

Vorlesung: Datenorganisation SS 2005

Dipl.-Ök. Simon König
koenig@iwi.uni-hannover.de
Raum 427

Vorlesung: Datenorganisation SS 2004

Konzeptionelle Datenmodellierung

koenig@iwi.uni-hannover.de

- 4 Konzeptionelle Datenmodellierung**
 - 4.1 Statischer Datenbankentwurf mit dem Entity-Relationship-Modell
 - 4.2 Beispiele mit dem E-R-Modell
 - 4.3 Strukturiertes Entity-Relationship-Modell S-E-R-M**
 - 4.4 Vergleich E-R-Modell / S-E-R-Modell
 - 4.5 Konzeptionelle Modellierung mit UML

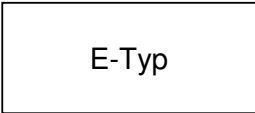

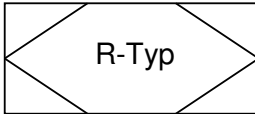

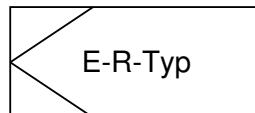




Strukturiertes Entity-Relationship-Modell





- arbeitet Struktureigenschaften heraus, die im E-R-M implizit enthalten sind
- Durch die Zusammenziehung eines E-Typs und eines R-Typs entsteht der neue ER-Typ
- Identifizierung und Ordnung von originären und abhängigen Objekttypen auf Basis der Existenzabhängigkeiten zwischen den Objekttypen
- Grundkonstrukte können bei der Gestaltung des Diagramms nicht frei verwendet werden, sondern sind entsprechend des Strukturprinzips anzuordnen
- Eine Abart des S-E-R-Ms wird von SAP zur Dokumentation der Datenstruktur ihres R/3-Systems verwendet



Konstruktionsoperatoren S-E-R-Modell

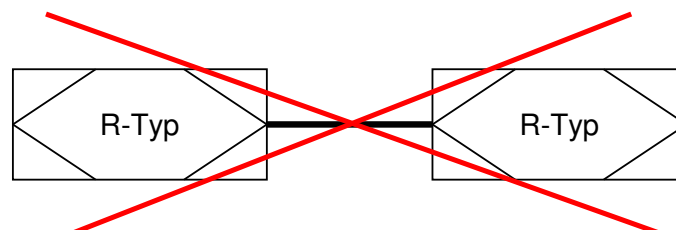
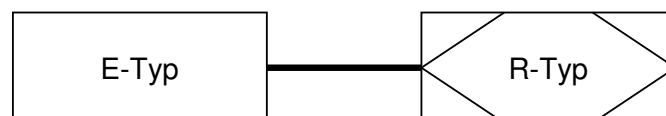
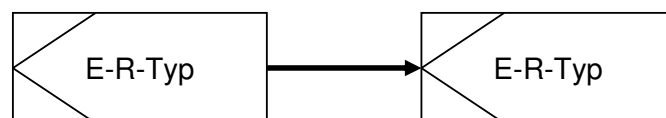
Objekttypen	Beziehungen	Existenzabhängigkeit
	$(0,1)$ 	einseitig
	$(0,*)$ 	einseitig
	$(1,*)$ 	wechselseitig
	$(1,1)$ 	wechselseitig

Kardinalitäten im S-E-R-Modell

		Minimum	
		0 einfache Linie	1 doppelte Linie
Maximum	1 ohne Pfeilspitze		 Sonderfall
	* mit Pfeilspitze		
		Einseitige Existenzabhängigkeit	wechselseitige Existenzabhängigkeit

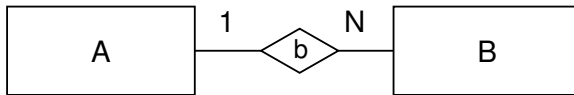
Darstellungsregeln für S-E-R-Diagramme

- immer von E (Rechteck) zu R (Raute)
- Kanten werden von links nach rechts dargestellt



Beziehung / Existenzabhängigkeit

E-R-Modell

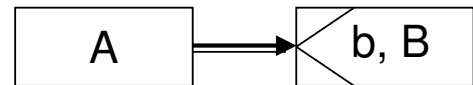
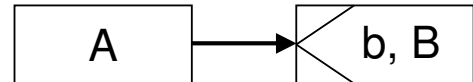


a) 0,* 1,1

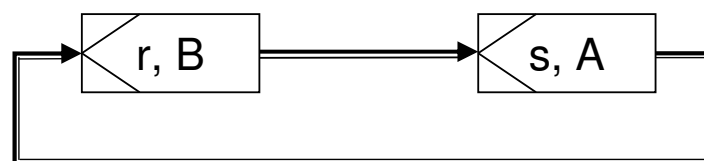
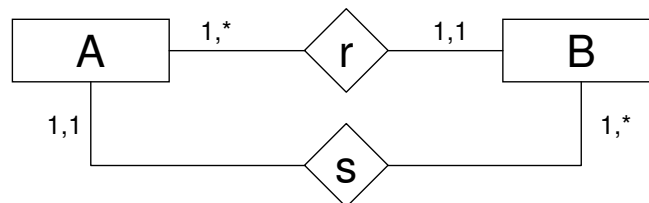
b) 1,* 1,1

c) 0,* 0,1

S-E-R-Modell



Vermeidung zyklischer Existenzabhängigkeiten



Verletzung der Darstellungsregeln!



4 Konzeptionelle Datenmodellierung

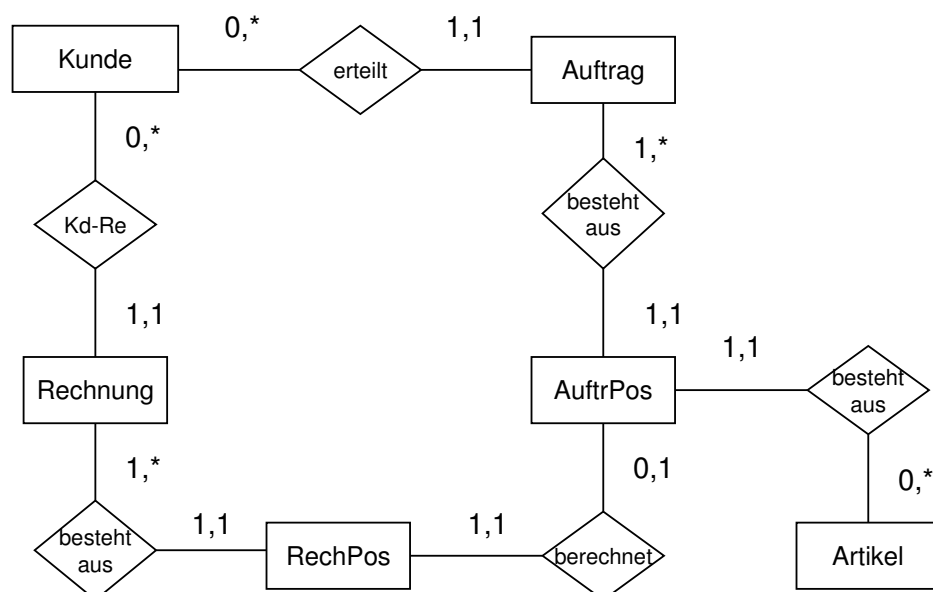
4.1 Statischer Datenbankentwurf mit dem Entity-Relationship-Modell

4.2 Beispiele mit dem E-R-Modell

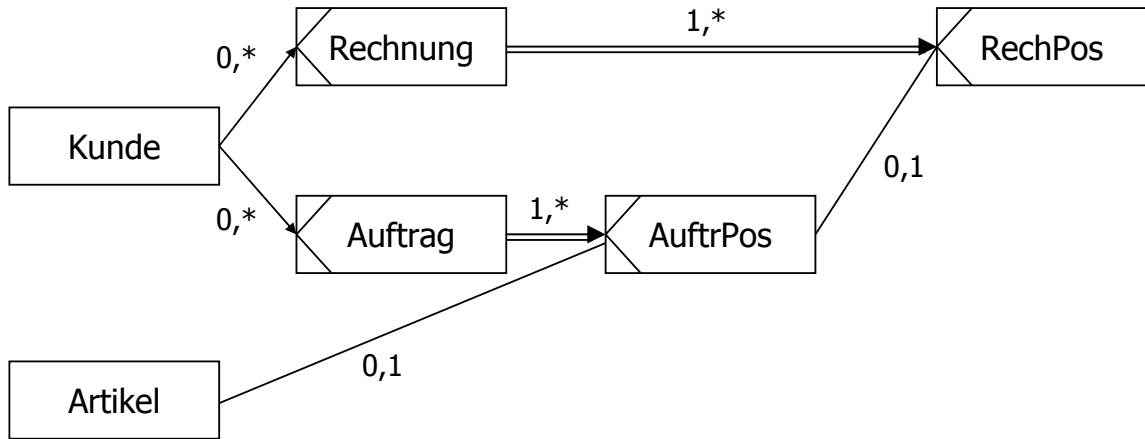
4.3 Strukturiertes Entity-Relationship-Modell S-E-R-M

4.4 Vergleich E-R-Modell / S-E-R-Modell

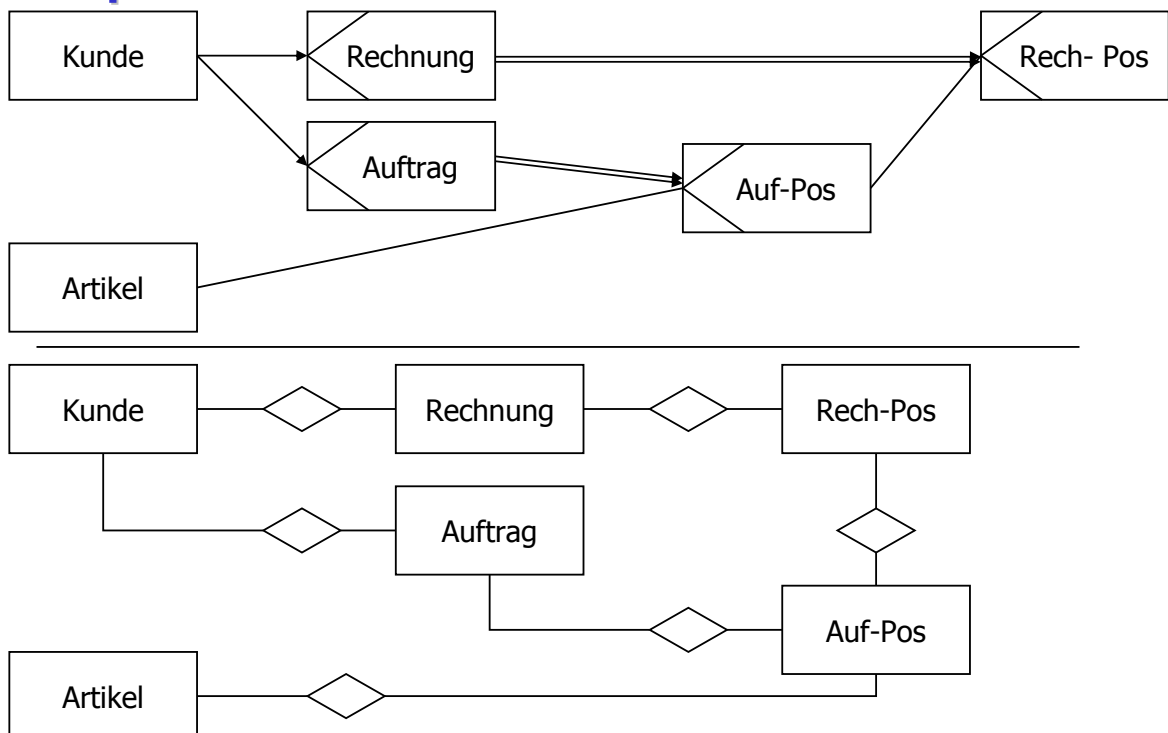
4.5 Konzeptionelle Modellierung mit UML



Beispiel eines S-E-R-Modells



Beispiel S-E-R- vs. E-R-Modell





4 Konzeptionelle Datenmodellierung

4.1 Statischer Datenbankentwurf mit dem Entity-Relationship-Modell

4.2 Beispiele mit dem E-R-Modell

4.3 Strukturiertes Entity-Relationship-Modell S-E-R-M

4.4 Vergleich E-R-Modell / S-E-R-Modell

4.5 Konzeptionelle Modellierung mit UML



Was ist die UML?

- **U**nified **M**odelling **L**anguage
- **Sprache** zur Spezifikation, Visualisierung, Konstruktion und Dokumentation
- von **Modellen** für Softwaresysteme, Geschäftsmodelle und andere Nicht-Softwaresysteme.
- UML ist seit 1998 **de-facto Standard** und liegt der **O**bjekt **M**anagement **G**roup (OMG) zur Standardisierung vor.



Was ist die UML?

- **Kombination** der besten Ideen objekt-orientierter Entwicklungsmethoden
 - Entwickelt von Booch, Jacobsen und Rumbaugh (Rational Software, heute IBM)
 - Unterstützung führender Unternehmen der Software-Branche (Microsoft, Oracle, Hewlett-Packard,...)
- **Keine Methode**, sondern Satz von Notationen zur Formung einer allgemeinen Sprache zur Softwareentwicklung.



Elemente von UML

- **Untermodule** für Entwurf von Softwaresystemen auf **unterschiedlichen Abstraktionsebenen**
 - Statische Struktur (**Klassenstruktur**)
 - Zusammenspiel von Objekten bzw. deren Operationen (**Sequenz- bzw. Interaktionsdiagramme**)
 - **Anwendungsfälle** (use cases)
 - Operationen, Benutzeraktivitäten (**Aktivitäts- und Zustandsübergangsdigramme**)
- Strukturelle Modellierung bestehend aus **Objektklassen** und **Assoziationen zwischen Klassen** für Datenbankentwurf am wichtigsten.



Begriffliche Grundlagen

Klasse

- Definition der Struktur eines Objektes.
- Enthält die Beschreibungen der Datenstruktur und der Methoden.

Datenelemente

- sind die zu einer Klasse gehörenden Daten,
- können einfache und komplexe Datentypen sein.

Methoden

- sind die zu einer Klasse gehörenden Methoden,
- enthalten Beschreibungen der Anweisungen, die bei den jeweiligen Operationen gegen ein Objekt ablaufen.



Begriffliche Grundlagen

Objekte bzw. Instanzen

- sind die Realisierungen der Klassen,
- z. B. Klasse „Mensch“ und Objekt „Peter“.

Nachricht

- regelt den Meldungsverkehr zwischen Objekten.
- Aufforderung eines Objektes an ein anderes, eine bestimmte Methode auszuführen.
- Ereignisse werden durch Nachrichten modelliert.
- Die Menge aller öffentlichen Nachrichten nennt man Protokoll.



Beispiel einer Klasse

Klassenname:	Kunde
Attribute:	- Kundennummer - Name - Vorname - Straße - Postleitzahl - Wohnort - Telefon
Operationen, Methoden:	+ Erstellen + Ändern + Löschen + Suchen

In UML fehlt das Konzept eines Schlüssels, da Objekte immer einen systemweit eindeutigen **Objektidentifikator** zugeordnet bekommen!



Surrogate

Forderungen:

- Eindeutigkeit
- Unveränderbarkeit
- Strukturunabhängigkeit
- Ortsunabhängigkeit



Beispiel einer Methode

```
Auftrag :: Erstellen
{
    Suche nächste Auftragsnummer;

    Abfrage Kundennummer:
    Nachricht an Kunde: Suche;
    Wenn nicht gefunden,
        Nachricht an Kunde: Erstelle;

    Wähle Auftragspositionen
    /* Beginn einer Schleife */
    {
        Abfrage Artikelnummer:
        Nachricht an Artikel: Suche;
        Abfrage Stückzahl;
    }
    /* Ende der Schleife */

    Ermittle Auftragswert
}
```



Assoziationen

Beziehungen werden **als Assoziationen** ausgedrückt.
Zu jeder Assoziation von K1 zu K2 existiert eine umgekehrte Assoziation von K2 zu K1.

- **Einfache Assoziationen (Typ 1)**
(Jedem Objekte in K1 ist genau ein Objekt in K2 zugeordnet)
- **Konditionell einfache Assoziation (Typ 0,1)**
(Jedem Objekt in K1 ist höchstens eine Objekt aus K2 zugeordnet)
- **Mehrfache Assoziationen (Typ 1,*)**
- **Konditionell mehrfache Assoziationen (Typ 0,*)**



Klassifizierung von Assoziationen

- einfach-einfache Beziehung



- Ein Musikfestspiel verpflichtet **höchstens ein** Festspielorchester
- Ein Festspielorchester spielt exklusiv für **genau ein** Musikfestspiel



Klassifizierung von Assoziationen

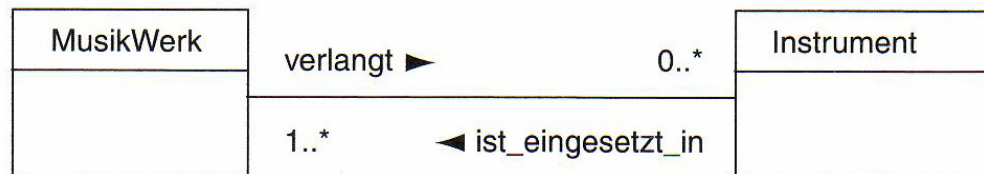
- komplex-einfache Beziehung



- Ein Musikhaus hat **mehrere** Musikabteilungen
- Eine bestimmte Musikabteilung gehört zu **genau einem** Musikhaus

Klassifizierung von Assoziationen

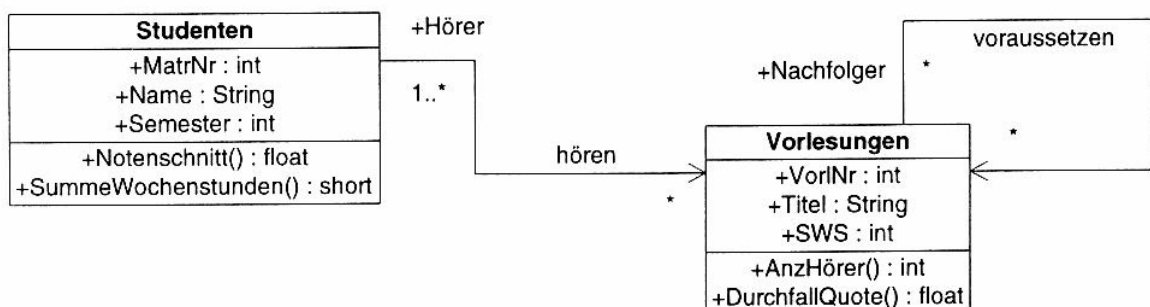
- komplex-komplex Beziehung



- Ein Musikwerk verlangt **kein, ein oder mehrere** Instrumente
- Ein Instrument ist eingesetzt in **mehreren** Musikwerken



Assoziationen zwischen Klassen

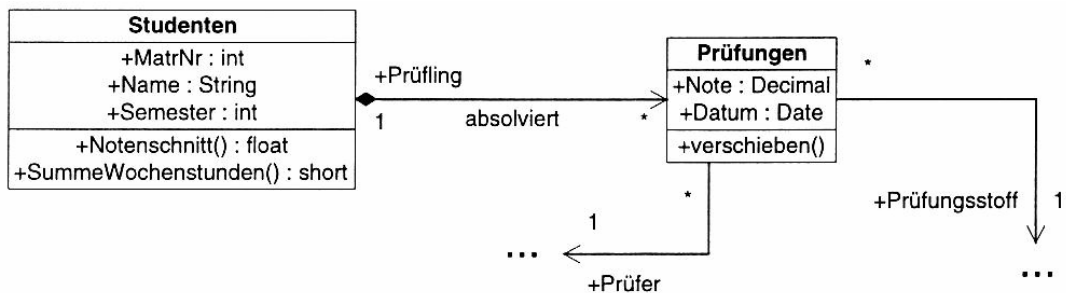
- rekursive Assoziationen sind möglich
- anders als im ERM kann man Assoziationen in UML eine Richtung zuordnen



Aggregation in UML

- Aggregation als besondere Form von Beziehungen / Assoziationen
- **Unterscheidung von zwei Arten**

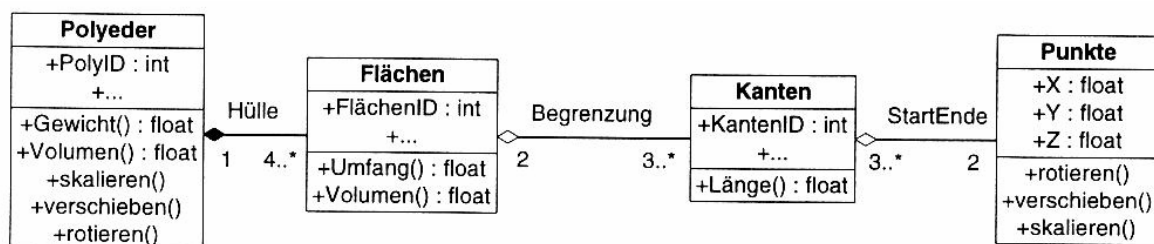
- exklusive Zuordnung (Komposition)  —
- „normale“ Teil / Ganzes-Beziehung  —



Folie 29
12.05.2005

Kemper, A; Eickler, A.: Datenbanksysteme: Eine Einführung. 4. Aufl. München 2001, S. 60.

Anwendungsbeispiel: Begrenzungsflächendarstellung von Polyedern

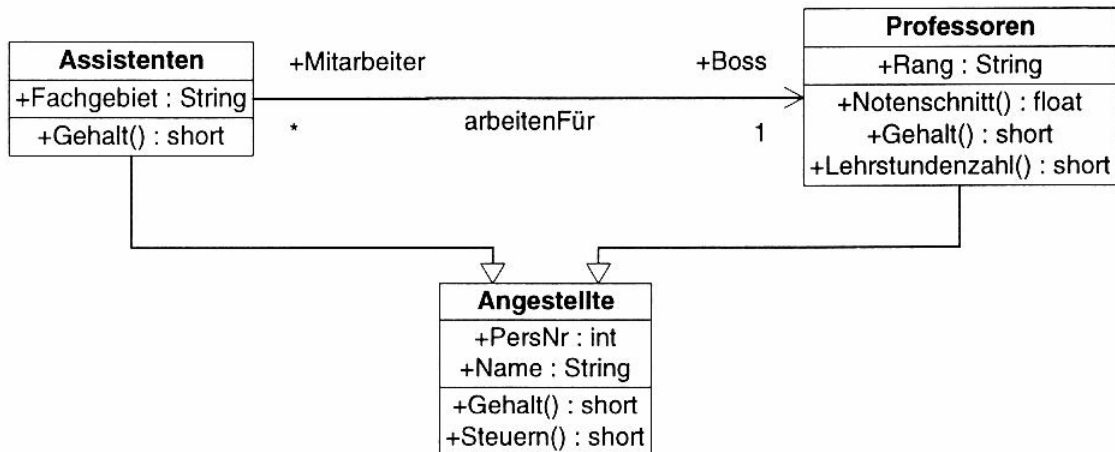


Folie 30
12.05.2005

Kemper, A; Eickler, A.: Datenbanksysteme: Eine Einführung. 4. Aufl. München 2001, S. 61.

Generalisierung in UML-Notation

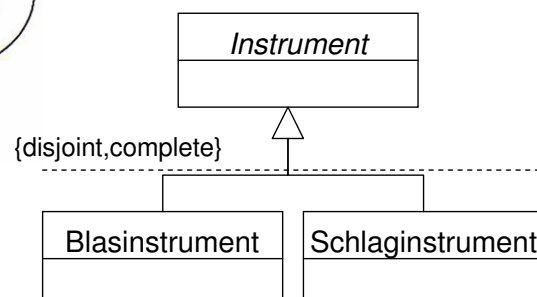
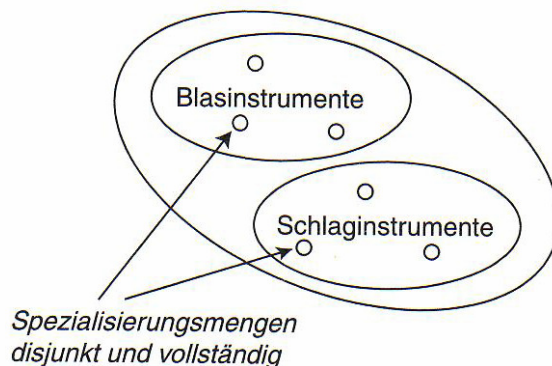
- Generalisierung bzw. Spezialisierung als **Abstraktionsmittel** in Zusammenhang mit **Vererbung**



Folie 31
12.05.2005

Kemper, A; Eickler, A.: Datenbanksysteme: Eine Einführung. 4. Aufl. München 2001, S. 62.

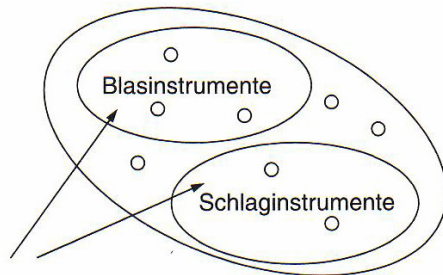
Vererbungsstrukturen (I)



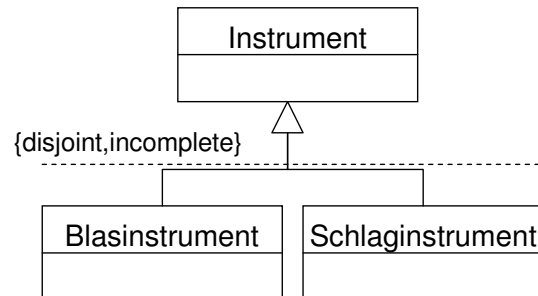
Folie 33
12.05.2005

In Anlehnung an Meier, A.; Wüst, T.: Objektorientierte und objektorientale Datenbanken. 2. Aufl. Heidelberg 2000, S. 30.

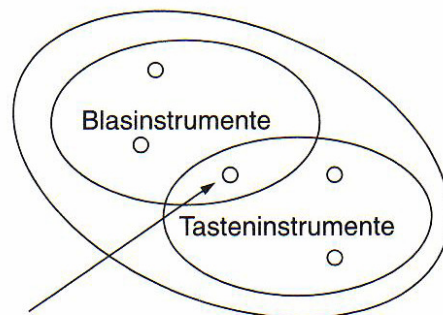
Vererbungsstrukturen (II)



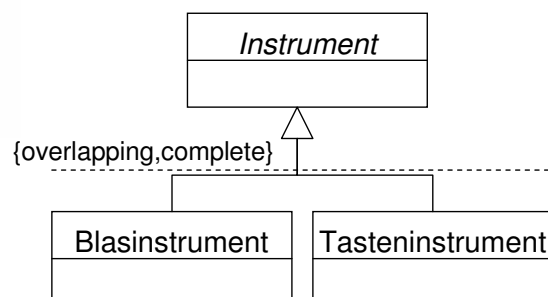
*Spezialisierungsmengen
disjunkt, aber unvollständig*



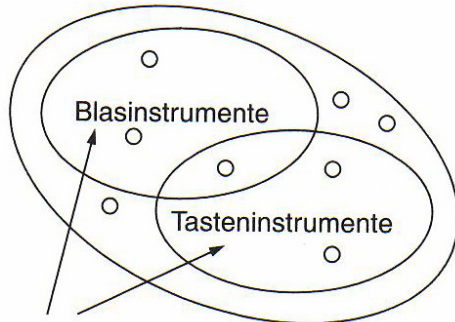
Vererbungsstrukturen (III)



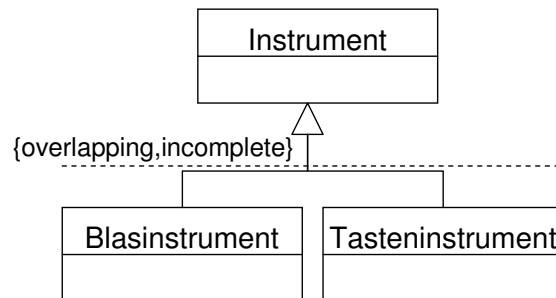
*Spezialisierungsmengen
überlappend und vollständig*



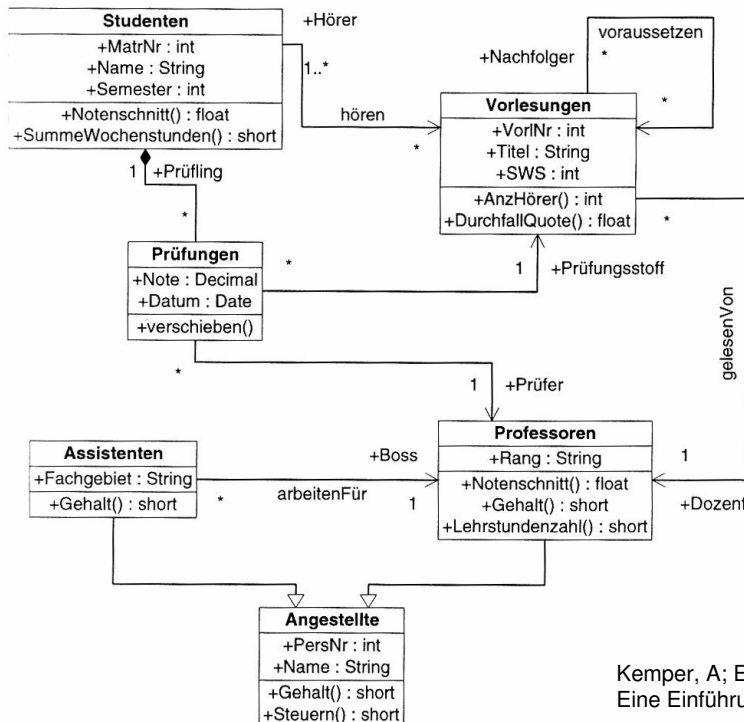
Vererbungsstrukturen (IV)



Spezialisierungsmengen überlappend und unvollständig



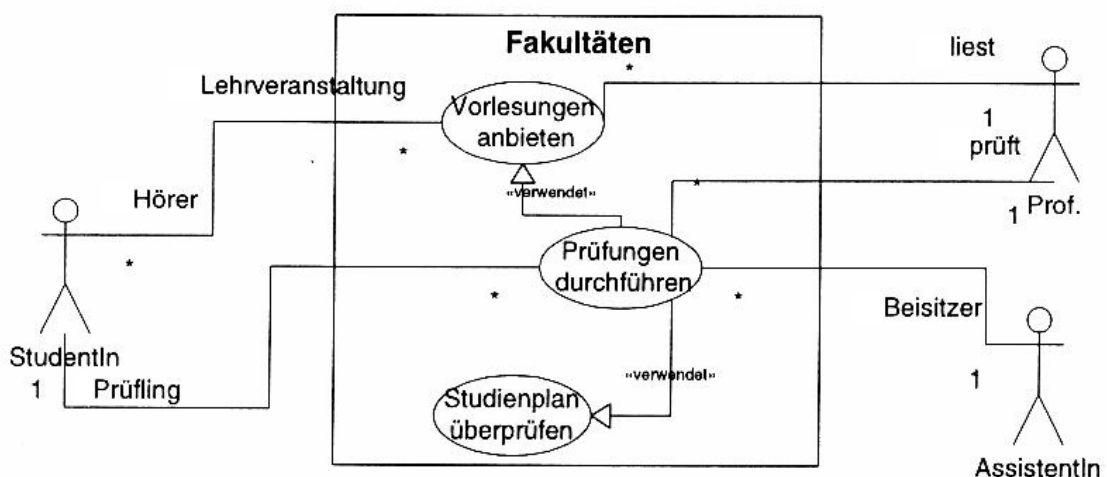
Modellierung der Universität in UML



Use-cases in UML

- **Grafische Modellierung** von Anwendungsfällen (use cases) innerhalb der Anforderungsanalyse.
- **Abstrakte Identifizierung wesentlicher Komponenten** des zu erstellenden Informationssystems.
- **Erleichtert die Kommunikation** mit späteren Anwendern (Kunden).
- Augenmerk liegt auf der **Identifikation von Akteuren**.

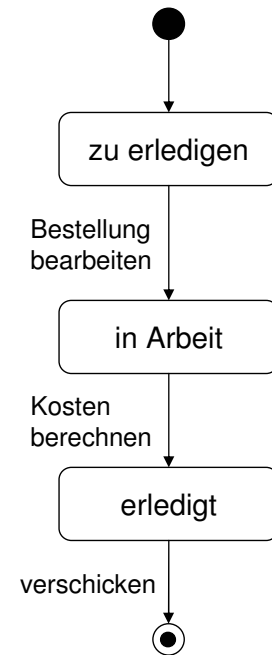
Beispiel: Anwendungsfall-Modellierung





Zustandsübergangsdiagramme

- Knoten repräsentieren Zustände
- Kanten stellen Zustandsänderungen dar
- Verhaltensmodellierung einzelner Objekte, eines Anwendungsbereichs oder des gesamten Systems

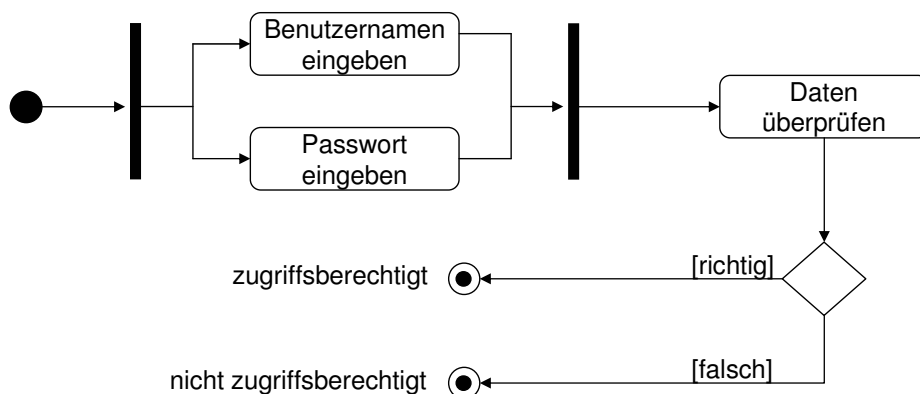


In Anlehnung an Meier, A.; Wüst, T.: Objekt-orientierte und objektrationale Datenbanken. 2. Aufl. Heidelberg 2000, S. 32.



Aktivitätsdiagramme

- Modellierung der Kontrollflüsse eines Systems



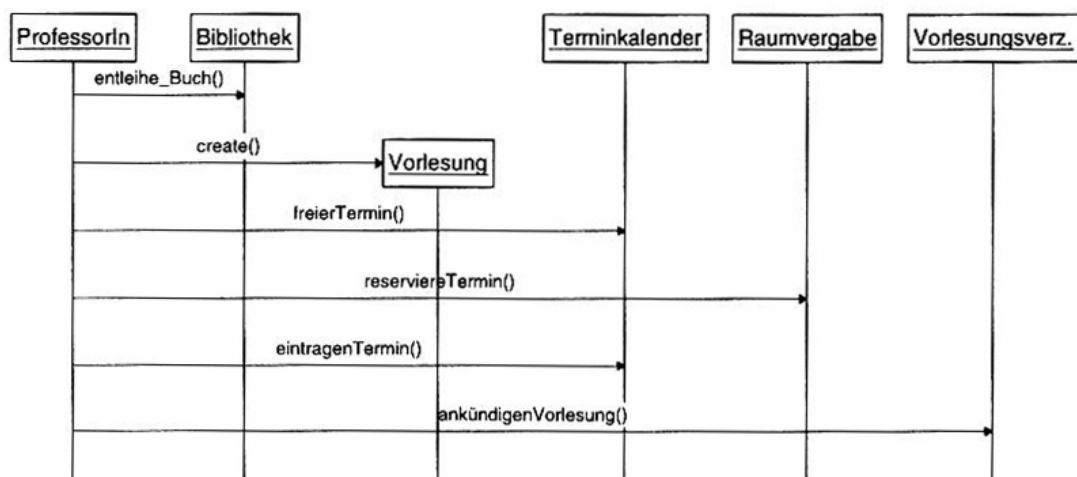


Interaktionsdiagramm

- gibt für jeden Prozess oder Teilprozess die **involvierten Objekte** mit **dazugehörigen Operationen** an,
- verdeutlicht das **Zusammenwirken von Objekten**.
- Einzelne Operationen (Nachrichten) werden durch **Pfeile** markiert.
- **Objektvorkommen** wird durch Unterstreichen des Klassennamens kenntlich gemacht.
- Starke Betonung der **Ablaufreihenfolge**:
 - Objekte horizontal nebeneinander
 - Operationen vertikal von oben nach unten



Interaktionsdiagramm zur Planung einer Vorlesung



Komponenten eines Objektmodells

