# KFZ-Beispiel

Größeres zusammenfassendes Beispiel für alle erlernten Aspekte der OOP,

insbesondere für die Vererbung.

Darüber hinaus soll es demonstrieren,

dass sich OOP-Software nachträglich in alle möglichen Richtungen erweitern lässt.

Dies ist ein Vorteil der OOP-Software gegenüber "Konkurrenzprodukten"

u. ein marktwirtschaftlicher Vorteil.

Es sollen drei ähnliche Klassen implementiert werden: PKW, LKW u. Traktoren.

Alle dreie haben die Attribute motorleistung u. eigengewicht gemeinsam.

Jede Klasse bringt ein zusätzliches Attribut ein:

* PKW: sitzplaetze (deren Anzahl)
* LKW: ladevolumen (im Kubikmetern)
* Traktor: zuglast (in kg)

Da die Klassen je zwei gemeinsame Attribute haben,

werden diese ausgelagert in eine Superklasse "KFZ".

Für alle Fahrzeuge soll eine Methode zur Ausgabe implementiert werden.

Um nicht in allen drei Subklassen die Ausgabe der gemeinsamen Attribute wiederholen zu müssen,

wird in der Klasse "KFZ" eine Methode "schreibe\_KFZ()" implementiert,

die die gemeinsamen Attribute ausgibt.

In den Subklassen wird darauf Bezug genommen

u. nur das jeweils dazugekommene Attribut ergänzt.

**KFZ**

motorleistung: int

eigengewicht: int

schreibe\_KFZ()

**Traktor**

zuglast: int

schreibe\_Traktor()

**LKW**

ladevolumen: int

schreibe\_LKW()

**PKW**

sitzplaetze: int

schreibe\_PKW()

ferrari

trecker

brummi

trabi

Verbesserungsmöglichkeiten:

1. Wir definieren Konstruktoren in den Subklassen,  
   um nicht jeweils einen Aufruf für den Standardkonstruktor zu benötigen  
   u. 3 weitere für die Zuweisung.  
   Die Zuweisung der beiden gemeinsamen Attribute in den drei Subklassen

wäre Codeduplizierung.  
Deshalb lagern wir diese aus in einen Konstruktor der Superklasse,  
den wir mit "super()" aufrufen u. die beiden geerbten Attribute übergeben.  
Wir müssen dann in den Subklassen jeweils nur das hinzugekommene

Attribut setzen durch Weiterreichen des Parameters.

1. Wir wollen alle Ausgabemethoden gleich nennen: "schreibe()".  
   Dabei haben wir den Effekt des Überschreibens von Methoden in den Subklassen.  
   Zur Laufzeit sucht sich der Compiler die richtige Implementierung  
   anhand der Klassenzugehörigkeit des Attributes heraus,  
   mit dem die Methode aufgerufen wurde.  
   Somit sparen wir uns Aufwand für die unterschiedliche Benennung  
   u. eliminieren die Fehlerquelle,

versehentlich die falsche Methode aufzurufen.

**KFZ**

motorleistung: int

eigengewicht: int

schreibe ()

**Traktor**

zuglast: int

schreibe ()

**LKW**

ladevolumen: int

schreibe ()

**PKW**

sitzplaetze: int

schreibe ()

ferrari

trecker

brummi

trabi

1. Erweiterung: "zur Seite":

Weitere Subklassen neben den bisherigen unter der Superklasse:  
Motorrad u. Panzer neben PKW, LKW u. Traktor.

Beim Motorrad mit zusätzlichem booleaschen Attribut "beiwagen",  
beim Panzer int-Attribut "kaliber" in cm.  
Instanzen leo von Panzer u. harley von Motorrad.

1. Erweiterung: "in die Tiefe":  
   Weiteres Attribut in der Superklasse

Attribut "anschaffungspreis" für alle Subklassen  
Durch Vererbungsstruktur muss es nur in der Superklasse eingefügt werden.

1. Erweiterung: "nach unten":

Bisherige Subklasse wird zur Superklasse, Instanzen verschieben sich

Bisherige Subklase "PKW" wird differenziert in   
"Sportwagen", "Strassenwagen" u. "Gelaendewagen".  
Bei Sportwagen kommt Attribut "turbo",  
bei Gelaendewagen "seilwinde" hinzu.  
Kein zusätzliches Attribut bei Klasse "Strassenwagen".

Instanzen verschieben sich.

Neue Instanz "jeep" von Gelaendewagen

1. Erweiterung: "nach oben":

Bisherige Superklasse wird zur Subklasse, neue Superklasse

Klasse PKW wird umbenannt in "Strassenfahrzeug".

Daneben Schiff u. Flugzeug.

Neue Superklasse darüber: Fortbewegungsmittel.

Instanzen für die neuen Klassen.

Verschiebung der Attribute

**KFZ**

motorleistung: int

eigengewicht: int

schreibe\_KFZ()

1. Erweiterung: "in die Tiefe":  
   Weiteres Attribut in der Superklasse
2. Erweiterung: "zur Seite":

Weitere Subklassen neben den bisherigen unter der Superklasse:

**Traktor**

zuglast: int

schreibe\_Traktor()

**LKW**

ladevolumen: int

schreibe\_LKW()

**PKW**

sitzplaetze: int

schreibe\_PKW()

anschaffungspreis: int

…

**Panzer**

kaliber: int

schreibe\_Panzer()

**Gelaendewagen**

seilewinde: boolean

schreibe\_Gelaendewagen()

**Strassenwagen**

-

schreibe\_Strassenwagen()

**Sportwagen**

turbo: boolean

schreibe\_Sportwagen()

3. Erweiterung: "nach unten":

Bisherige Subklasse wird zur Superklasse, Instanzen verschieben sich