

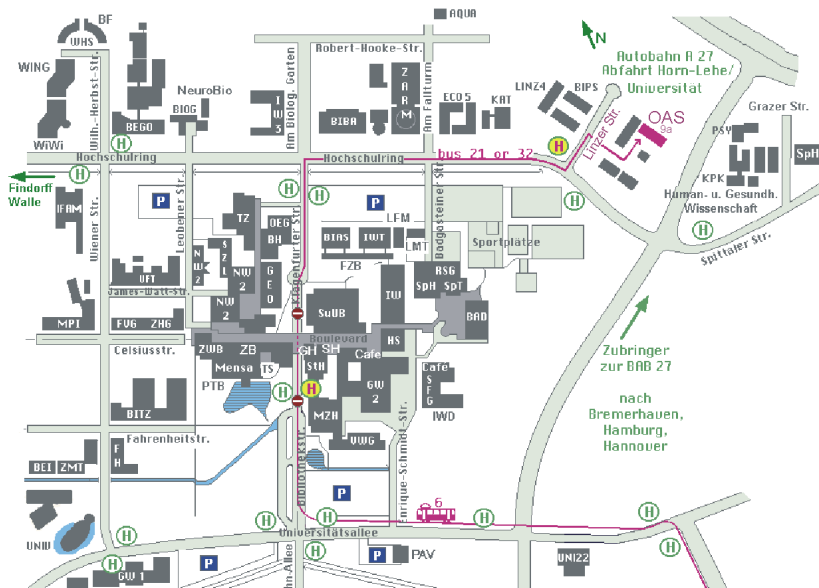
## Objektorientierte Modellierung

- Identifiziere Aktoren
- Beschreibe Anwendungsfälle
- Bestimme statisches Modell
  - Identifiziere Objekte
  - Identifiziere Eigenschaften der Objekte
  - Bestimme Assoziationen zwischen Objekten
  - Fasse Objekte zu Klassen zusammen
  - Ordne Klassen in Vererbungshierarchien ein
  - Bestimme Multiplizitäten der Assoziationen
- Erstelle Verhaltensmodell
  - Identifiziere Ereignisse und modelliere Interaktionen in Anwendungsfällen
  - Identifiziere Verhalten der Objekte
  - Beschreibe das Verhalten (Vor- und Nachbedingungen)

Schwierig: Dinge im Abstrakten beschreiben.

Einfacher: von konkreten Anwendungsfällen ausgehen.

# Beispiel Raumplanung



## Operation: Einkauf

- Kunde König möchte Flaschenhalter kaufen; besitzt mehrere Fahrräder

## Operation: Einkauf

- Kunde König möchte Flaschenhalter kaufen; besitzt mehrere Fahrräder
- König tritt in den Laden und zückt sein PDA

## Operation: Einkauf

- Kunde König möchte Flaschenhalter kaufen; besitzt mehrere Fahrräder
- König tritt in den Laden und zückt sein PDA
- PDA bietet alle Räder des Kunden an

## Operation: Einkauf

- Kunde König möchte Flaschenhalter kaufen; besitzt mehrere Fahrräder
- König tritt in den Laden und zückt sein PDA
- PDA bietet alle Räder des Kunden an
- König wählt sein 28"-Rennrad mit Rahmen-Standardbohrung aus; Rahmenfarbe: magenta

## Operation: Einkauf

- Kunde König möchte Flaschenhalter kaufen; besitzt mehrere Fahrräder
- König tritt in den Laden und zückt sein PDA
- PDA bietet alle Räder des Kunden an
- König wählt sein 28"-Rennrad mit Rahmen-Standardbohrung aus; Rahmenfarbe: magenta
- König wählt "Hinzufügen" aus



## Operation: Einkauf

- Kunde König möchte Flaschenhalter kaufen; besitzt mehrere Fahrräder
- König tritt in den Laden und zückt sein PDA
- PDA bietet alle Räder des Kunden an
- König wählt sein 28"-Rennrad mit Rahmen-Standardbohrung aus; Rahmenfarbe: magenta
- König wählt "Hinzufügen" aus
- König gibt "Flaschenhalter" ein und quittiert

## Operation: Einkauf

- Kunde König möchte Flaschenhalter kaufen; besitzt mehrere Fahrräder
- König tritt in den Laden und zückt sein PDA
- PDA bietet alle Räder des Kunden an
- König wählt sein 28"-Rennrad mit Rahmen-Standardbohrung aus; Rahmenfarbe: magenta
- König wählt "Hinzufügen" aus
- König gibt "Flaschenhalter" ein und quittiert
- PDA nimmt Verbindung mit Laden-Host-Rechner auf

## Operation: Einkauf

- Kunde König möchte Flaschenhalter kaufen; besitzt mehrere Fahrräder
- König tritt in den Laden und zückt sein PDA
- PDA bietet alle Räder des Kunden an
- König wählt sein 28"-Rennrad mit Rahmen-Standardbohrung aus; Rahmenfarbe: magenta
- König wählt "Hinzufügen" aus
- König gibt "Flaschenhalter" ein und quittiert
- PDA nimmt Verbindung mit Laden-Host-Rechner auf
- PDA übermittelt Anfrage für Flaschenhalter (verschlüsselt!)

# Beispiel Fahrradladen

## Operation: Einkauf

- Kunde König möchte Flaschenhalter kaufen; besitzt mehrere Fahrräder
- König tritt in den Laden und zückt sein PDA
- PDA bietet alle Räder des Kunden an
- König wählt sein 28"-Rennrad mit Rahmen-Standardbohrung aus; Rahmenfarbe: magenta
- König wählt "Hinzufügen" aus
- König gibt "Flaschenhalter" ein und quittiert
- PDA nimmt Verbindung mit Laden-Host-Rechner auf
- PDA übermittelt Anfrage für Flaschenhalter (verschlüsselt!)
- Host-Rechner übermittelt Sortiment Flaschenhalter (verschlüsselt!)

## Operation: Einkauf

- Kunde König möchte Flaschenhalter kaufen; besitzt mehrere Fahrräder
- König tritt in den Laden und zückt sein PDA
- PDA bietet alle Räder des Kunden an
- König wählt sein 28"-Rennrad mit Rahmen-Standardbohrung aus; Rahmenfarbe: magenta
- König wählt "Hinzufügen" aus
- König gibt "Flaschenhalter" ein und quittiert
- PDA nimmt Verbindung mit Laden-Host-Rechner auf
- PDA übermittelt Anfrage für Flaschenhalter (verschlüsselt!)
- Host-Rechner übermittelt Sortiment Flaschenhalter (verschlüsselt!)
- PDA prüft auf Verträglichkeit mit ausgewähltem Rad

# Beispiel Fahrradladen (Forts.)

## Operation: Einkauf (Forts.)

- PDA präsentiert alle Flaschenhalter, die passen (hartes Kriterium)

## Operation: Einkauf (Forts.)

- PDA präsentiert alle Flaschenhalter, die passen (hartes Kriterium)
- König deselektiert alle Flaschenhalter mit unpassender Farbe (weiches Kriterium)

## Operation: Einkauf (Forts.)

- PDA präsentiert alle Flaschenhalter, die passen (hartes Kriterium)
- König deselektiert alle Flaschenhalter mit unpassender Farbe (weiches Kriterium)
- König lässt nach Preis sortieren



# Beispiel Fahrradladen (Forts.)

## Operation: Einkauf (Forts.)

- PDA präsentiert alle Flaschenhalter, die passen (hartes Kriterium)
- König deselektiert alle Flaschenhalter mit unpassender Farbe (weiches Kriterium)
- König lässt nach Preis sortieren
- König wählt Flaschenhalter in bestimmtem Preissegment aus

## Operation: Einkauf (Forts.)

- PDA präsentiert alle Flaschenhalter, die passen (hartes Kriterium)
- König deselektiert alle Flaschenhalter mit unpassender Farbe (weiches Kriterium)
- König lässt nach Preis sortieren
- König wählt Flaschenhalter in bestimmtem Preissegment aus
- König geht mit Auswahl zu Verkäufer

# Beispiel Fahrradladen (Forts.)

## Operation: Einkauf (Forts.)

- PDA präsentiert alle Flaschenhalter, die passen (hartes Kriterium)
- König deselektiert alle Flaschenhalter mit unpassender Farbe (weiches Kriterium)
- König lässt nach Preis sortieren
- König wählt Flaschenhalter in bestimmtem Preissegment aus
- König geht mit Auswahl zu Verkäufer
- Verkäufer Volker betrachtet Auswahl und berät

## Definition

### Anwendungsfall

- beschreibt eine Menge von Aktionssequenzen (Varianten eingeschlossen)
- jede Sequenz repräsentiert die Interaktion zwischen externen Aktoren mit dem System
- Folge ist beobachtbares Resultat, relevant für Akteur

# Anwendungsfall (Use-Case)

## Definition

### Anwendungsfall

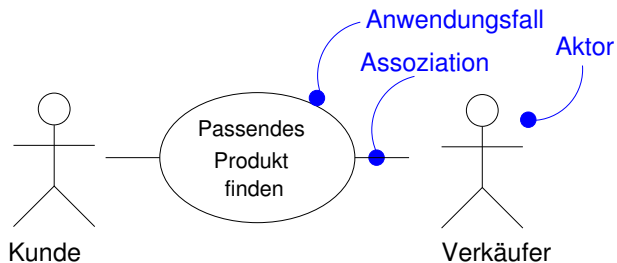
- beschreibt eine Menge von Aktionssequenzen (Varianten eingeschlossen)
- jede Sequenz repräsentiert die Interaktion zwischen externen Aktoren mit dem System
- Folge ist beobachtbares Resultat, relevant für Aktor

## Definition

### Aktor

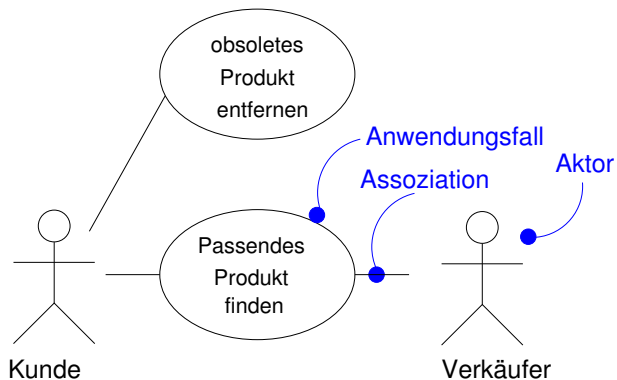
- repräsentiert eine kohärente Menge von Rollen, die von Benutzern in der Interaktion mit dem System eingenommen werden können
- können Menschen und andere Dinge sein (z.B. andere automatisierte Systeme)

# UML-Notation für Anwendungsfälle (OMG)



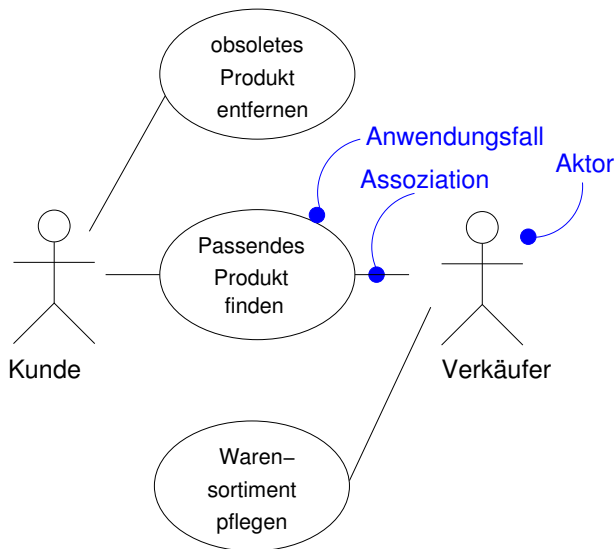
- ① Identifiziere Aktoren
- ② Betrachte System aus der Sicht der Aktoren
- ③ Bestimme Anwendungsfälle für Aktoren
  - liefert möglicherweise neue Aktoren
- ④ zurück zu 1, bis keine neuen Aktoren/Anwendungsfälle mehr gefunden werden können

# UML-Notation für Anwendungsfälle (OMG)

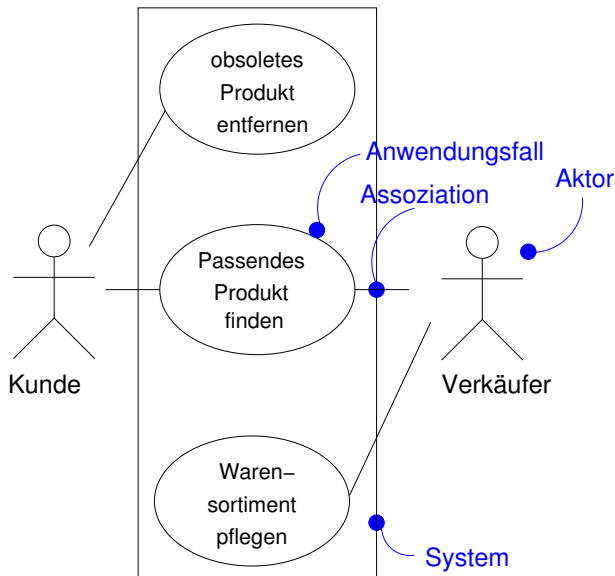




# UML-Notation für Anwendungsfälle (OMG)



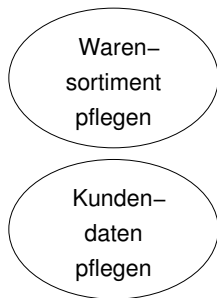
# UML-Notation für Anwendungsfälle (OMG)



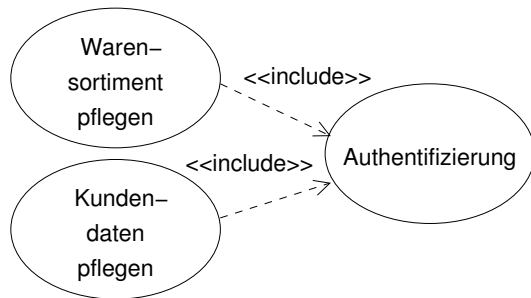
## Strukturiere Anwendungsfälle

- ① identifiziere gemeinsame Anteile in Anwendungsfällen und faktoriere entsprechend
- ② fasse ähnliche Anwendungsfälle und Aktoren in Vererbungshierarchien zusammen

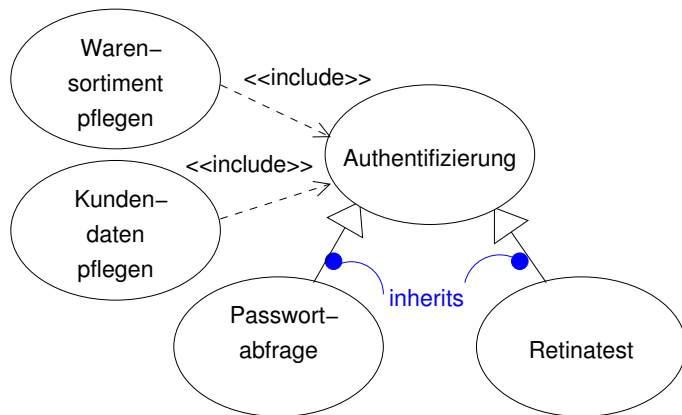
# UML-Notation für Anwendungsfälle (OMG)



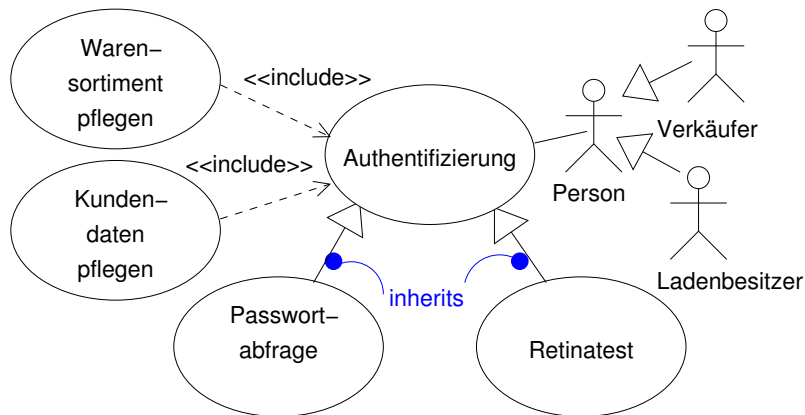
# UML-Notation für Anwendungsfälle (OMG)



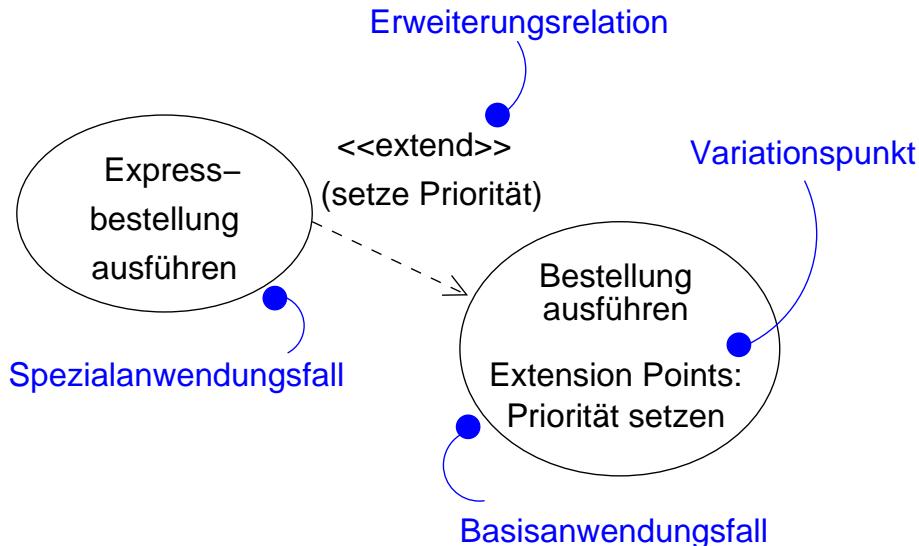
# UML-Notation für Anwendungsfälle (OMG)



# UML-Notation für Anwendungsfälle (OMG)



# UML-Notation für Anwendungsfälle (OMG)





## Anwendungsfalldiagramme:

- deklarieren voneinander unabhängige Anwendungsfälle
- beschreiben die Einbettung der Anwendungsfälle in den Systemkontext (externe Akteure)
- beschreiben (konkretisieren) die Anwendungsfälle jedoch nicht  
→ folgt später
- sind Ausgangspunkt für die objektorientierte Modellierung
  - statische Eigenschaften (Attribute)
  - dynamische Eigenschaften (Verhalten)

## Objektorientierte Modellierung

- Identifiziere Aktoren
- Beschreibe Anwendungsfälle
- Bestimme statisches Modell
  - Identifiziere Objekte
  - Identifiziere Eigenschaften der Objekte
  - Bestimme Assoziationen zwischen Objekten
  - Fasse Objekte zu Klassen zusammen
  - Ordne Klassen in Vererbungshierarchien ein
  - Bestimme Multiplizitäten der Assoziationen
- Erstelle Verhaltensmodell
  - Identifiziere Ereignisse und modelliere Interaktionen in Anwendungsfällen
  - Identifiziere Verhalten der Objekte
  - Beschreibe das Verhalten (Vor- und Nachbedingungen)

## Objektorientierte Modellierung

- Identifiziere Aktoren
- Beschreibe Anwendungsfälle
- **Bestimme statisches Modell**
  - **Identifiziere Objekte**
  - **Identifiziere Eigenschaften der Objekte**
  - **Bestimme Assoziationen zwischen Objekten**
  - **Fasse Objekte zu Klassen zusammen**
  - **Ordne Klassen in Vererbungshierarchien ein**
  - **Bestimme Multiplizitäten der Assoziationen**
- **Erstelle Verhaltensmodell**
  - Identifiziere Ereignisse und modelliere Interaktionen in Anwendungsfällen
  - Identifiziere Verhalten der Objekte
  - Beschreibe das Verhalten (Vor- und Nachbedingungen)

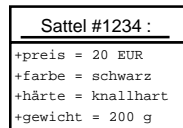
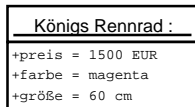
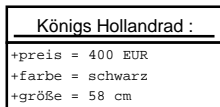
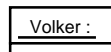
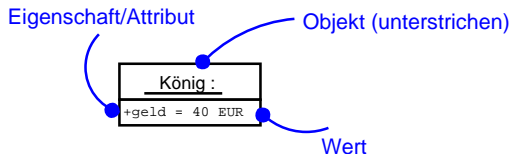
# Beispiel Fahrradladen

Identifiziere Objekte: Suche nach Substantiven in Anwendungsfällen.

## Operation: Einkauf

- **Kunde König** möchte **Sattel** kaufen; besitzt mehrere **Fahrräder**
- ...
- König wählt sein **28"-Renncrad** mit aus; **Rahmenfarbe**: magenta
- ...
- **PDA** nimmt Verbindung mit **Laden-Host-Rechner** auf
- ...
- König lässt nach **Preis** sortieren
- ...
- **Verkäufer Volker** betrachtet **Auswahl** und berät

**Objekt:** „Ding“ (Gegenstand, Entität) mit eigener Identität

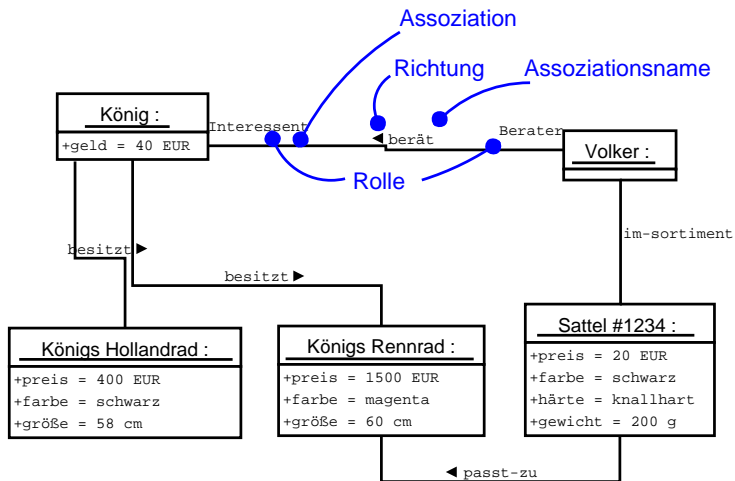


## Objektorientierte Modellierung

- Identifiziere Aktoren
- Beschreibe Anwendungsfälle
- Bestimme statisches Modell
  - Identifiziere Objekte
  - Identifiziere Eigenschaften der Objekte
  - **Bestimme Assoziationen zwischen Objekten**
  - Fasse Objekte zu Klassen zusammen
  - Ordne Klassen in Vererbungshierarchien ein
  - Bestimme Multiplizitäten der Assoziationen
- Erstelle Verhaltensmodell
  - Identifiziere Ereignisse und modelliere Interaktionen in Anwendungsfällen
  - Identifiziere Verhalten der Objekte
  - Beschreibe das Verhalten (Vor- und Nachbedingungen)

# Anwendungsfall → Objekte

**Assoziation:** Beziehung zwischen Objekten



## Objektorientierte Modellierung

- Identifiziere Aktoren
- Beschreibe Anwendungsfälle
- Bestimme statisches Modell
  - Identifiziere Objekte
  - Identifiziere Eigenschaften der Objekte
  - Bestimme Assoziationen zwischen Objekten
  - Fasse Objekte zu Klassen zusammen
  - Ordne Klassen in Vererbungshierarchien ein
  - Bestimme Multiplizitäten der Assoziationen
- Erstelle Verhaltensmodell
  - Identifiziere Ereignisse und modelliere Interaktionen in Anwendungsfällen
  - Identifiziere Verhalten der Objekte
  - Beschreibe das Verhalten (Vor- und Nachbedingungen)



## Instanz-Ebene

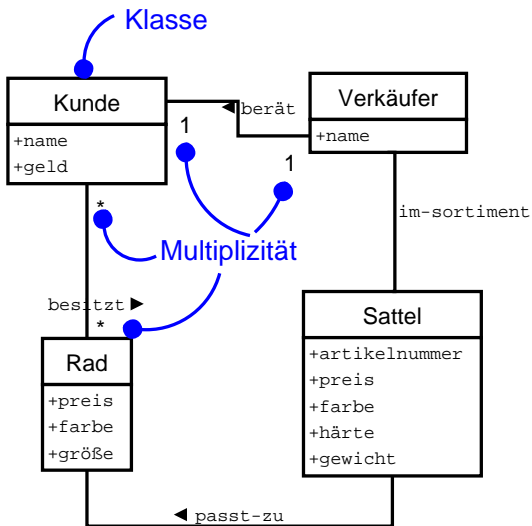
- **Objektdiagramme** beschreiben statische Zusammenhänge auf der Ebene einzelner, konkreter Dinge

## Schema-Ebene

- **Klassendiagramme** beschreiben statische Zusammenhänge unabhängig von Details konkreter Objekte, auf der Ebene mehrerer gleichartiger Dinge

# Objekte → Klassen

**Klasse:** Menge von gleichartigen Objekten mit gemeinsamen Eigenschaften



## Objektorientierte Modellierung

- Identifiziere Aktoren
- Beschreibe Anwendungsfälle
- Bestimme statisches Modell
  - Identifiziere Objekte
  - Identifiziere Eigenschaften der Objekte
  - Bestimme Assoziationen zwischen Objekten
  - Fasse Objekte zu Klassen zusammen
  - Ordne Klassen in Vererbungshierarchien ein
  - Bestimme Multiplizitäten der Assoziationen
- Erstelle Verhaltensmodell
  - Identifiziere Ereignisse und modelliere Interaktionen in Anwendungsfällen
  - Identifiziere Verhalten der Objekte
  - Beschreibe das Verhalten (Vor- und Nachbedingungen)

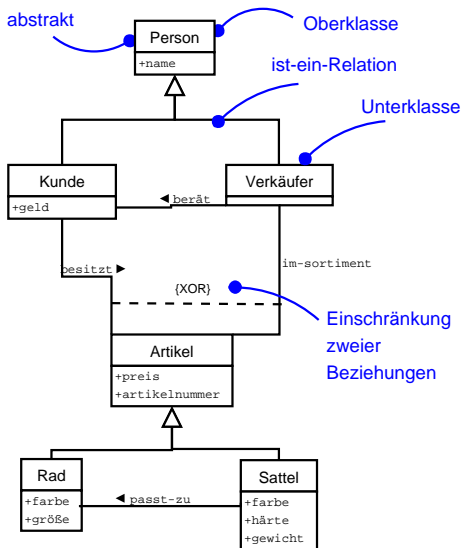
- sind spezielle Beziehung auf Schema-Ebene
- beschreiben Beziehungen zwischen einer allgemeineren Klasse (Oberklasse, Superclass) und einer spezielleren Klasse (Unterklasse, Subclass)
- die Unterklasse „erbt“ die Eigenschaften der Oberklasse:
  - Attribute
  - Methoden und deren Aufrufchnittstellen
  - Assoziationen

- sind spezielle Beziehung auf Schema-Ebene
- beschreiben Beziehungen zwischen einer allgemeineren Klasse (Oberklasse, Superclass) und einer spezielleren Klasse (Unterklasse, Subclass)
- die Unterklasse „erbt“ die Eigenschaften der Oberklasse:
  - Attribute
  - Methoden und deren Aufrufchnittstellen
  - Assoziationen

Unterklassen können

- neue Attribute, Methoden und Assoziationen definieren
- ererbte Attribute, Methoden und Assoziationen redefinieren (überschreiben)
- von mehreren Oberklassen erben (Mehrfachvererbung)

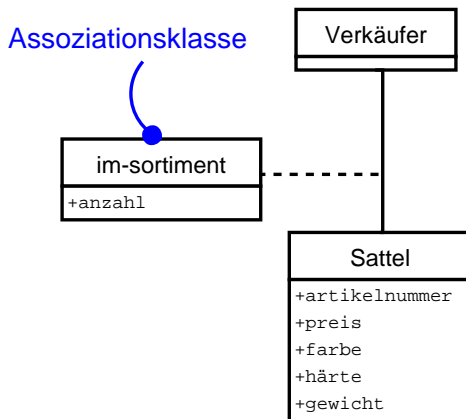
# Klassen → Klassenhierarchien



# Attributierte Assoziationen

## Assoziationsklassen

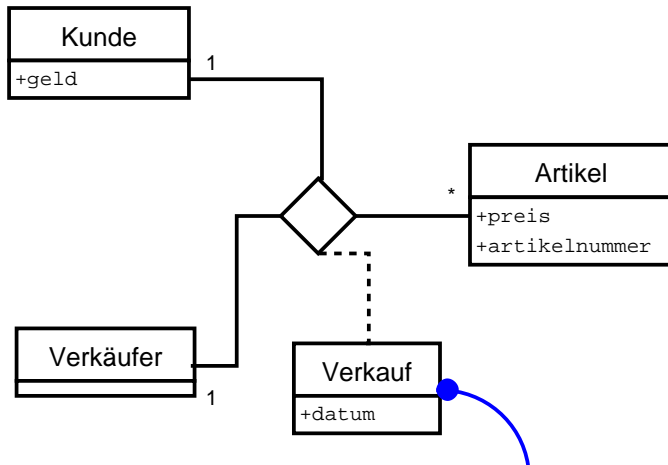
- beschreiben Beziehungen, die gleichzeitig Klassen sind
- werden genutzt, um Assoziationen Attribute zuzuordnen
- können eigene Beziehungen eingehen



# Mehrstellige Assoziationen

## Assoziationsklassen

- können auch mehrere Klassen in Beziehung setzen



Assoziationsklasse



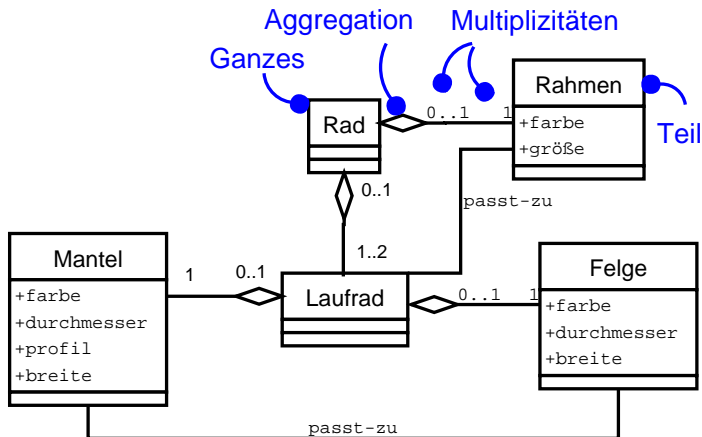
# Aufbau von Objekten



# Aufbau von Objekten

## Aggregation

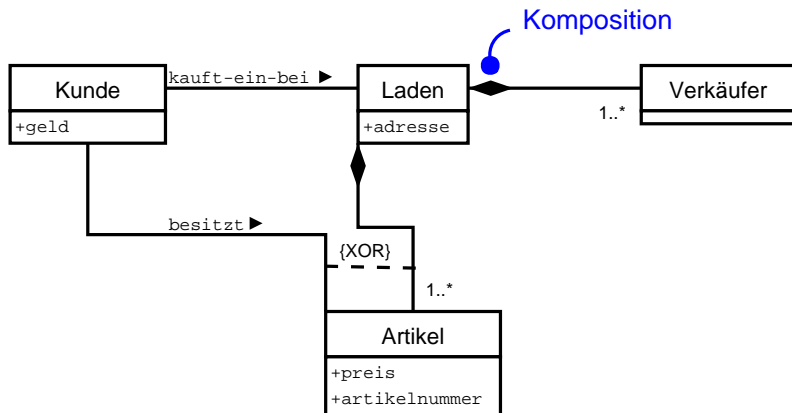
- ist spezielle Assoziation zur Verdeutlichung von „Teil-Ganzes-Beziehungen“
- beschreibt Zusammenfassung von Komponenten zu einem Aggregat



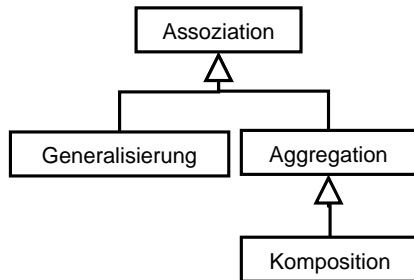
# Aufbau von Objekten

**Komposition** ist spezielle Aggregation:

- Existenz der Komponenten ist an die Existenz des Aggregats gekoppelt,
- jede Komponente gehört zu genau einem Aggregat (strong ownership)



# Vergleich der Assoziationstypen



- Vererbung: ist-ein-Relation (Liskovs Substitutionsprinzip erfüllt)
- Aggregation: teil-von-Relation
- Komposition: teil-von-Relation
  - Teil gehört zu genau einem Ganzen
  - Teil existiert nur im Kontext des Ganzen
- allgemeine Assoziation: sonst