

Software-Engineering

Sebastian Iwanowski
FH Wedel

Kapitel 4: Systemanalyse
Teil 2: Datenorientierte Sicht

Systemanalyse: Datenorientierte Sicht

Entity-Relationship-Modellierung (ERM):

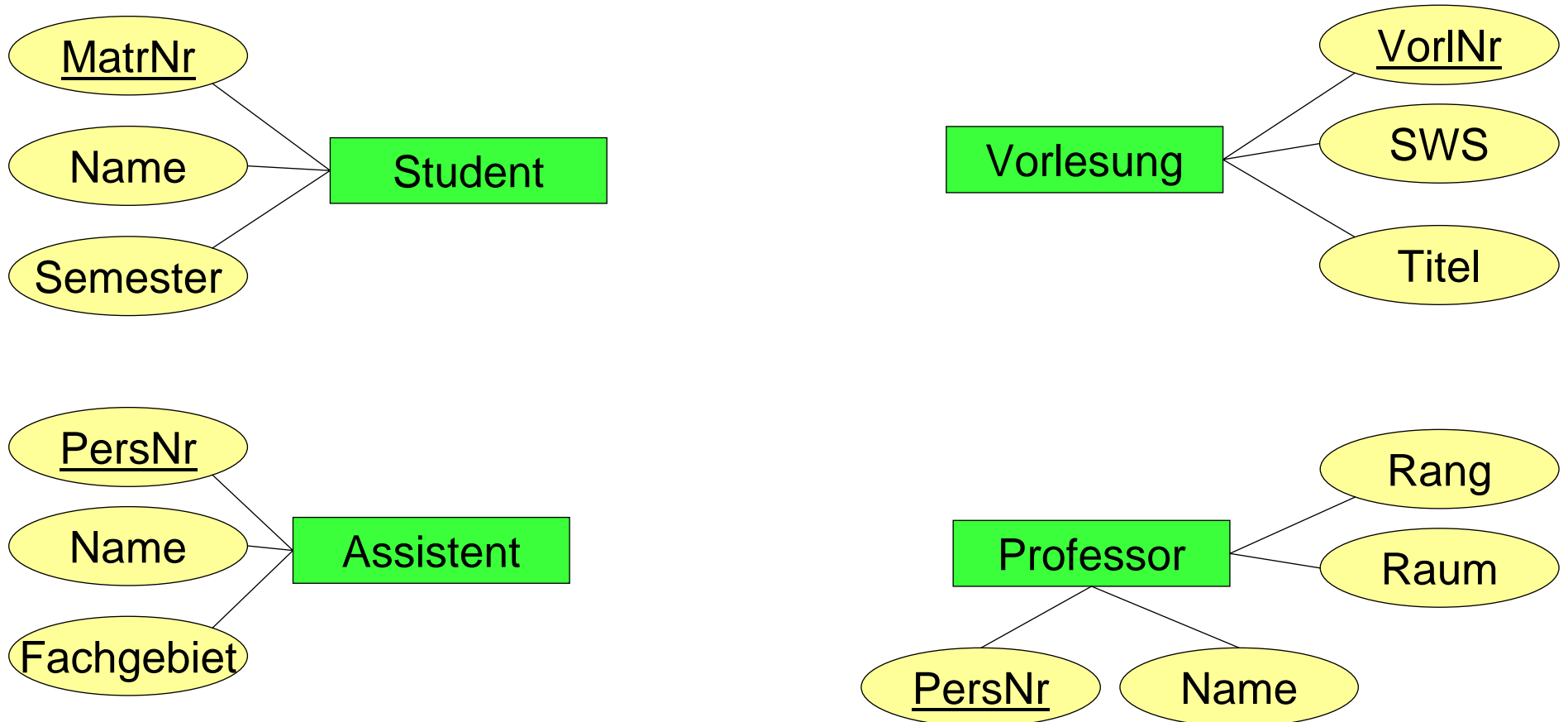
(Chen 1976)

Entitäten

- Entitäten sind die zu unterscheidenden Einheiten des Systemmodells.
- Entitäten werden durch Attribute beschrieben.
- Die Attribute haben bestimmte Wertebereiche.
- Ein Entitätstyp fasst Entitäten mit gleichen Attributen zu einer Menge von gleichartigen Entitäten zusammen.
- Innerhalb eines Entitätstyps ist jede Entität durch Schlüsselattribute (häufig nur ein Schlüsselattribut) eindeutig bestimmt.

Entity-Relationship-Modell

Universitätsbeispiel: Entitätstypen



Zwischen den *Entitäten* bestehen *Beziehungen*

Entity-Relationship-Modell

Beziehungen (Relationships) zwischen Entitäten:

- Entitäten desselben Typs haben gleichartige Beziehungen

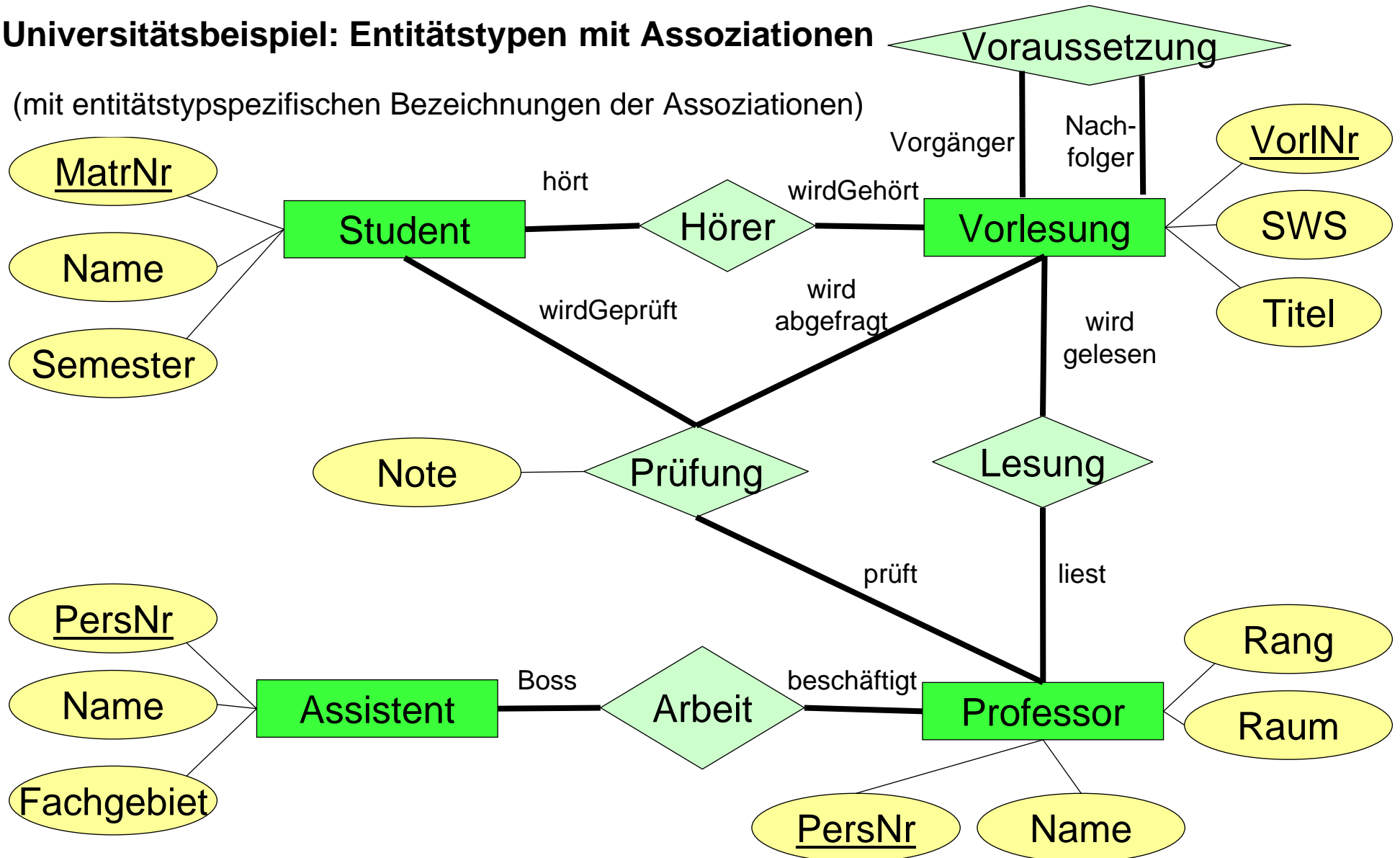
Assoziationen zwischen Entitätstypen:

- Assoziationen verbinden zwei oder mehr Entitätstypen
- Assoziationen sind nicht gerichtet.
- Assoziationen haben eindeutige Namen.
- Assoziationen können Attribute haben.
- Assoziationen können von Entitätstypen aus spezifisch bezeichnet werden.
- **Assoziationen legen die Art der Beziehung zwischen den zu den Typen zugehörigen Entitäten fest.**

Entity-Relationship-Modell

Universitätsbeispiel: Entitätstypen mit Assoziationen

(mit entitätstypspezifischen Bezeichnungen der Assoziationen)



Entity-Relationship-Modell

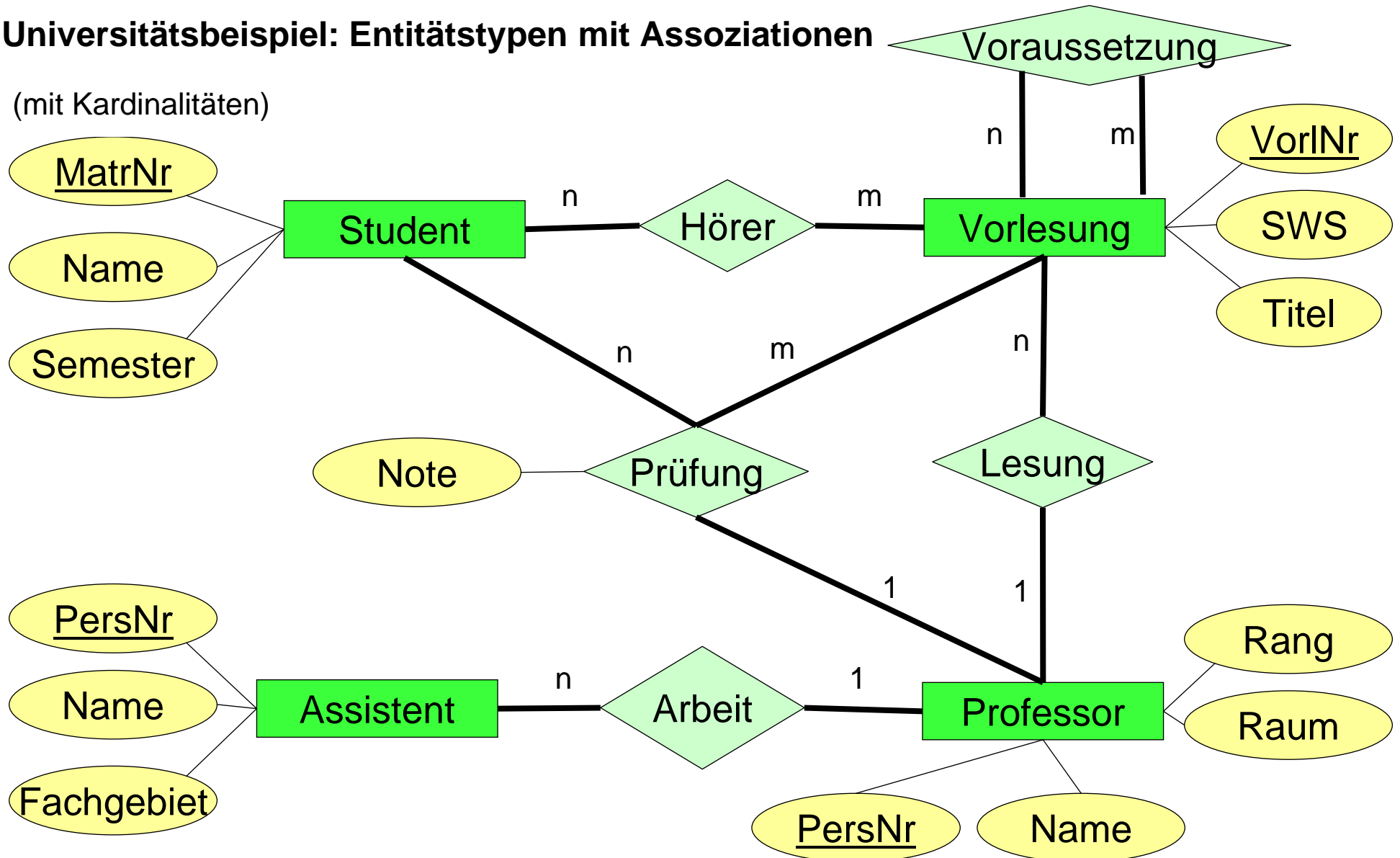
Kardinalitäten für Assoziationen

- Kardinalitäten geben für jeden an einer Assoziation beteiligten Entitätstyp an, wieviele Entitäten dieses Typs an der Assoziation beteiligt sein dürfen (müssen), wenn die Entitäten auf der anderen Seite der Assoziation fest gewählt sind.
- Kardinalität 1 bedeutet: genau eine
- Kardinalität 3 bedeutet: genau 3
- Kardinalität n bedeutet: mehrere (mindestens eine)
- Kardinalität m bedeutet: mehrere (mindestens eine), aber nicht notwendig genauso viele wie auf der anderen Seite (wenn dort n steht)
- Kardinalität 2..5 bedeutet: zwischen 2 und 5
- Kardinalität 0..1 bedeutet: eine oder gar keine

Entity-Relationship-Modell

Universitätsbeispiel: Entitätstypen mit Assoziationen

(mit Kardinalitäten)



Entity-Relationship-Modell

Wichtiges Anwendungsgebiet für ERM:

Modellierung für Datenbanken

Ein ER-Modell kann also auch für den Softwareentwurf eingesetzt werden !

Methode zur Umwandlung eines ER-Modells in ein relationales Datenbankmodell (unter Beachtung der Normalformen):

- Aus den Entitätstypen werden Tabellen
(mit den Attributen und n:1-Assoziationen als Spalten und den Entitäten als Zeilen)
- Aus den m:n-Assoziationen werden Tabellen
(mit den Attributen und Entitätstypen als Spalten und den konkreten Beziehungen zwischen den Entitäten als Zeilen)

Achtung: ER-Modell \neq Relationales Modell

Relationales Modell

Universitätsbeispiel: Tabellen *in Normalform*

Professor			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Student		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Vorlesung			
VorINr	Titel	SWS	wirdGelesen
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

Voraussetzung	
Vorgänger	Nachfolger
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

Hörer	
MatrNr	VorINr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

Assistent			
PersINr	Name	Fachgebiet	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2126

Prüfung			
MatrNr	VorINr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Systemanalyse: Datenorientierte Sicht

Was fehlt bei ER-Modellen ?

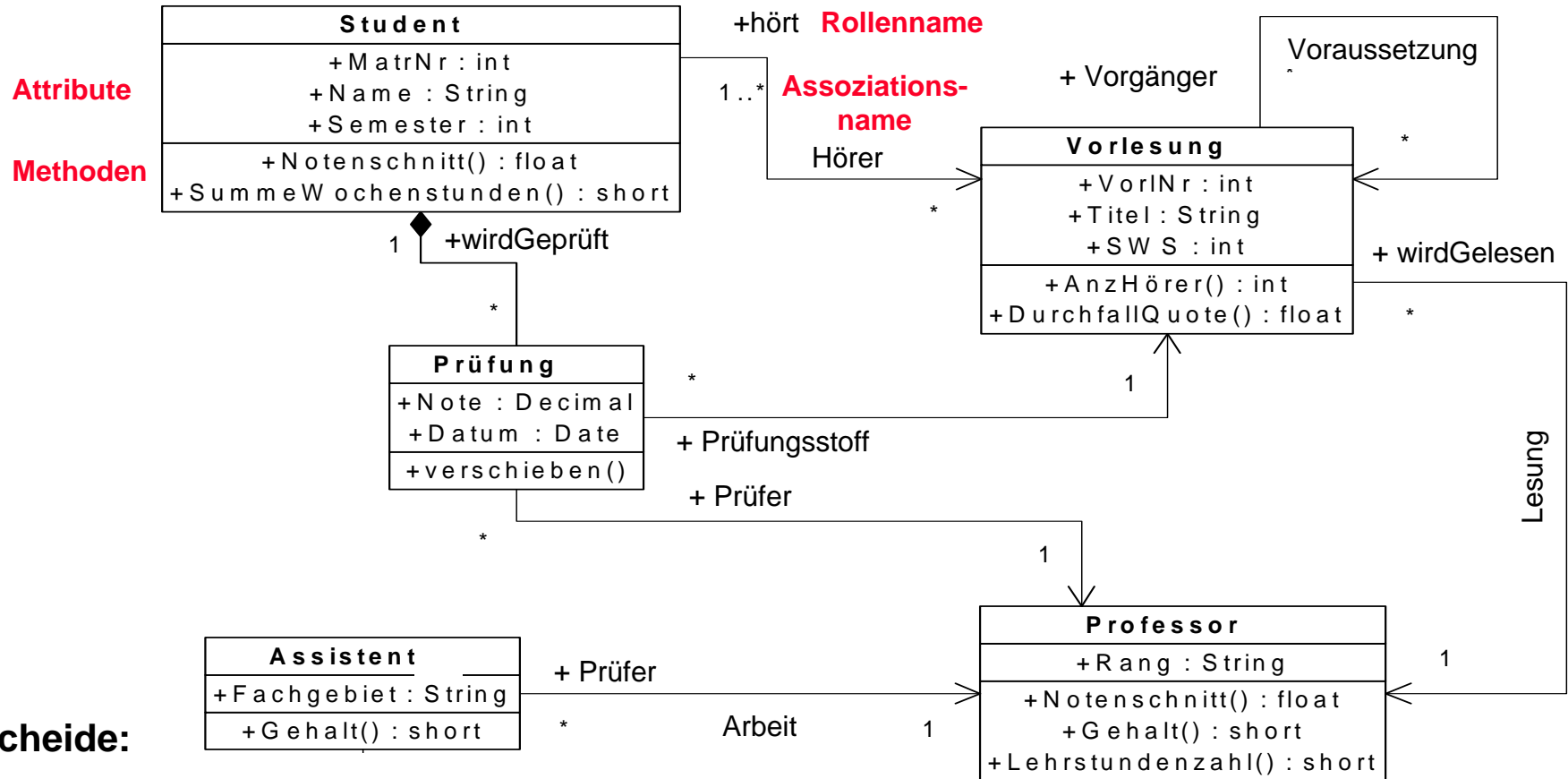
- keine *standardisierte* Kennzeichnung und Unterscheidung von hierarchischer und modularer Zerlegung
- keine klare Trennung zwischen Entität und Beziehung
- keine klare Abgrenzung verschiedener Entitätstypen voneinander
- keine Sichtbarkeits- und Zugriffsregeln auf die Entitäten von außen

Lösung: Objektorientierte Modellierung

- **Vererbung** als Mittel hierarchischer Zerlegung
- **Kapselung** der Objekte (Entitäten): Zugriff von außen nur über Methoden
- weitere Kennzeichnungen und Unterscheidungen (nur in UML-Notation)

Objektorientierte Modellierung

Universitätsbeispiel: Objektorientierte Modellierung (in UML-Notation):

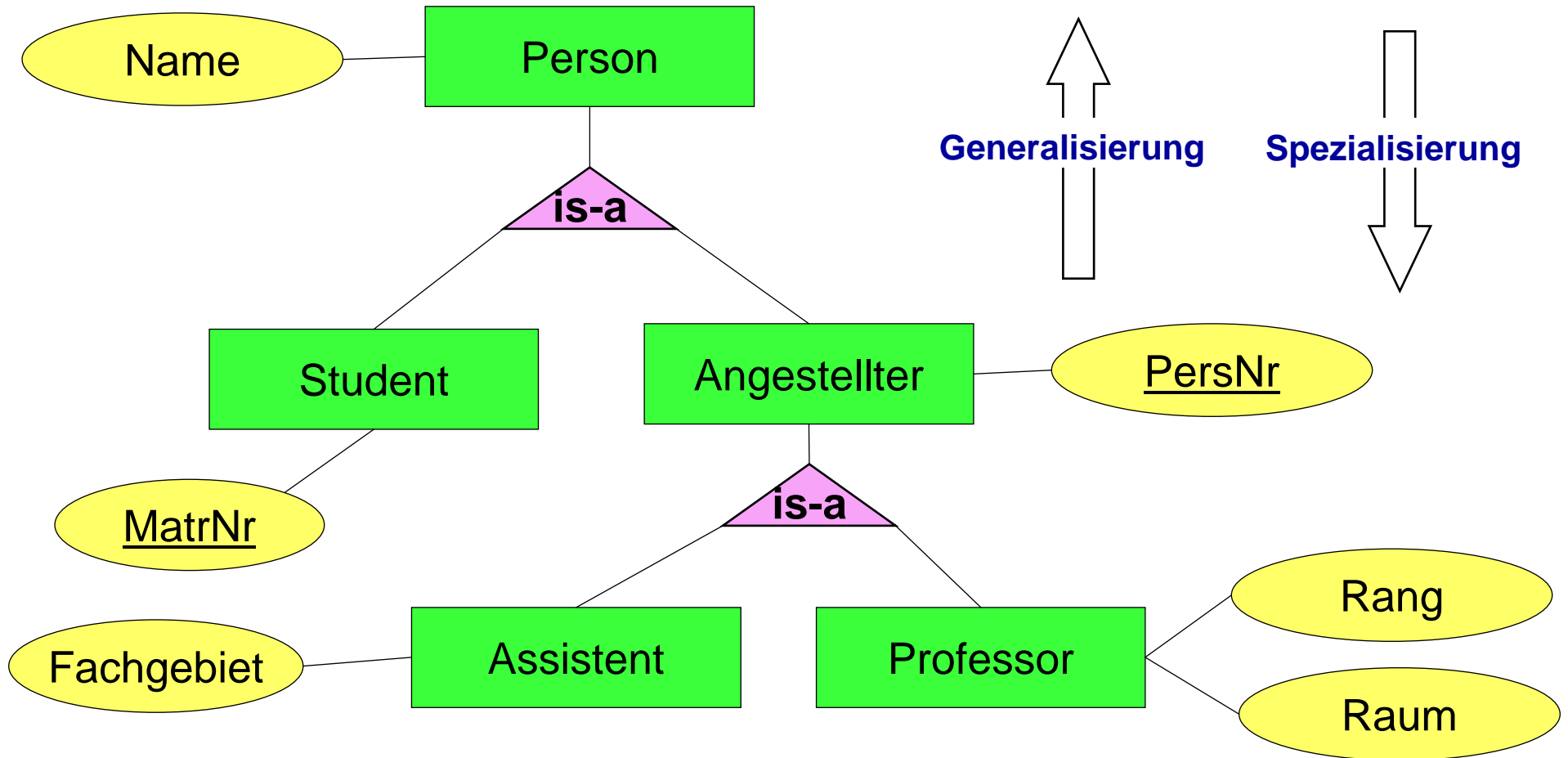


Unterscheide:

- Attribute von Methoden
- Rollen von Assoziationen
- gerichtete von ungerichteten Assoziationen (erst bei Systementwurf relevant)

Objektorientierte Modellierung: Vererbung

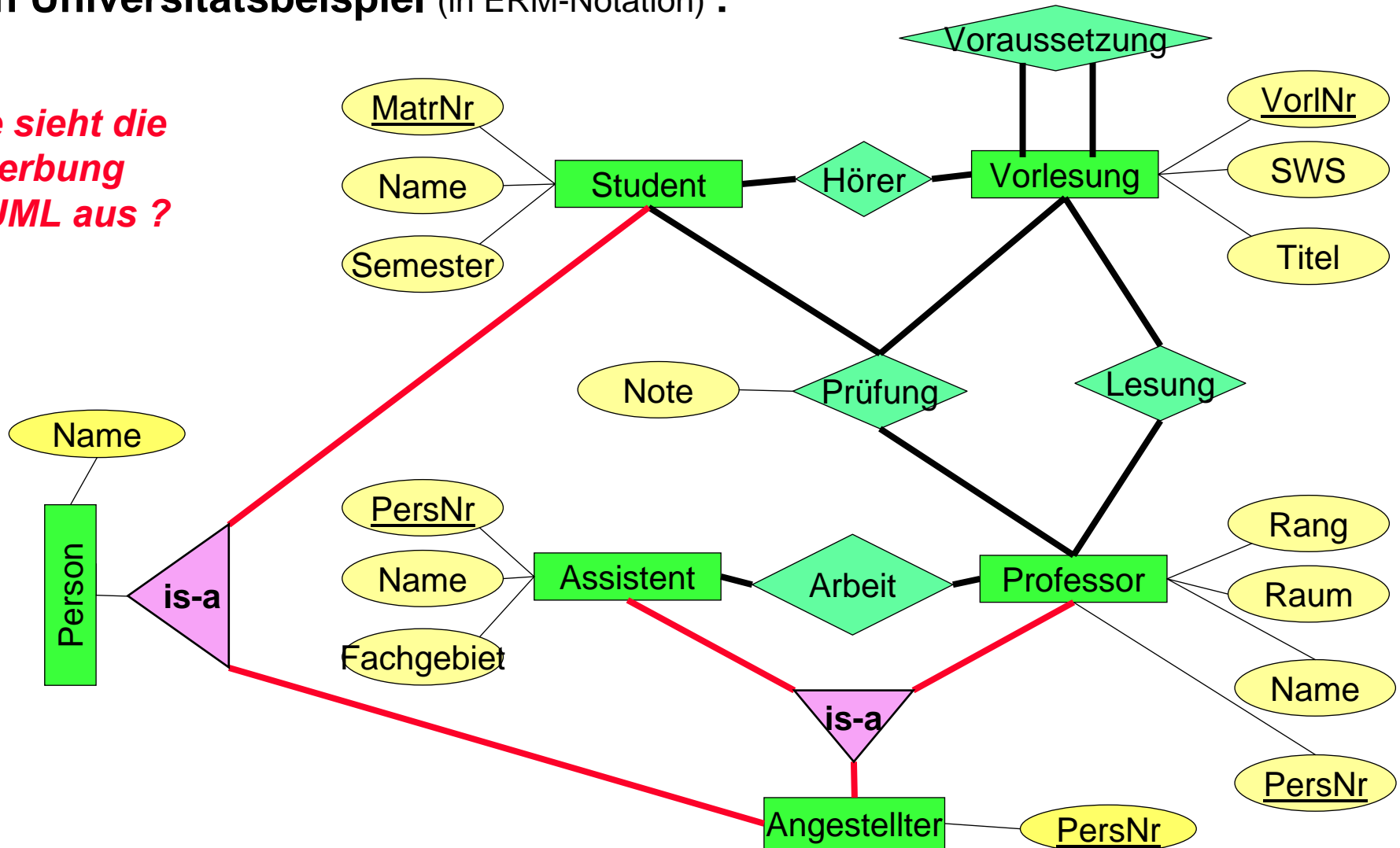
Beispiel: **Personenhierarchie** (in ERM-Notation):



Objektorientierte Modellierung: Vererbung

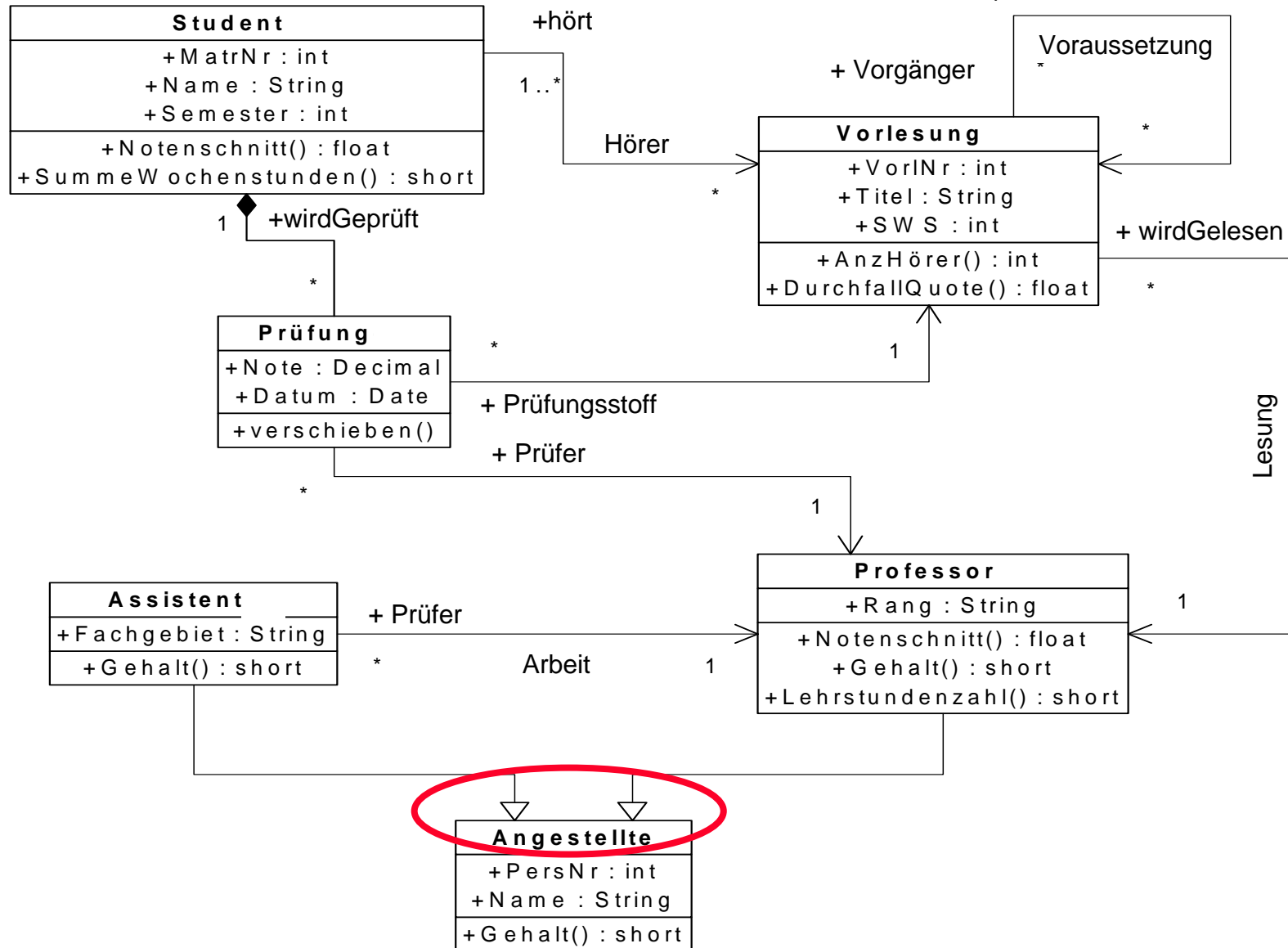
Einbau der Personenhierarchie
in Universitätsbeispiel (in ERM-Notation) :

Wie sieht die Vererbung in UML aus ?

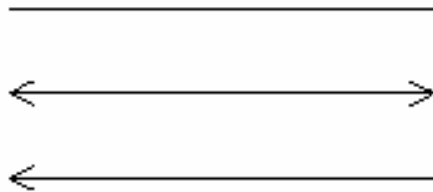


Das Vererbungsbeispiel in UML-Notation

(nur eine Hierarchiestufe)

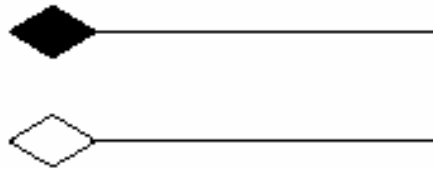


UML-Notation für Beziehungen

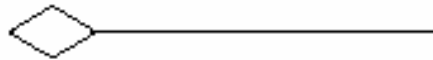


Assoziation (gerichtet / ungerichtet)
(nicht genauer spezifizierte Beziehung)

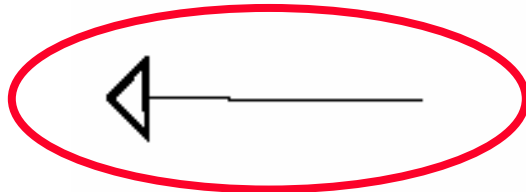
Speziellere Beziehungen:



Komposition
Aggregation



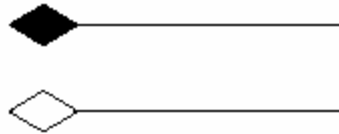
Abhängigkeit



Generalisierung
(für die Vererbung)

UML-Notation für Beziehungen

Was bedeuten diese beiden Pfeile ?

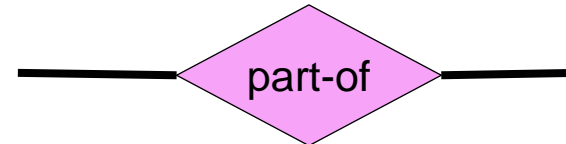


Komposition
Aggregation

Aggregation:

- Zerlegung eines Objekts in seine Bestandteile

Aggregationen in ERM-Notation:

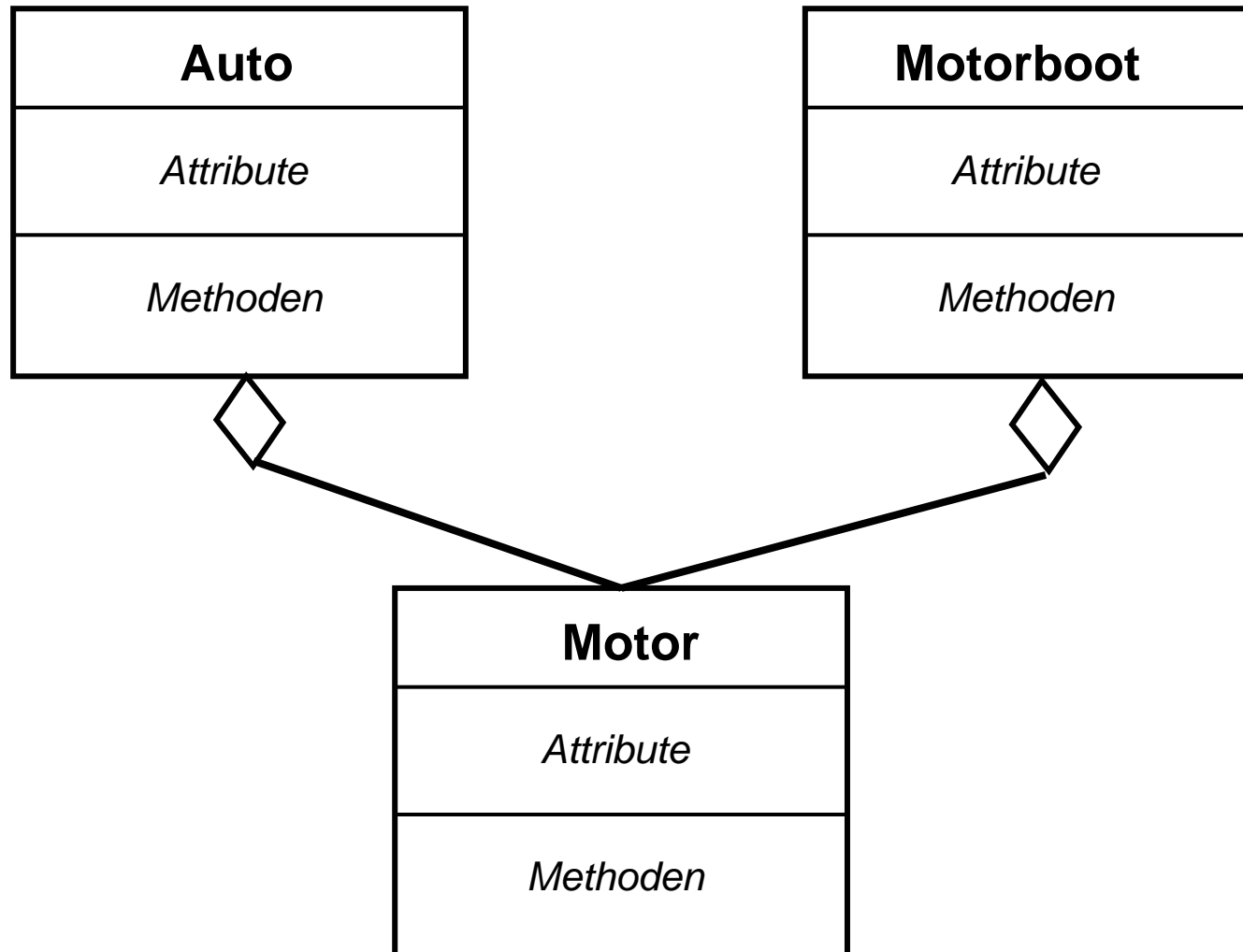


Komposition:

- Zerlegung eines Objekts in Bestandteile, die selbständig gar nicht existieren können

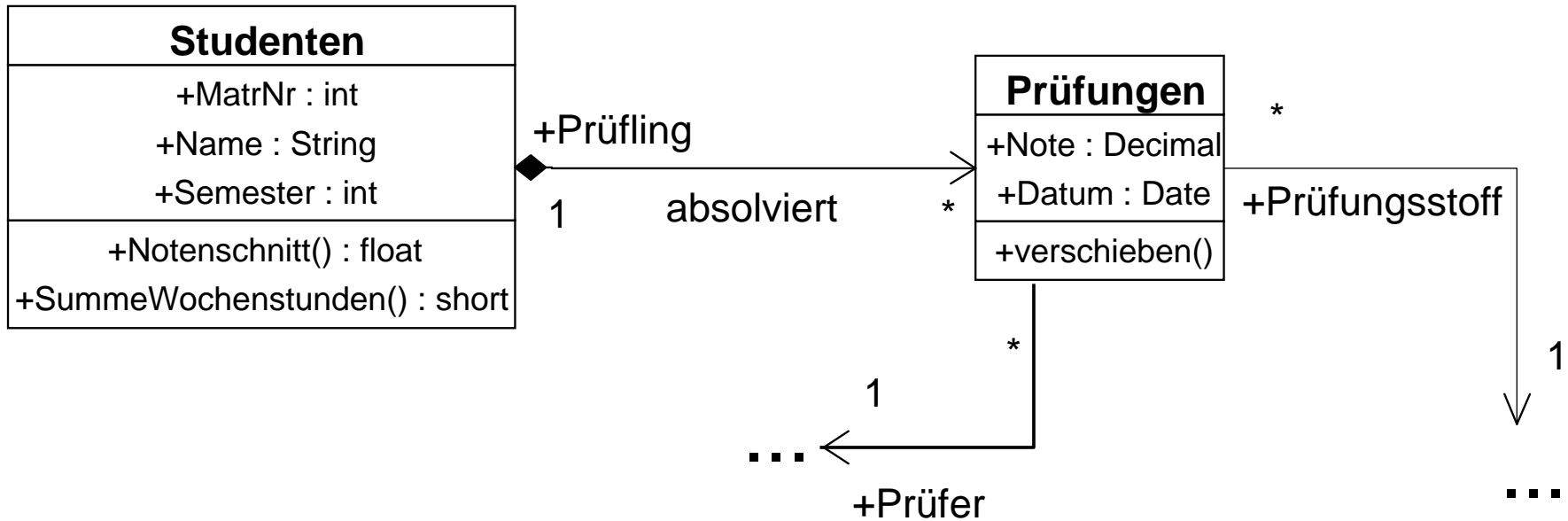
UML-Notation für Beziehungen

Bsp. für Aggregation



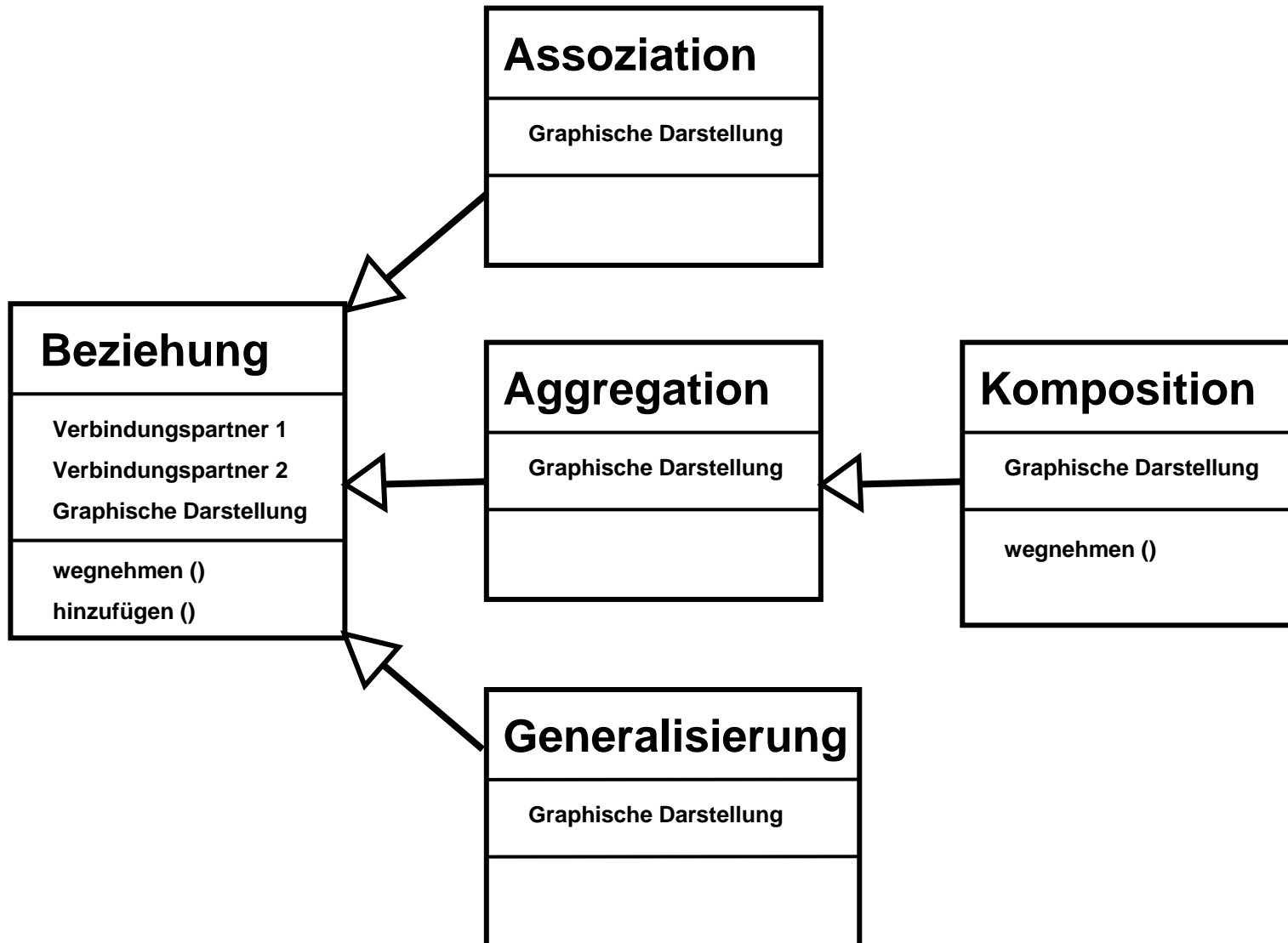
UML-Notation für Beziehungen

Bsp. für Komposition



UML-Notation für Beziehungen

Hierarchische Zerlegung der Verbindungsklasse "Beziehung"



Objektorientierte Modellierung: Methoden

Methoden

- **sind Funktionen, die auf eine Instanz der Klasse angewendet werden dürfen, in der sie beschrieben sind.**
- **„Anwendung“ heißt Ausführung der Funktion, d.h. des in der Funktion beschriebenen Algorithmus**
- **Methoden haben implizit immer die Instanz, auf die sie angewendet werden, als Parameter und optional noch weitere Parameter (die in der Beschreibung der Methode explizit festgelegt sein müssen)**

Objektorientierte Modellierung: Bsp. für Methoden

Attribute

Auto

Zündschloss
Gaspedal
Kupplung
Bremse
Lenkrad

Methoden

anlassen ()
fahren ()
anhalten ()

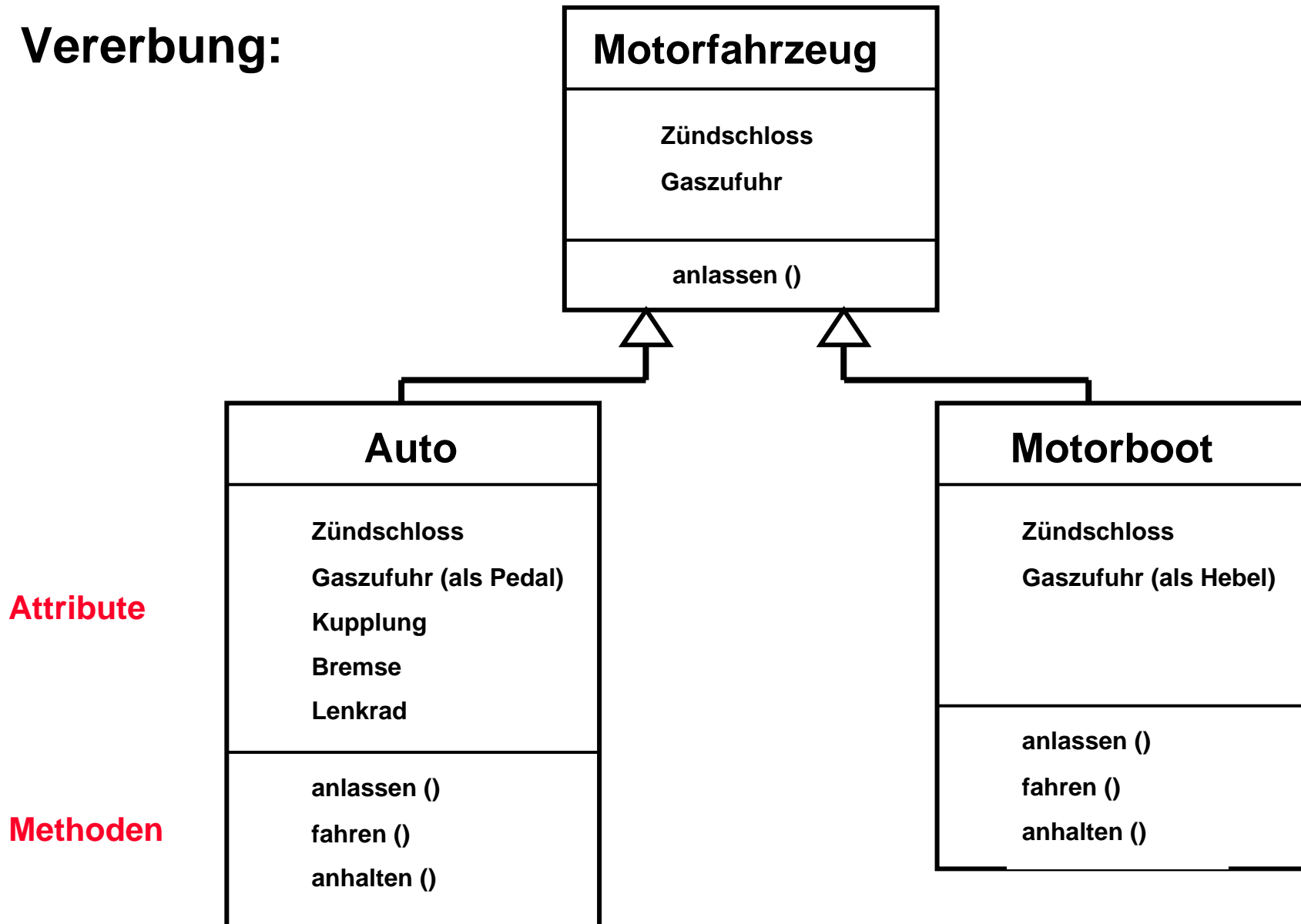
Motorboot

Zündschloss
Gashebel

anlassen ()
fahren ()
anhalten ()

Objektorientierte Modellierung: Bsp. für Methoden

Vererbung:



Vergleich der beiden datenorientierten Sichten

Rein datenorientierte Sicht: ERM

Datenorientierte Sicht mit funktionalen Aspekten: Objektorientierte Sicht (UML)

Konzeptionelle Gemeinsamkeiten:

- *Entitäten* werden zu gleichartigen *Entitätstypen* zusammengefasst.
- *Instanzen* werden zu gleichartigen *Klassen* zusammengefasst.

- *Entitäten* werden durch *Attribute* näher beschrieben.
- *Instanzen* werden durch *Attribute* näher beschrieben.

- *Entitätstypen* sind durch *Assoziationen* miteinander verbunden, welche die Beziehungen der zugehörigen Entitäten beschreiben. Diese *Assoziationen* können von den *Entitätstypen* aus verschieden bezeichnet werden. Außerdem kann die Anzahl der beteiligten Entitäten eines *Entitätstyps* festgelegt werden.
- *Klassen* sind durch *Assoziationen* miteinander verbunden, welche die Beziehungen der zugehörigen Instanzen beschreiben. Diese *Assoziationen* werden von den Instanzen aus mit *Rollennamen* bezeichnet. Außerdem kann die Anzahl der beteiligten Instanzen einer Klasse festgelegt werden.

Vergleich der beiden datenorientierten Sichten

Rein datenorientierte Sicht: ERM

Datenorientierte Sicht mit funktionalen Aspekten: Objektorientierte Sicht (UML)

Konzeptionelle Unterschiede:

- *Entitäten* werden durch *Schlüsselattribute* eindeutig bestimmt.
 - *Instanzen* werden durch ihren *Namen* eindeutig bestimmt.
- **Zugriff auf die Attribute nicht versteckt. Keine entitätsspezifischen Prozesse**
 - **Methoden regeln den Zugriff auf die Instanzen sowie Operationen auf ihnen. Außerdem werden durch Methoden Operationen mit den Instanzen beschreiben. Methoden werden in den zugehörigen Klassen deklariert.**
- **Das Konzept der hierarchischen Ordnung von Entitätstypen wurde nachträglich in die ER-Modellierung aufgenommen (is-a). Die Vererbung bezieht sich auf die Attribute.**
 - **Klassen können hierarchisch geordnet werden (Generalisierung und Spezialisierung). Die Instanzen einer spezialisierten Klasse erben alle Attribute und Methoden der generalisierten Klasse, es sei denn, ihre eigene Klasse hat selbst ein Attribut / eine Methode desselben Namens.**