

# Modul Softwaretechnik mit 9 oder 6 ECTS

## Datenmodellierung mit UML

*Peter Forbrig*

## Datenmodellierung mit UML

### Unified Modeling Language

- Objekte, Klassen und Assoziationen
- Multiplizitäten - Kardinalitäten
- Rollen (Wiederholung)
- Einfache Abbildung von Modellen auf Tabellen
- Empfehlungen zur Qualitätssicherung
- Bewertung
- Datenmodelle und Quelltext

## Objekte, Klassen und Assoziationen (Wiederh.)

### **Objekt** (eine erste Sicht, die später erweitert wird)

- **Individuelles und identifizierbares Exemplar von**
  - Dingen
  - Personen oder
  - Begriffen der realen oder der Vorstellungswelt
- **Durch Eigenschaften (Attribute) beschrieben**

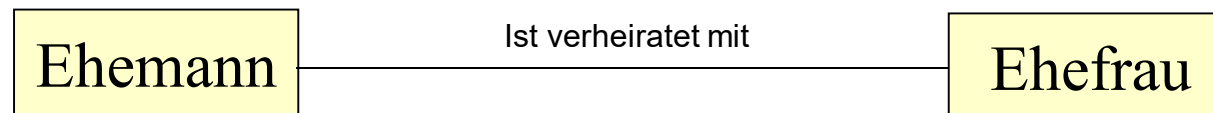
### **Klasse**

- **Zusammenfassung von Objekten mit gleichen Eigenschaften unter eindeutigem gemeinsamen Oberbegriff**
- **Dargestellt durch ein Rechteck, das den Namen der Klasse enthält**

## Objekte, Klassen und Assoziationen (Wiederh.) 2

### Assoziation

- **Wechselwirkungen und Abhängigkeiten zwischen Objekten werden durch Beziehungen (Relationen) dargestellt**
- **Assoziation ist die Zusammenfassung gleichartiger Beziehungen (Relationen) zwischen Objekten**
- **Wird durch Linie dargestellt**



## Multiplizitäten (in UML)

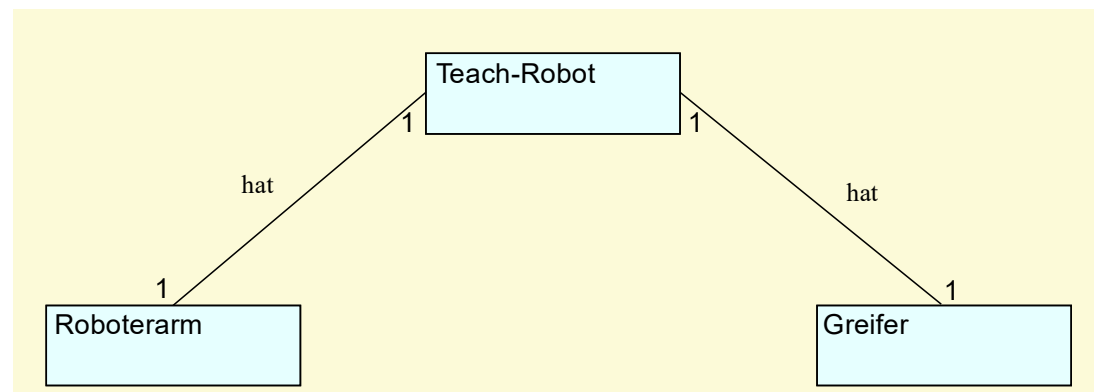
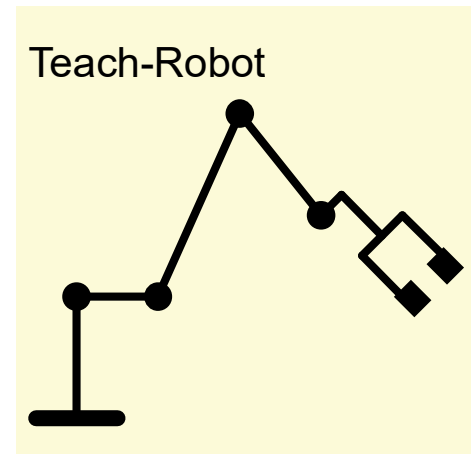
### Kardinalitäten (bei anderen Sprachen)

- **1:1-Assoziation**
- **0:1-Assoziation**
- **1:n-Assoziation**
- **n:m-Assoziation**
  - **0..n:0..m** (In UML immer n oder \*!)
  - **1..n:0..m** (n hat keinen bestimmten Wert!)
  - **0..n:1..m**
  - **1..n:1..m**

## 1:1-Assoziation

### Beispiel

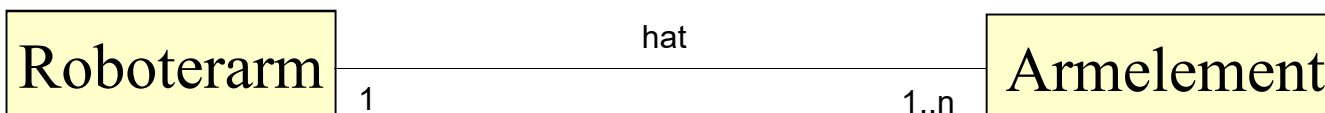
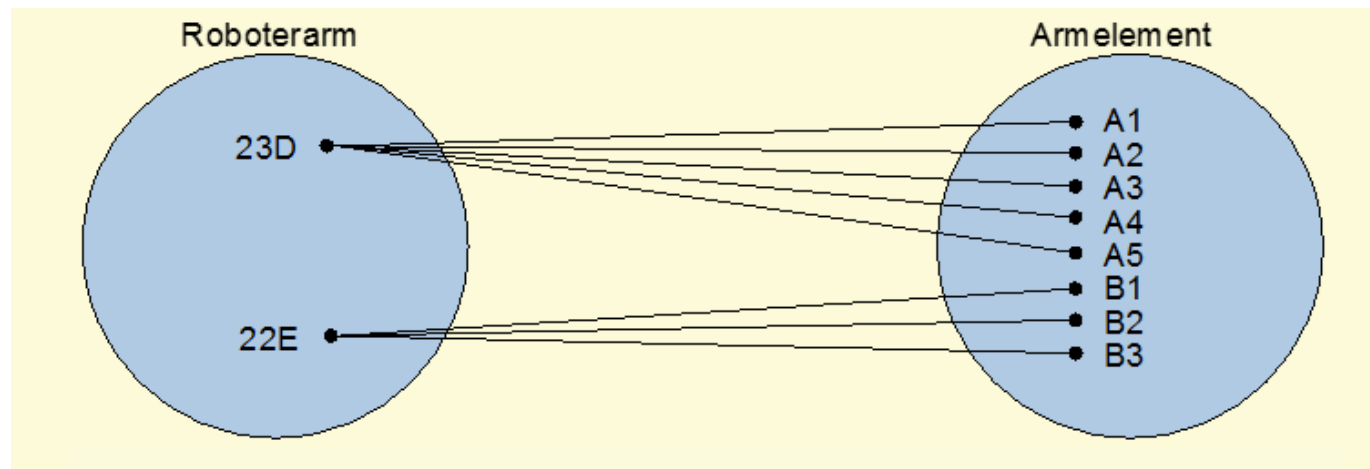
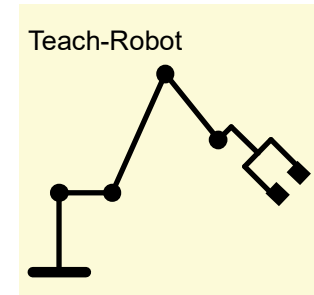
- Ein Roboter besteht aus einem Roboterarm
  - ◆ Alle Roboter vom Typ Teach-Robot besitzen einen Roboterarm und genau einen Greifer



## 1:n-Assoziation

### Beispiel

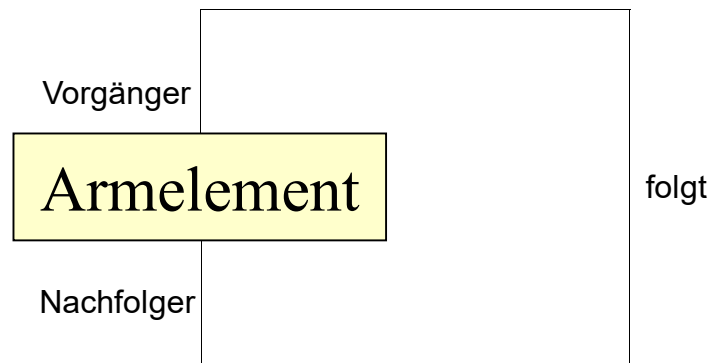
- Ein Roboterarm besteht aus n Armelementen
- Umgekehrt ist ein Armelement Teil eines Roboterarms



## Rollen

### Beispiel

- Jedes Element eines Roboterarms kann sich in einer Armelement-Sequenz befinden
- Jede Armelement-Entität spielt in der Assoziation »folgt« die Rolle des Vorgängers und die Rolle des Nachfolgers



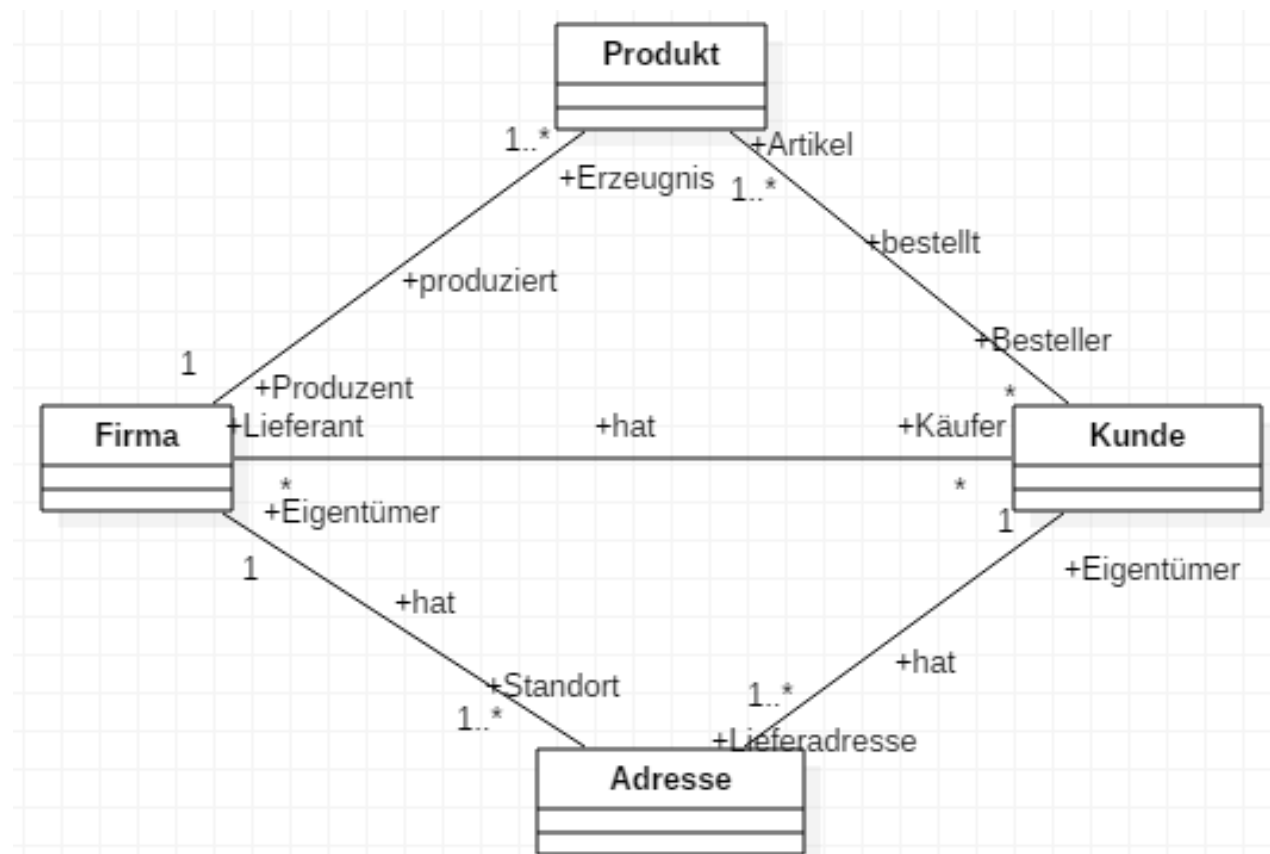


## Fragen

- **Wie sieht das Datenmodell für eine einfach verkettete Liste ganzer Zahlen aus?**
- **Wie sieht das Modell für eine doppelt verkettete Liste ganzer Zahlen aus?**
- **Wie sieht das Datenmodell für einen binären Baum aus, dessen Knoten ganze Zahlen als Wert haben?**

## Anwendungsmodell

### Beispiel



## Einfache Abbildung von Modellen auf Tabellen

- Für Datenmodelle muss man sich Speicherstrukturen überlegen, die eine Abspeicherung der entsprechenden Objekte ermöglichen
- Tabellen sind die grundlegenden Datenstrukturen von Datenbanken
- Es werden drei Regeln präsentiert, die eine Abbildung von Klassen und Assoziationen auf Tabellen ermöglichen

## Einfache Abbildung von Modellen auf Tabellen

### Regel 1

- Für jede Klasse wird eine Tabelle benötigt.
- Attribute liefern die Namen der Spalten.
- Jedes Objekt der entsprechenden Klasse stellt mit seinen Attributwerten einen Eintrag (Zeile) in dieser Tabelle dar.
- Beispiel
  - Die Klasse Firma könnte (mit den entsprechenden Attributen, Objekten und Attributwerten) folgende Tabelle ergeben:

Firma	Kurzname	Name	...	Umsatz
	Innosoft	Innovative Software GmbH	...	20.800.-
	HardSoft	Hard & Soft KG	...	33.200.-

## Einfache Abbildung von Modellen auf Tabellen

### Schlüssel

- Minimale, identifizierende Attributkombination, die unterstrichen wird.
- Oft sind auch mehrere Schlüssel möglich
  - Klasse: Stadt
  - Attribute: PLZ, Staat, Einwohnerzahl, Vorwahl
  - Schlüssel: PLZ und Staat oder Vorwahl
- In einem solchen Fall wird stets ein Schlüssel als **Primärschlüssel** ausgezeichnet
- Im Falle der Firma kann der Kurzname als Primärschlüssel genutzt werden

Firma	<u>Kurzname</u>	Name	...	Umsatz
	Innosoft	Innovative Software GmbH	...	20.800.-
	HardSoft	Hard & Soft KG	...	33.200.-

## Einfache Abbildung von Modellen auf Tabellen

### Regel 2

- Sind zwei Klassen **A** und **B** durch eine **1:1-** oder **n:1-Assoziation** verbunden, dann wird der **Schlüssel von B** als sogenannter **Fremdschlüssel** in **A** eingetragen, d.h. als zusätzliches Attribut

## Einfache Abbildung von Modellen auf Tabellen

### Beispiel für Regel 2



Kunde	<u>Personal-Nr.</u>	Funktion	Umsatz	Kurzname
	10	Berater	5.200.-	InnoSoft
	20	Systemanalytiker	10.300.-	HardSoft

Firma	<u>Kurzname</u>	Name	...	Umsatz
	InnoSoft	Innovative Software GmbH	...	20.800.-
	HardSoft	Hard & Soft KG	...	33.200.-

## Einfache Abbildung von Modellen auf Tabellen

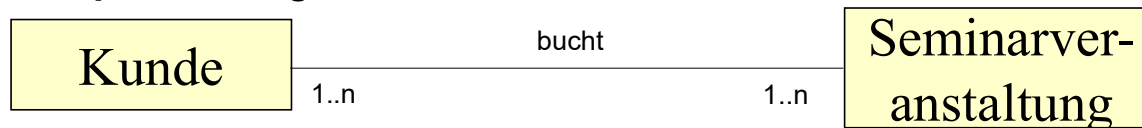
### Regel 3

- Sind zwei Klassen **A** und **B** durch eine **m:n-Assoziation** verbunden, dann wird für die Assoziation eine eigenen Tabelle angelegt.
- Als Attribute werden die **Schlüssel** der Klassen verwendet, die die Assoziation verbindet



## Einfache Abbildung von Modellen auf Tabellen

### Beispiel für Regel 3



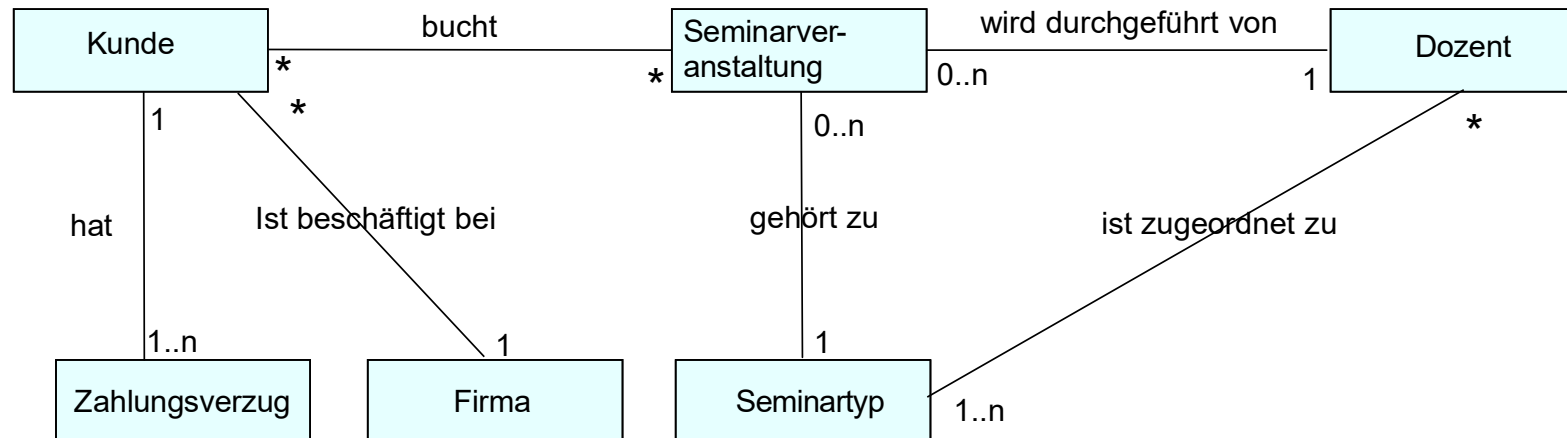
Kunde	Personal-Nr.	Funktion	Umsatz	Kurzname
	10	Berater	5.200.-	InnoSoft
	20	Systemanalytiker	10.300.-	HardSoft

Seminarveranstaltung	Veranstaltungs-Nr.	Dauer	Vom	Teiln. Aktuell
	22	3	01.03.2016	15
	94	2	04.07.2016	8
	37	1	10.10.2016	128

bucht	Personal-Nr.	Veranstaltungs-Nr.
	10	94
	20	22
	27	37

## Beispiel für ein Datenmodell

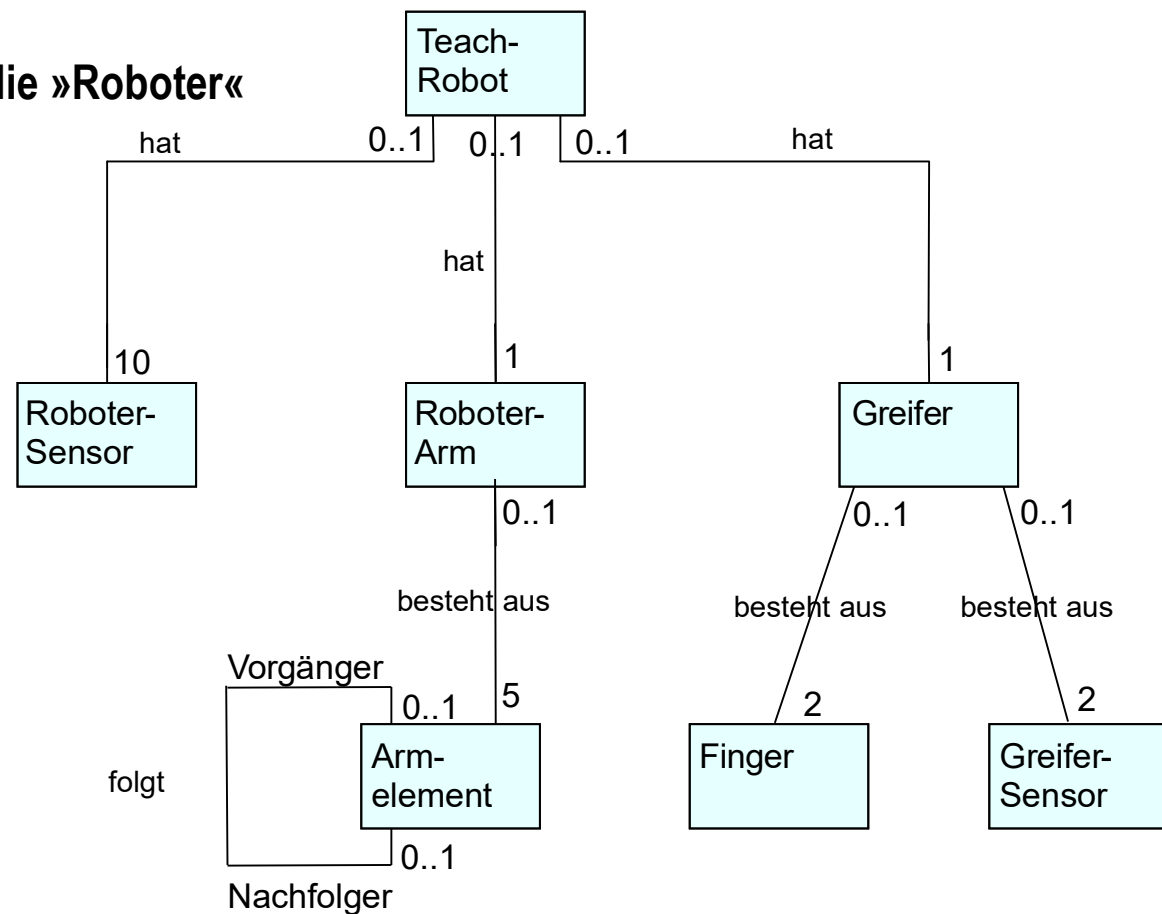
### Datenmodell der Fallstudie »Seminarorganisation«



- Man Unterscheidet zwischen Seminartyp und Seminarveranstaltung
  - In der Umgangssprache sind wir häufig nicht so exakt!

## Beispiel für ein Datenmodell

### Datenmodell der Fallstudie »Roboter«



## Qualitätssicherung

### Empfehlungen

- **Aufgabe eines Datenmodells**
  - Sind relevante Klassen dargestellt?
  - Sind Assoziationen zwischen den Klassen dargestellt?
- **Überprüfungen**
  - Besitzt jede Klasse mindestens ein Attribut?
    - Ist dies nicht der Fall, dann macht Klasse keinen Sinn
  - Sind die Klassen durch Substantive, die Assoziationen durch Verben beschrieben?
    - Ist dies nicht der Fall, dann sind die Beziehungen zu überprüfen

## Qualitätssicherung

### Empfehlungen – Überprüfung (Fortsetzung)

- Sind zwei Klassen identisch?
  - Identität kann vorliegen, wenn eine oder mehrere der folgenden Bedingungen erfüllt sind:
    - Die Klassen stehen in einer 1:1-Assoziation
    - Sie sind durch dieselben Assoziationen mit der Umgebung verbunden
    - Sie besitzen dieselben Attribute
- Jede Assoziation ist zu überprüfen auf...
  - ihre Notwendigkeit, d.h. bringt sie neue Informationen
  - korrekte Darstellung des Sachverhalts

## Qualitätssicherung

### Empfehlungen – Überprüfung (Fortsetzung)

- **Liegt eine Klasse oder ein Attribut vor?**
  - Klasse muss eindeutig identifizierbar sein und durch Attribute beschrieben werden
  - Attribut liegt vor, wenn es selbst keine weiteren Attribute besitzt
  - In der Systemanalyse hört man auf angemessenen Abstraktionsniveau auf, z.B. bei **Adresse**
    - Ein solches Attribut zunächst als elementar ansehen
    - **Farbe** ist ein Attribut vom **Auto**
    - Würde aber die Zusammensetzung von **Farbe** interessieren, so wäre Farbe eine Klasse
  - Abhängig vom Blickwinkel können Attribute zu Klassen werden und umgekehrt

## Bewertung des Datenmodellansatzes

### Positive Aspekte

- + Im kaufmännischen Anwendungsbereich ist eine Datenmodellierung ein absolutes Muss
- + Auch in vielen technischen Bereichen ist die Komplexität der Daten so groß, dass ein Datenmodell erforderlich ist (Beispiel: Roboter-Modellierung)
- + Ist Voraussetzung für einen relationalen Datenbankentwurf

## Bewertung des Datenmodellansatzes

### Negative Aspekte

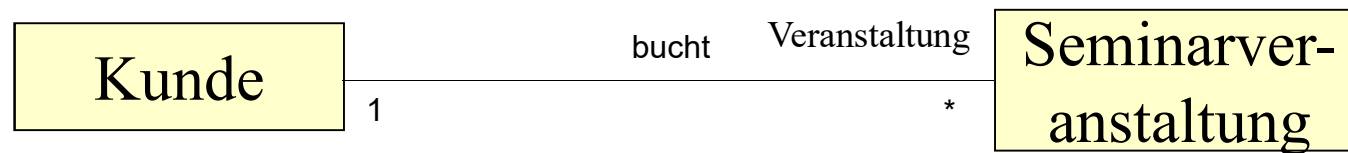
- Erfordert ein höheres Abstraktionsniveau als die bisher vorgestellten Basiskonzepte, daher schwerer zu erlernen und schwerer zu verstehen
- Datenmodelle können sehr umfangreich werden und sind dann schwer zu überblicken
- Es fehlt ein Verfeinerungsmechanismus um mehrere Abstraktionsebenen bilden zu können

\* In den folgenden Veranstaltungen werden die Datenmodelle zu objektorientierten Modellen erweitert.



## Umsetzung von Assoziationen in Java

### Einführendes Beispiel



```
public class Kunde {  
    private Vector<Seminarveranstaltung> veranstaltung  
        = new Vector<Seminarveranstaltung>();  
}
```

```
public class Seminarveranstaltung {  
    private Kunde kunde;  
}
```

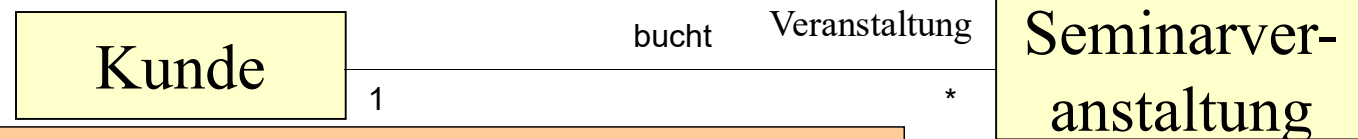
## Umsetzung von Assoziationen in Java

### Regeln

- Rollennamen werden zu Attributnamen
- Klassennamen beginnen mit Großbuchstaben, Attributnamen mit Kleinbuchstaben
- Der Zugriff auf ein Attribut `name` sollte nicht direkt erfolgen, sondern durch die Methoden `setName` und `getName`
- Bei Vektoren müssen Methoden zum Hinzufügen und Entfernen von Elementen existieren.

## Umsetzung von Assoziationen in Java

### Einführendes erweitertes Beispiel



```

public class Kunde {
    private Vector<Seminarveranstaltung> veranstaltung
        = new Vector<Seminarveranstaltung>();

    public void hinzufügenVeranstaltung (Seminarveranstaltung r) {
        veranstaltung.add(r);    }
    public void entfernenVeranstaltung (Seminarveranstaltung r) {
        veranstaltung.remove(r);
    }
}
    
```

```

public class Seminarveranstaltung {
    private Kunde kunde;

    public void setKunde (Kunde k) {
        kunde = k;    }
    public Kunde getKunde () {
        return kunde;    }
}
    
```

## Gerichtete Assoziationen

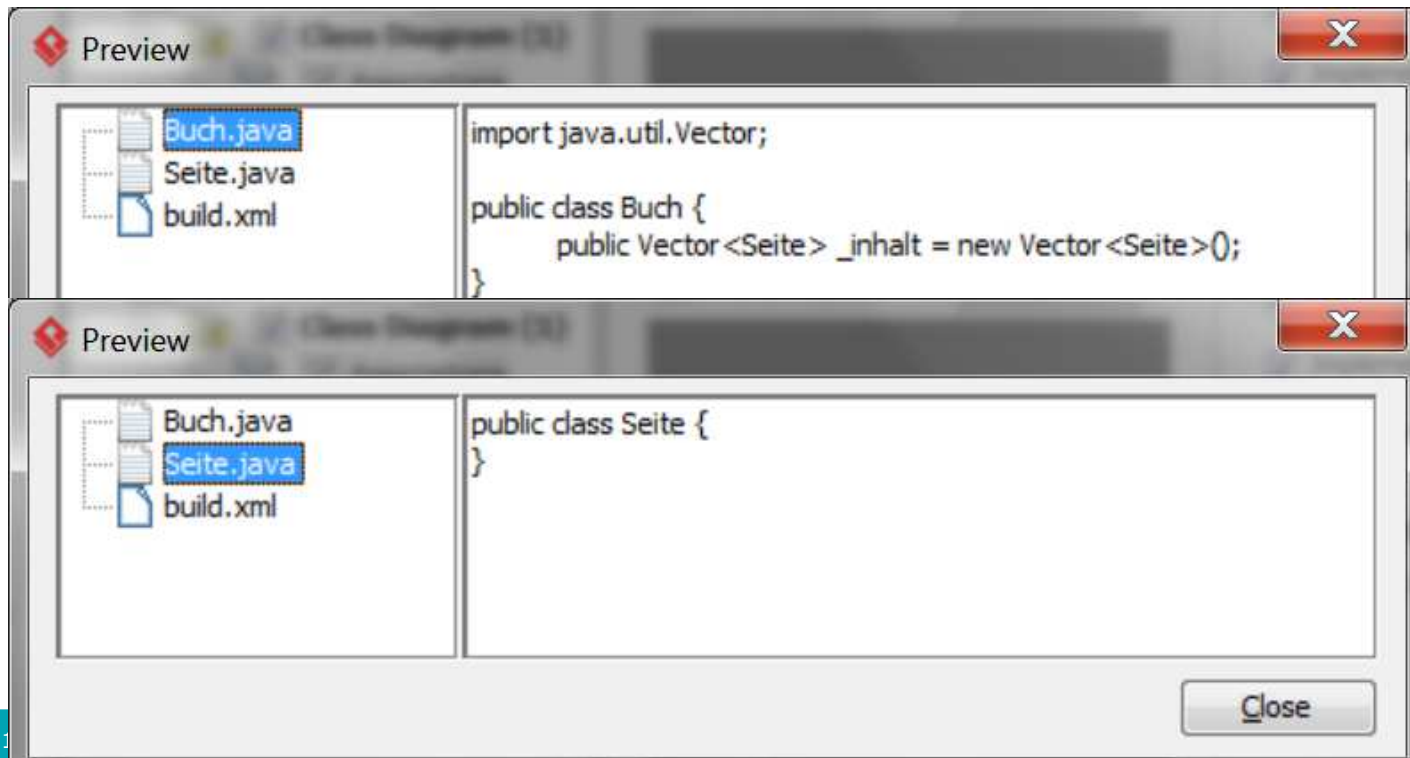
### Beispiel



Ein Buch kennt seine Seiten. Die Seiten kennen aber nicht das Buch, zu dem sie gehören.

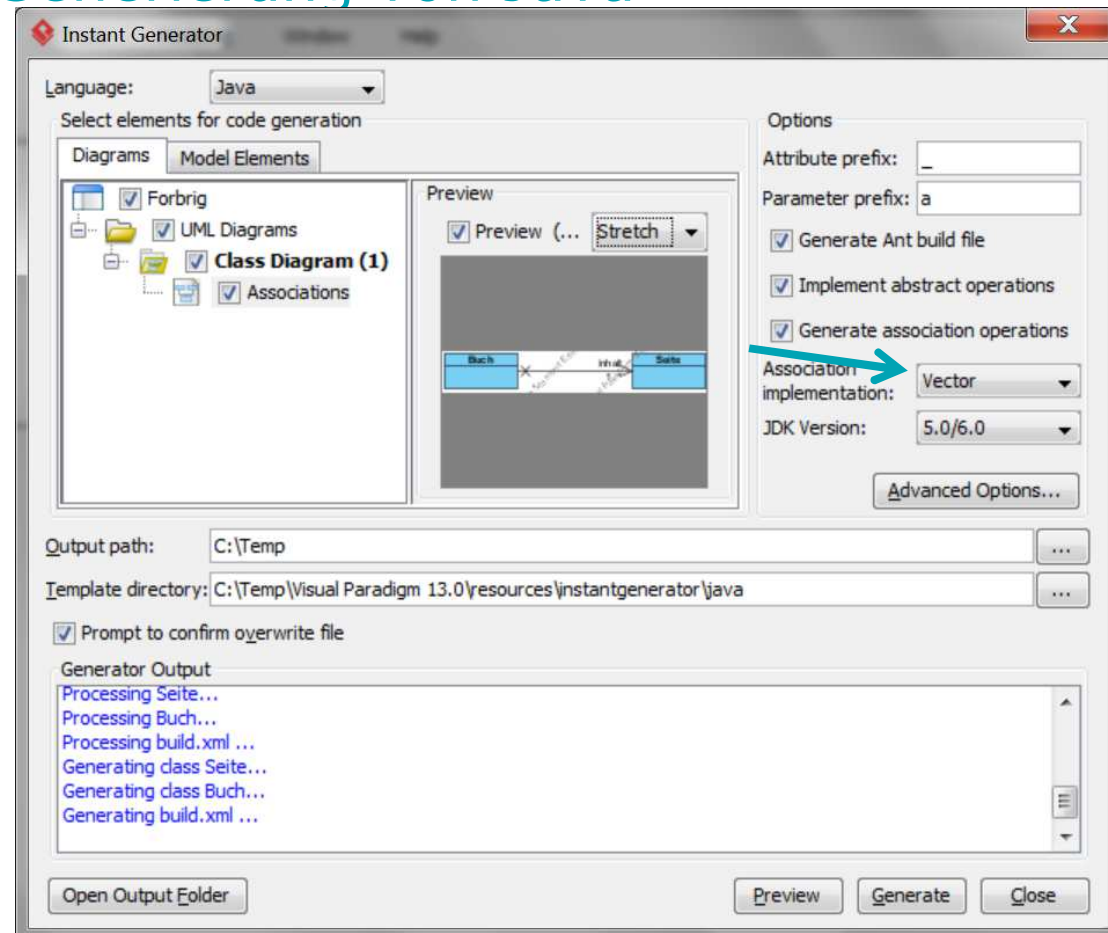
## Gerichtete Assoziation

Umsetzung in Java



## Anpassung der Generierung von Java

### Beispiel Visual Paradigm



## Mögliche Navigationsspezifikationen

a) genau einseitige Navigation



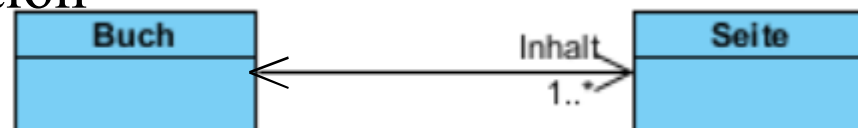
b) mindestens einseitige Navigation



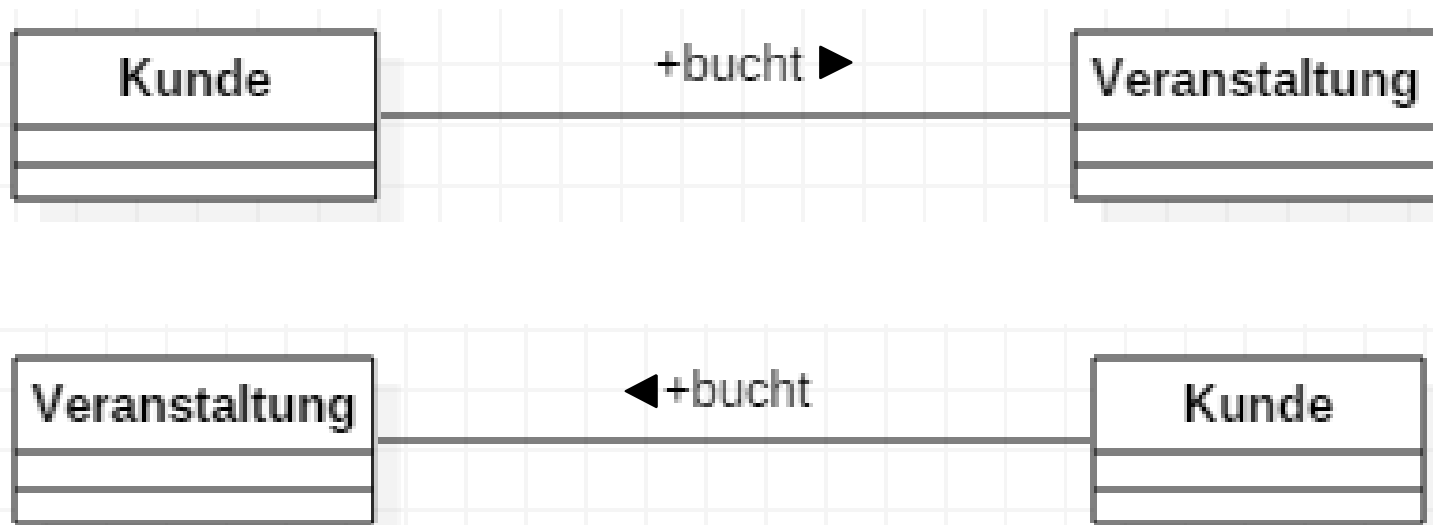
c) zweiseitige Navigation oder  
unspezifiziert



d) zweiseitige Navigation



## Assoziation und Leserichtung





## Modellierung

### Modelle

#### **Deskriptive Modelle:**

Abbilder von etwas

#### **Präskriptive Modelle:**

Vorbilder für etwas

### Modellmerkmale

- **Abbildungsmerkmal:**

Zum Modell gibt es das Original, ein Gegenstück, das wirklich vorhanden, geplant oder fiktiv sein kann.

- **Verkürzungsmerkmal:**

Ein Modell erfaßt nicht alle Attribute des Originals, sondern nur einen Ausschnitt, der vor allem durch den Zweck des Modells bestimmt ist.

- **Pragmatisches Merkmal:**

Modelle können unter bestimmten Bedingungen und bezüglich bestimmter Fragestellungen das Original ersetzen.

## Modellierung

33

### **Tätigkeiten in der Modellbildung** (M.Glinz)

- **Reflektieren**: überlegen und verstehen, was modelliert werden soll (Pragmatik des Modells, abzubildende/ wegzulassende Attribute, Umfang,...)
- **Gewinnen**: Informationen über das Original und die Intentionen der Wissensträger gewinnen (diskutieren, lesen, fragen, rückfragen, suchen, analysieren,...)
- **Beschreiben**: gewonnene Informationen verstehen, ordnen, strukturieren, bewerten,... und mit geeigneten Mitteln beschreiben
- **Validieren**: Modelle (auch Zwischenergebnisse) durch Wissensträger überprüfen lassen: Ist es das, was sie wollen und brauchen?

## Modellierung

### Modelle in der Softwaretechnik

- **Wir arbeiten fast nur mit Modellen -> mehrstufige Modelle:**
  - Anforderungen = Modell der Software-Spezifikation
  - Software-Spezifikation = Modell des Codes
  - Code = Modell des ausführbaren Programms
  - ausführbares Programm = Modell der Ausführung...
- **Unterscheidung in**
  - Software-Modelle
  - Vorgehens- und Prozessmodelle
- **Modelle oft durch Graphen dargestellt**
  - meist gerichtete Graphen, oft Bäume
  - reiche Beschriftung von Kanten und Knoten
  - Darstellung ist wichtig ("Die Darstellung ist der Graph.")
  - Konnotationen von Darstellungen sind auch gefährlich!

## Zusammenfassung

- Datenmodelle sind eine wichtige Form der Modellierung von Domänen
  - Regeln zur Notation
- Sie sind ein Ausgangspunkt für die Softwareentwicklung und der Ausgangspunkt für die Datenbanken
  - Regeln zur Erzeugung von Tabellen
  - Regeln zur Erzeugung von Quelltext
- In der nächsten Veranstaltung steht die Erweiterung von Datenmodellen zu objektorientierten Spezifikationen im Mittelpunkt.