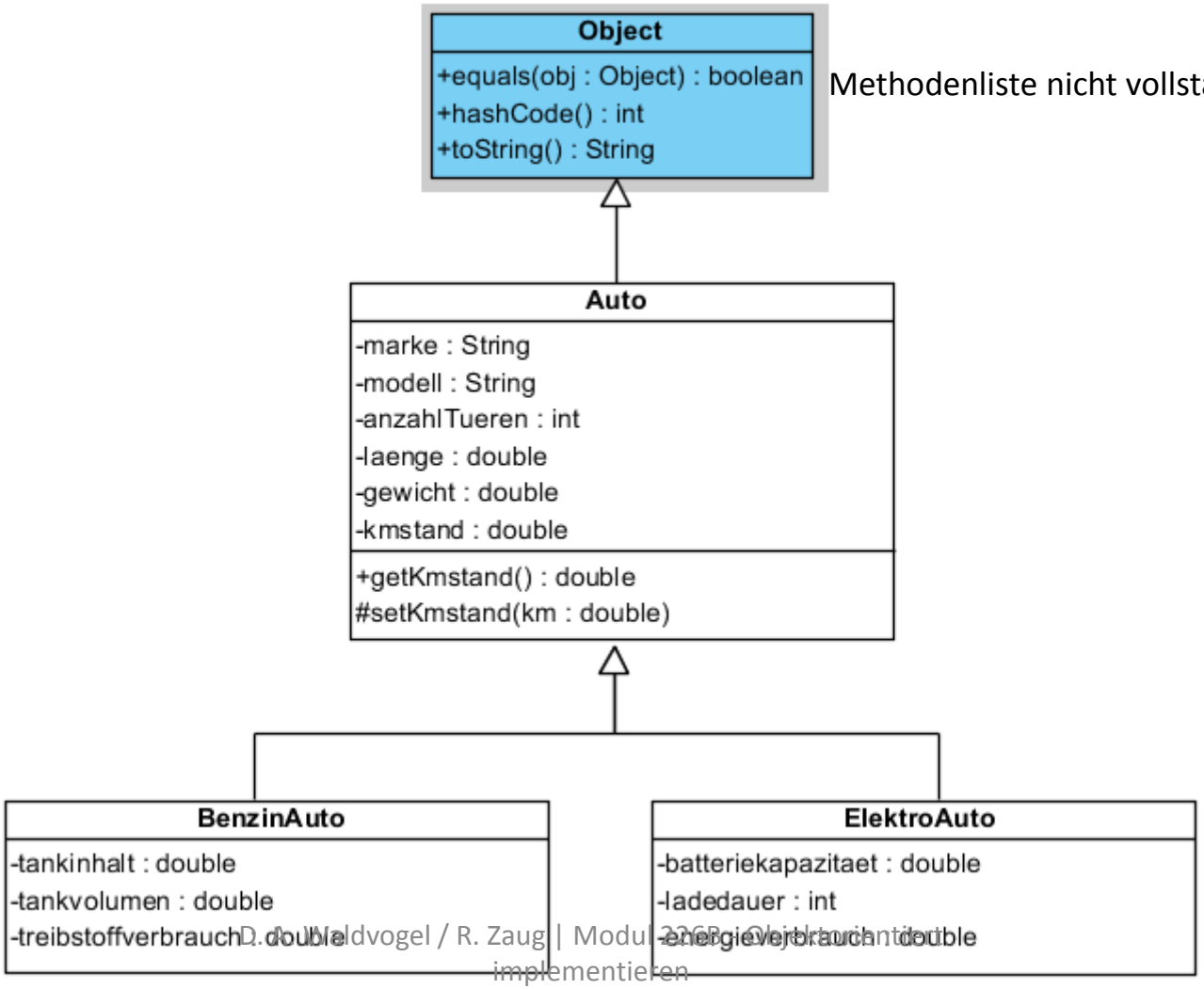
super

Technische Berufsschule Zürich

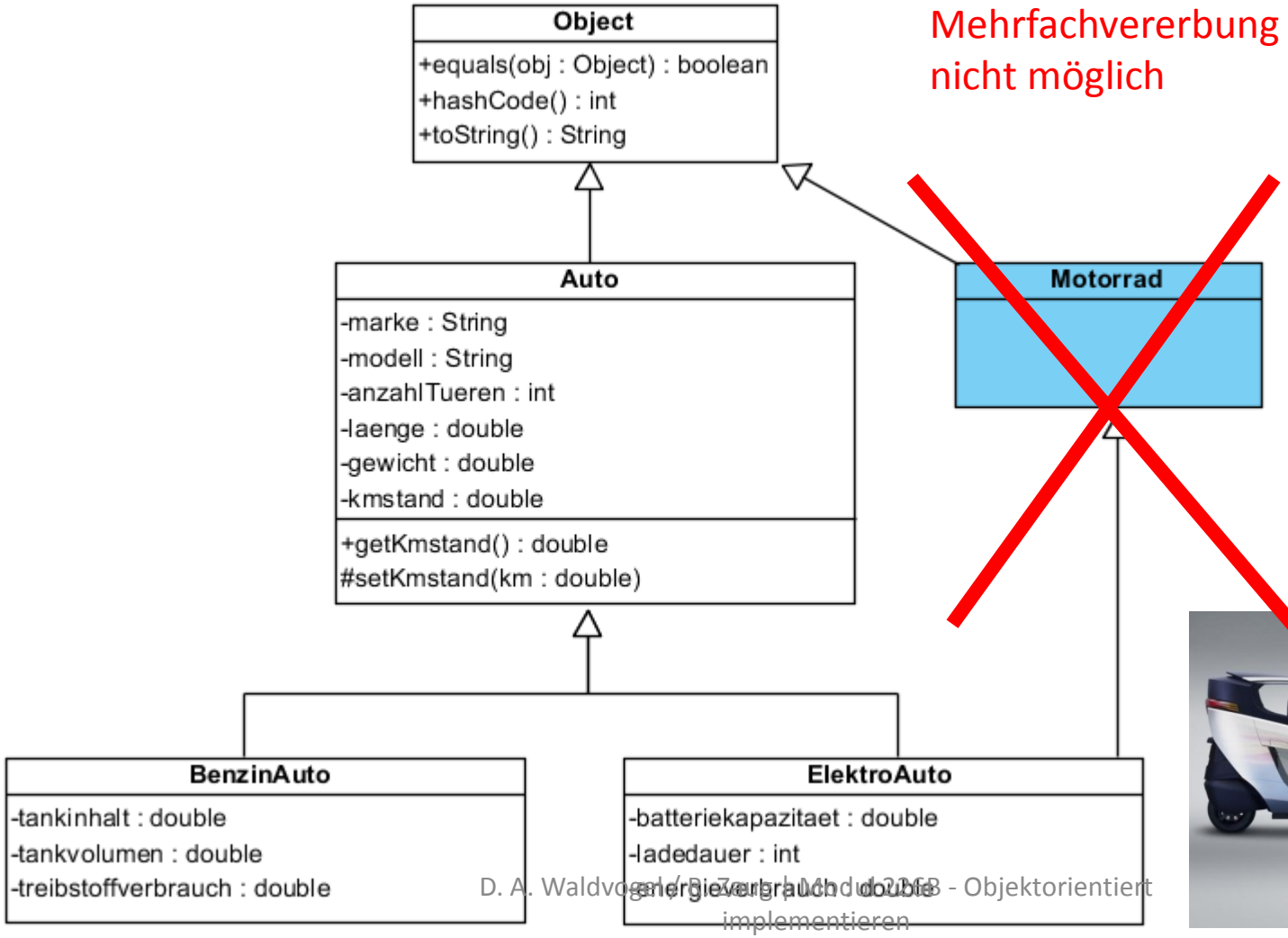
Grundsätze:

- super() ruft den Konstruktor der Oberklasse auf
- Der super Aufruf in einem Konstruktor muss immer die erste Anweisung sein
- Definiert die Oberklasse einen Konstruktor **mit** Parameter, muss die Unterklasse **zwingend** den oberen Konstruktor mit super(…) aufrufen
- Definiert die Oberklasse einen Konstruktor **ohne** Parameter, ist der Aufruf in der Unterklasse **fakultativ** – aber nicht falsch, der Konstruktor der Oberklasse wird implizit sowieso aufgerufen

Mutter aller Klassen



Mehrfachvererbung



Mehrfachvererbung ist in Java nicht möglich

Übung

- Zeit: 30’
- **Auftrag:**
  - Laden Sie sich die Übung U_Vererbung.pdf von BSCW herunter
  - Lösen Sie die Übung in Ihrer Entwicklungsumgebung

Fragen?