



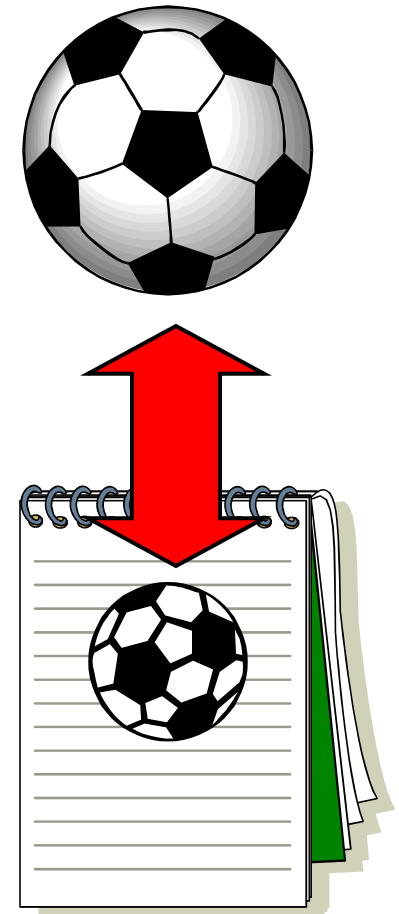
Java Einführung

Objektorientierte

Grundkonzepte

Inhalt

- Verständnis der grundlegenden Konzepte der Objektorientierung:
 - Objekte
 - Nachrichten
 - Kapselung
 - Klassen und Instanzen
 - Vererbung
 - Polymorphismus
- Darstellung in UML



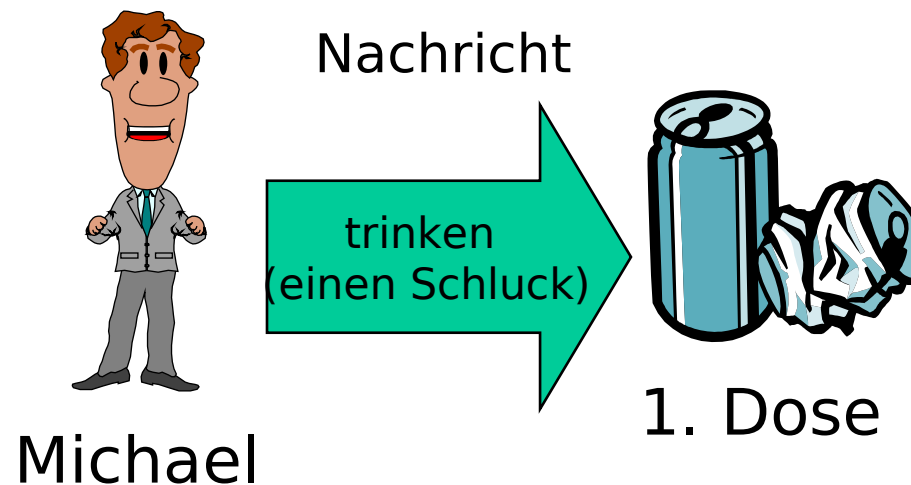
Objekte

- „Dinge“ aus der realen Welt
(Gegenstände, Personen, Rechte,...)
- Haben eine Identität und besitzen:
 - Eigenschaften (**Attribute**)
 - Verhalten (**Operationen**)
- Kommunizieren mit anderen Objekten durch Nachrichten



Nachrichten und Methoden

- Nachrichten sind Signale von einem Objekt an ein anderes.
- Nachrichten lösen Verhalten (Operationen) des anderen Objekts aus.



Kapselung (Encapsulation, Data Hiding)

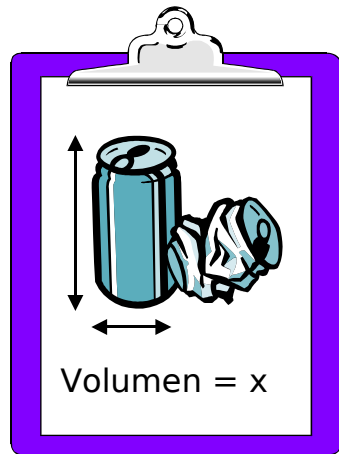
- Objekte können Attribute und Operationen (und damit Komplexität) verbergen.
- Operationen werden vor dem Zugriff von außen geschützt.
- Die allgemein zugänglichen Operationen nennt man **Interface**



Wieviel Flüssigkeit ist in dieser Dose?
Kann jeder diese Dose verschließen?

Klassen (-definition)

- Bauplan für ähnliche Objekte.
- Beschreibt Operationen und Attribute die eine Instanz bekommt.
- **Instanzen** werden aus Klassen erzeugt



Klassendefinition



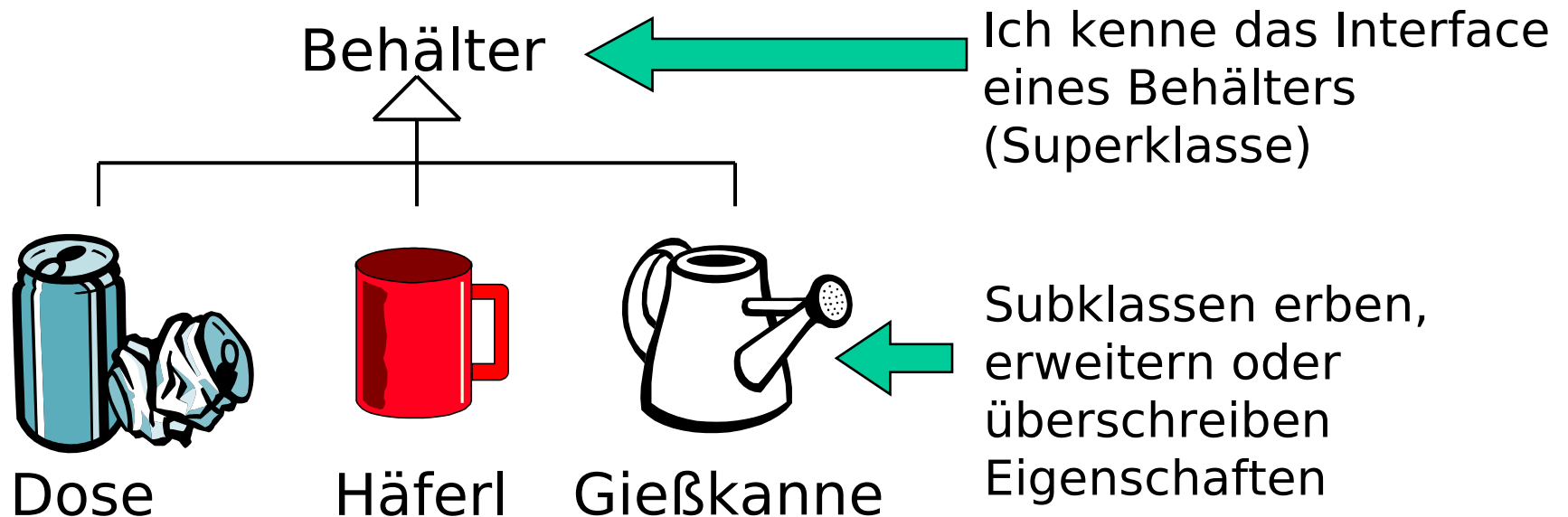
Instanz Nr. 1



Instanz Nr. 2

Generalisierung, Vererbung (Inheritance)

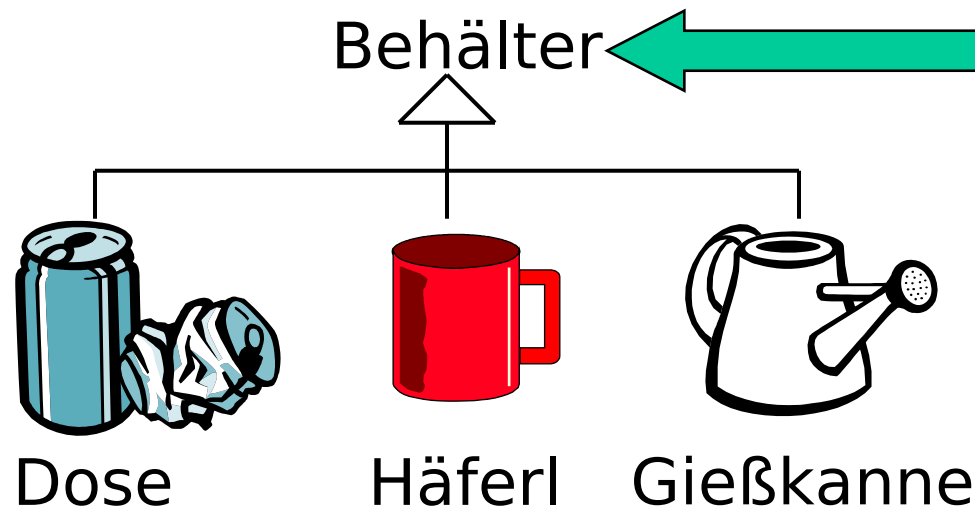
- Hierarchische Beziehung zwischen Klassen mit ähnlichen Eigenschaften.
- Kinder (Subklassen) erben die Eigenschaften von Eltern (Superklassen).
- Stellt ein gemeinsames Interface sicher.



Polymorphismus

(Griechisch: Vielgestaltigkeit)

- Subklassen können auch als Instanz der Superklasse (Behälter) behandelt werden, da sie durch die Vererbung ein gemeinsames Interface haben.



Ich verwende einen Behälter

Jede Art von Behälter hat das gleiche Interface. Einige Operationen reagiert aber bei unterschiedlichen Behältertypen ein bisschen anders!

Modellierung und Abstraktion

- Eigenschaften und Verhalten müssen modelliert werden.
 - Modellierung ist Vereinfachung!
 - Modellierung ist Einschränkung!
 - **Was ist das Wesentliche?**
Hängt vom Zweck ab!

Ist diese Nutzung eines Häferls durch das Modell vorgesehen?



Vorteile der Objekt-Orientierung

- Objekte in Programmen entsprechen realen Objekten (von Analyse bis zur Implementierung)
- Modularität durch die Kapselung in Objekte
- Wiederverwendung von Objektbibliotheken (Interfaces)
- Geringere Kosten bei Wartung weil der Programmcode übersichtlicher wird.

Die Unified Modeling Language

- UML ist eine von der OMG standardisierte Notation zur Darstellung und Entwicklung von objekt-orientierten Systemen
- Die Notation enthält unter anderem Diagramme für die Beschreibung von:
 - Anforderungen (Use Case Diagram)
 - Verhalten/Ablauf (Sequence D., Communication D.)
 - **Logischer Aufbau des Systems (Klassendiagramm)**



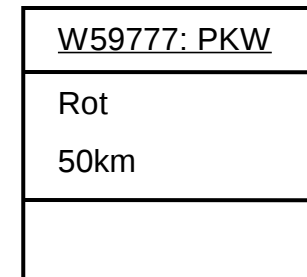
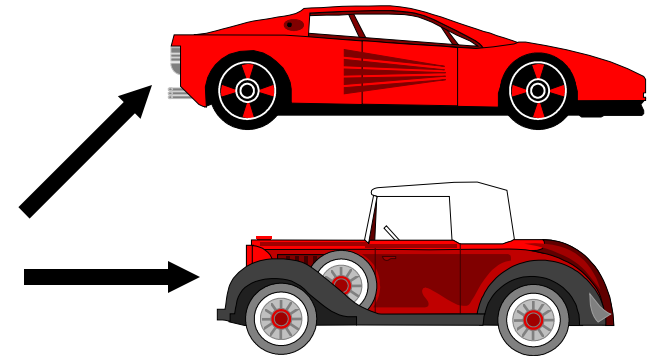
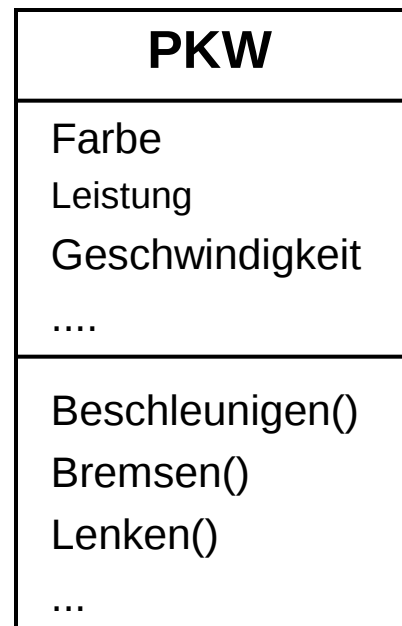
Darstellung von Klassen in UML Diagrammen

Klassen sind die Baupläne für Objekte (Instanzen)

Darstellung in UML



Beispiel einer Klasse in UML



Elemente Klassendiagramm I

Klassen



Beziehungen (Paths/Associations): für die Übermittlung von Nachrichten (Messages = Aufruf von Methoden)



Richtung der Beziehungen (Navigability): Nachrichten können nur in diese Richtung gehen



Elemente Klassendiagramm II

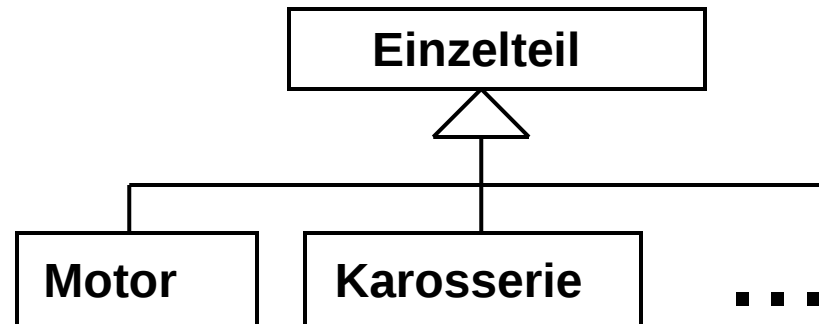
Kardinalitäten (Multiplicity): Anzahl der möglichen Instanzen



Aggregation (Aggregation/Composition): Besteht-aus Beziehung



Generalisierung (Generalization): is-a Beziehung -> Vererbung in Java

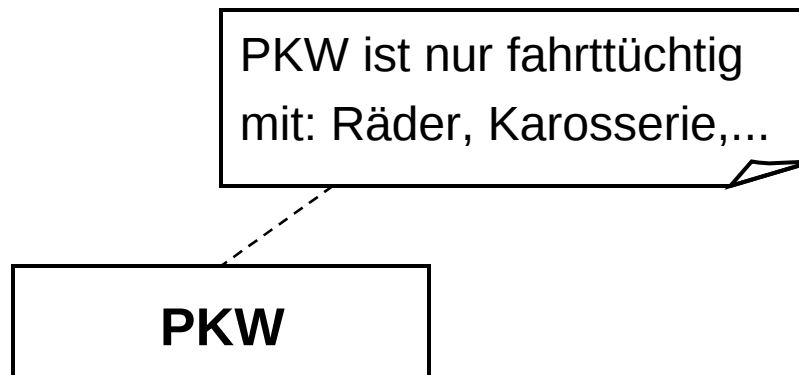


Elemente Klassendiagramm III

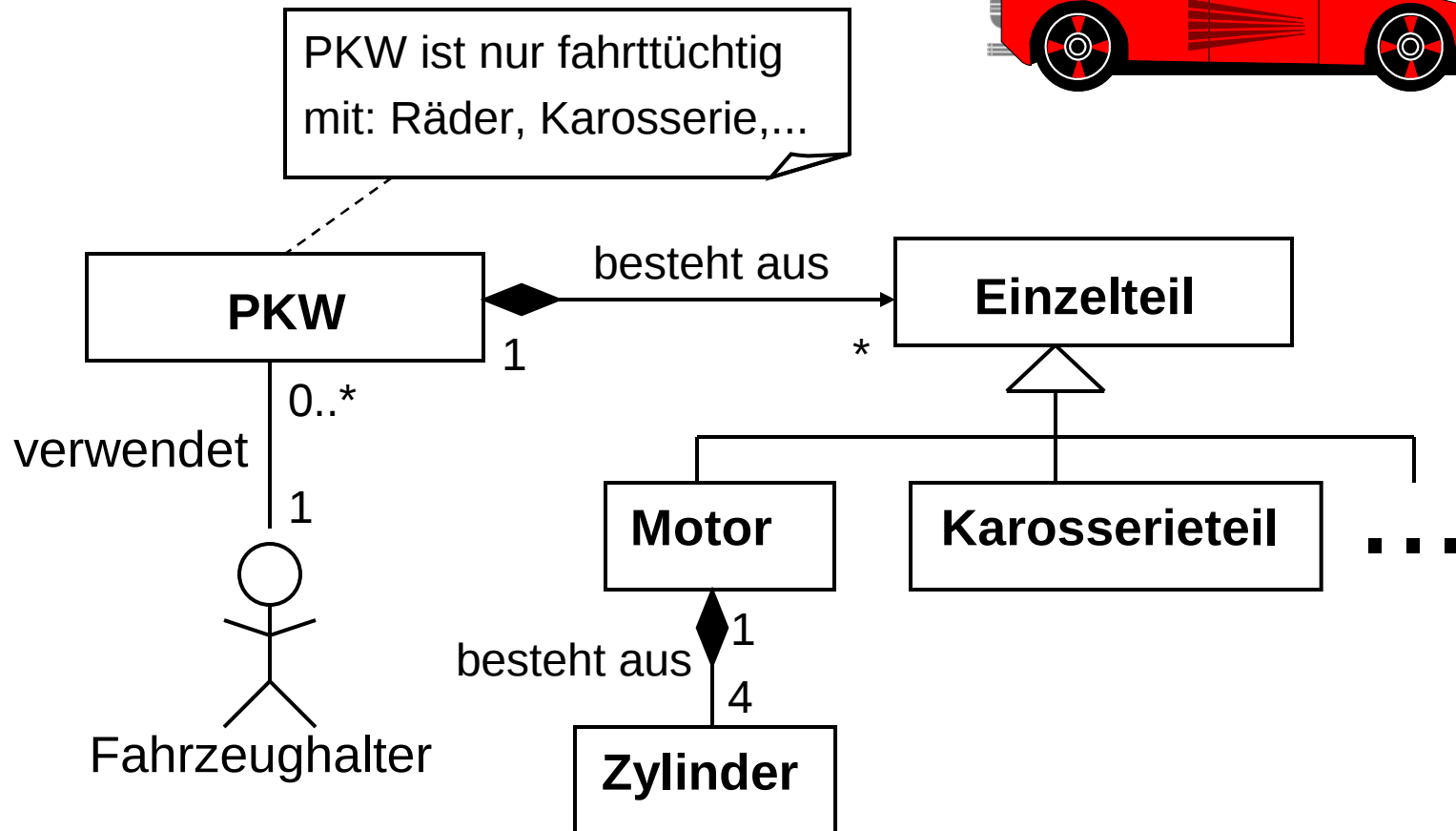
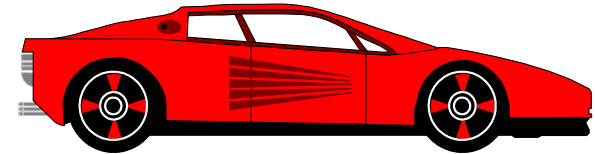
Abhängigkeiten (Dependencies)



Anmerkungen oder Einschränkung



Beispiel: Klassendiagramm



Was Sie nach dieser Einheit wissen sollten...

- Was sind Objekte, Instanzen und Klassen?
- Was sind die wichtigsten Konzepte der Objektorientierung und was bedeuten Sie?
- Wie werden Klassen in UML Klassendiagrammen dargestellt?