Datenbankentwurf

…bisher:

- konzeptueller Entwurf mittels E/R-Modell
- einfache Funktionalitäten
- (min, max)-Notation
- n-stellige Relationships (n>2)

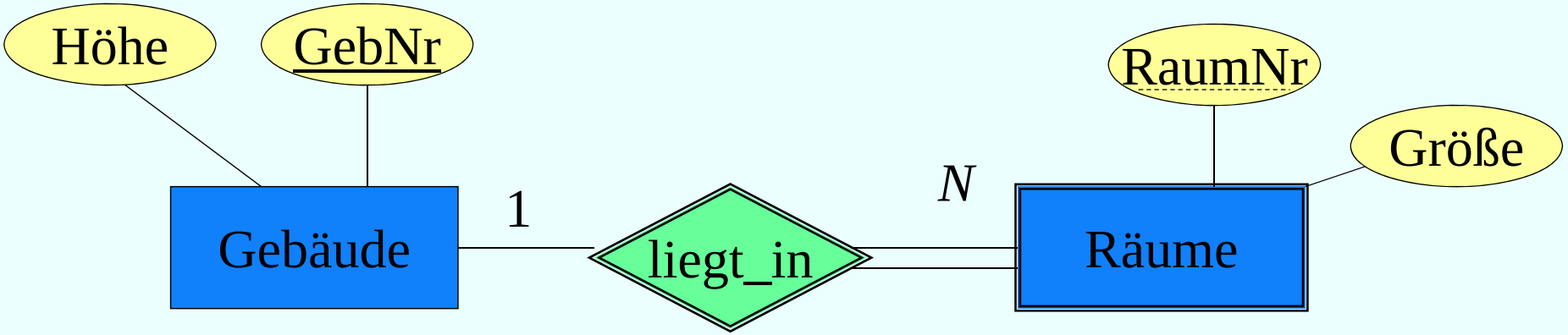
…nun:

schwache Entities, Generalisierung, Aggregation

… dann:

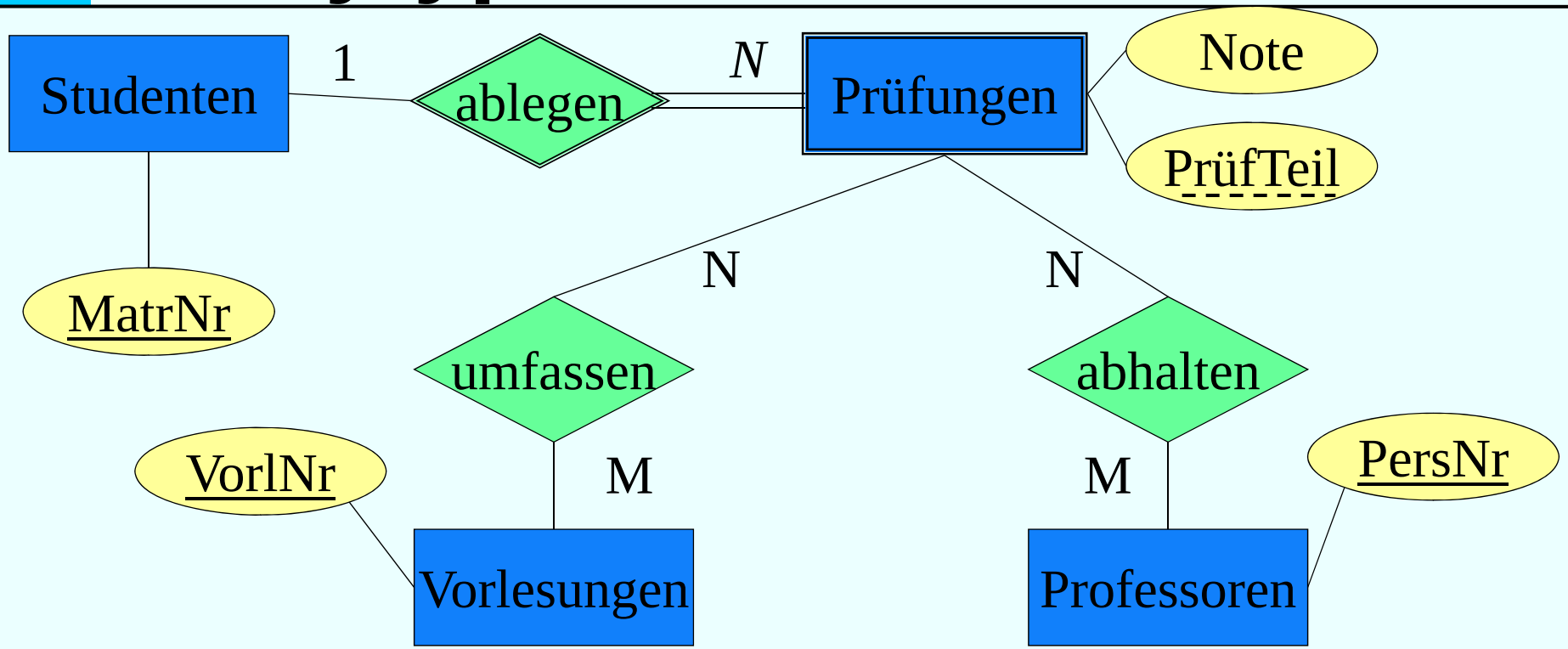
UML-Modell

Schwache, existenzabhängige Entities



- Beziehung zwischen "starken" und schwachem Typ ist immer 1:*N* (oder 1:1 in seltenen Fällen)
- Warum kann das keine *N:M*-Beziehung sein?
- RaumNr ist nur innerhalb eines Gebäudes eindeutig
- Schlüssel ist: GebNr **und** RaumNr

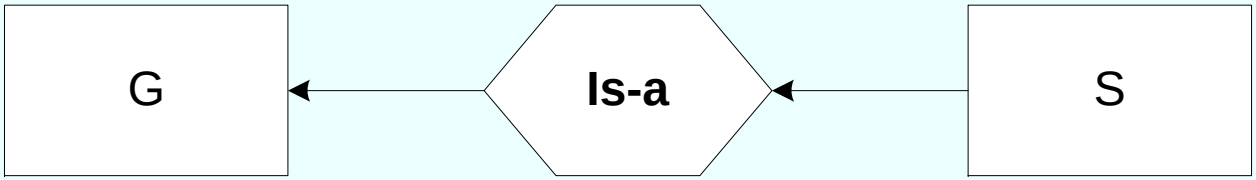
Prüfungen als schwacher Entitytyp



- Mehrere Prüfer in einer Prüfung
- Mehrere Vorlesungen werden in einer Prüfung abgefragt

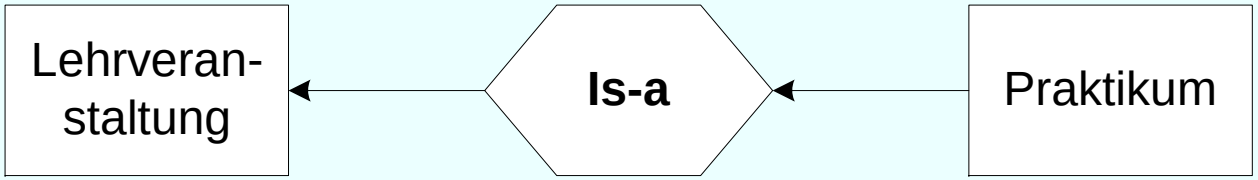
Generalisierung

Generalisierung / Spezialisierung:

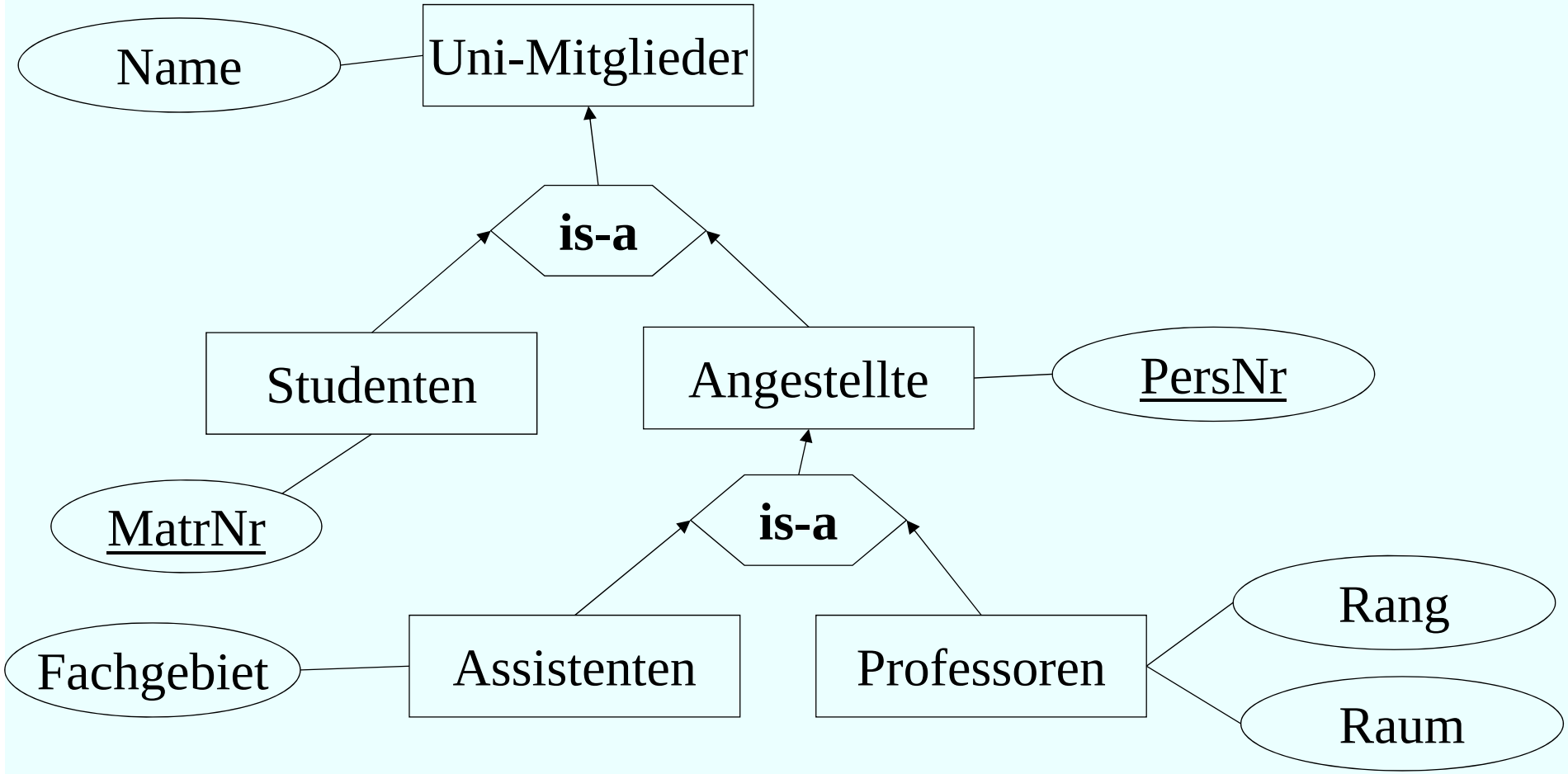


S ist eine Spezialisierung von G

Beispiel:



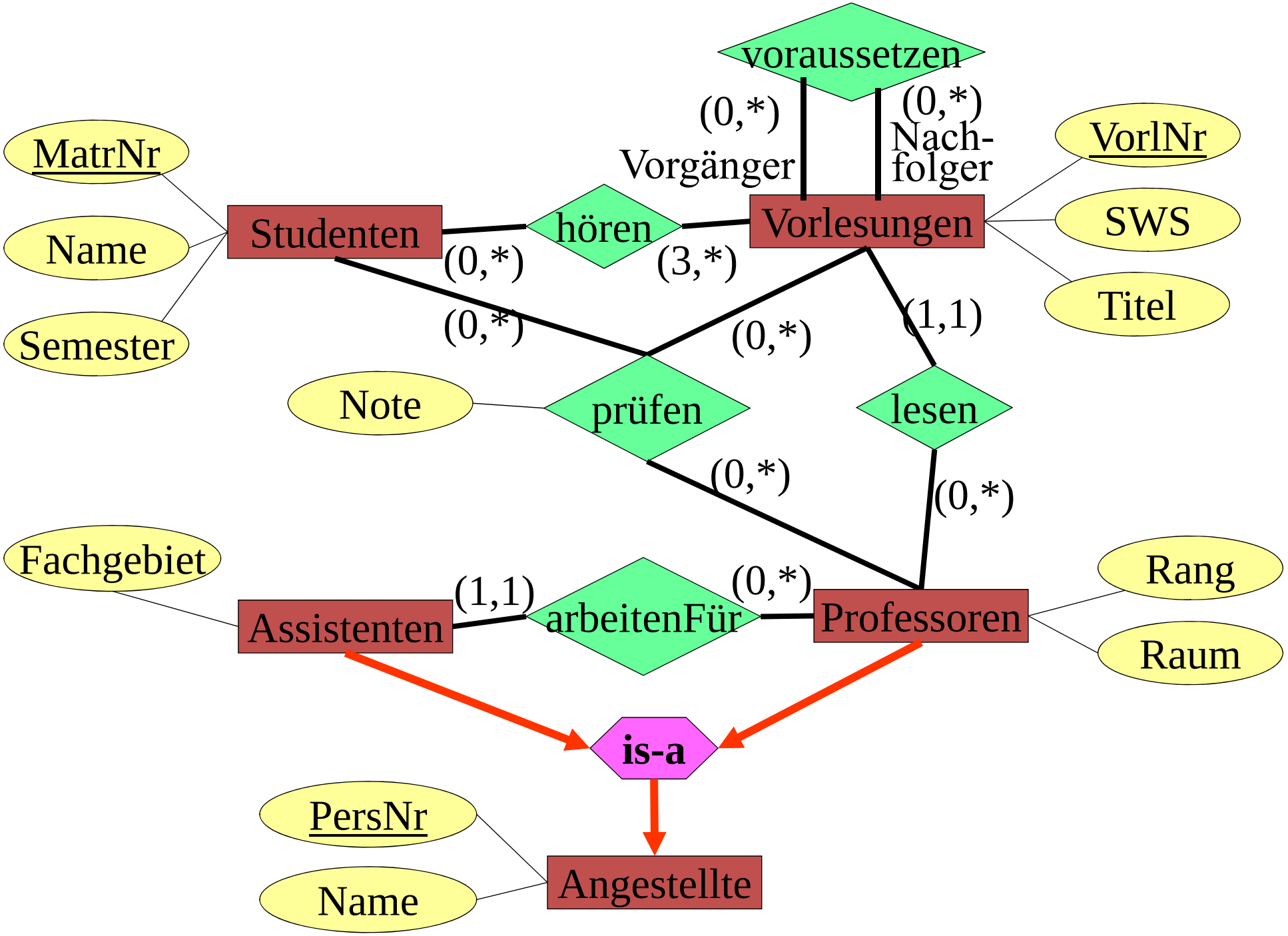
Generalisierung Uni-Beispiel



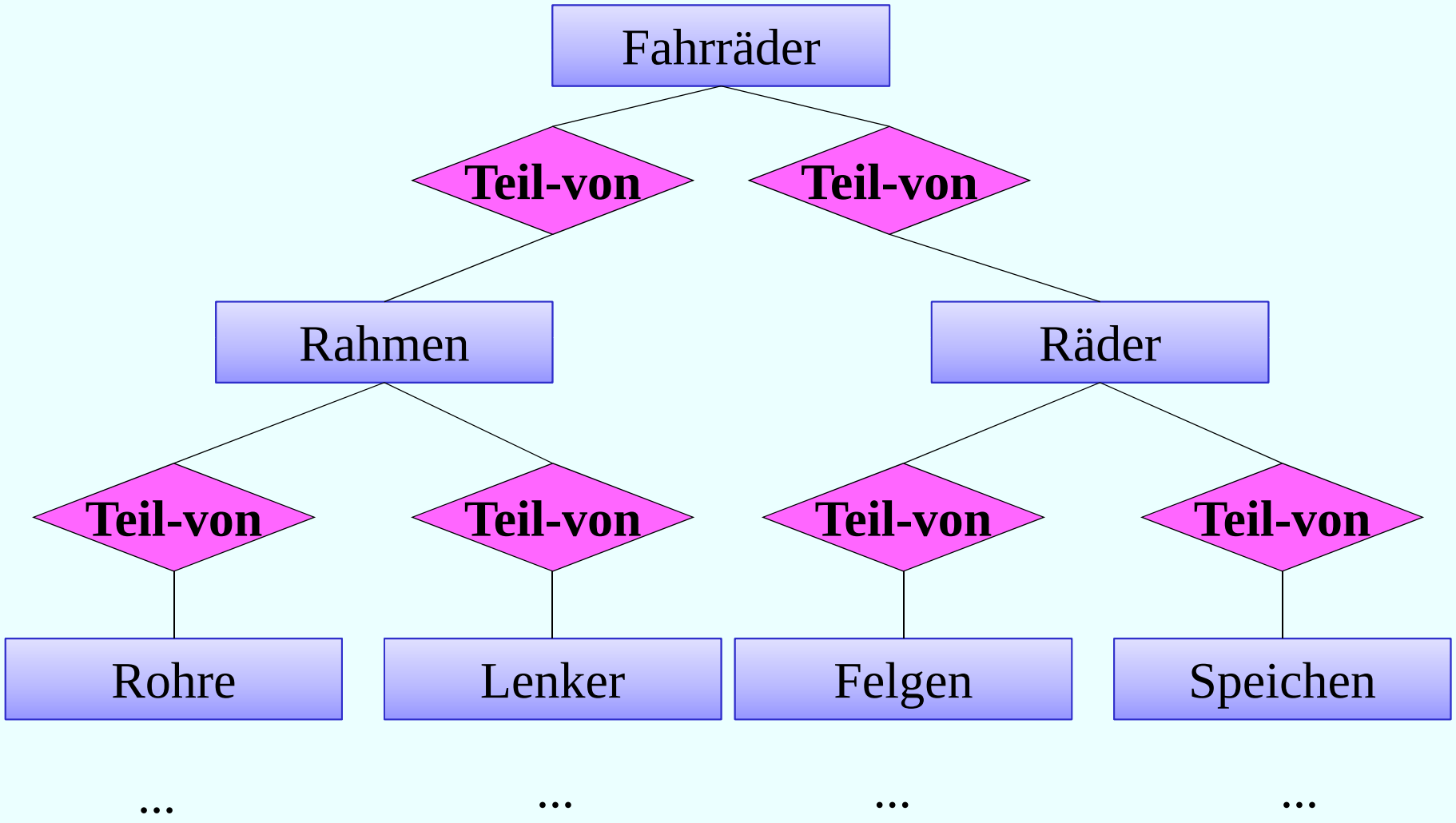
Zusamenfassung

Universitätsschema mit Generalisierung und (min, max)-Markierung

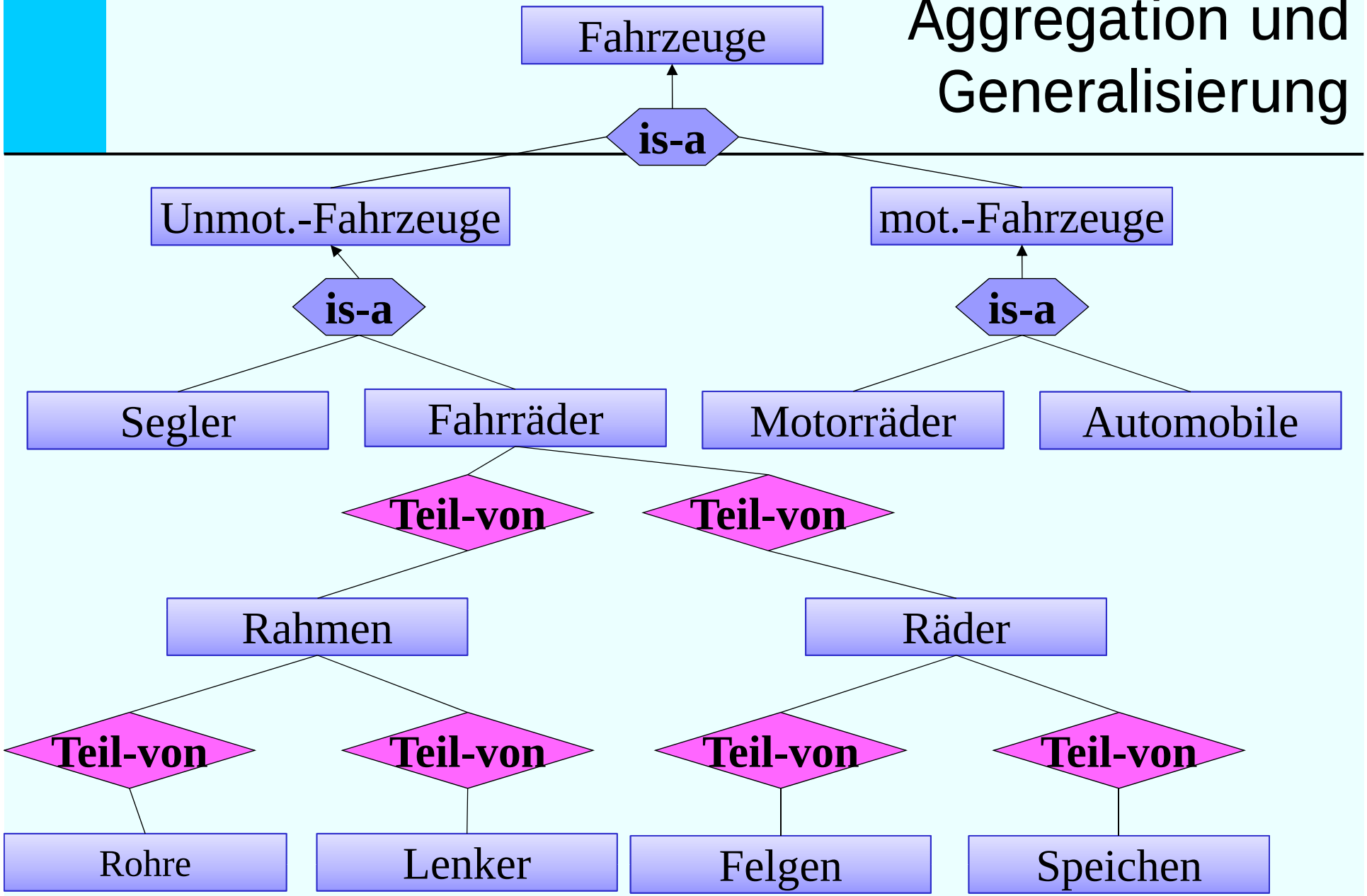
➔ Nächste Seite


voraussetzen
(0,*)
Vorgänger
(0,*)
Nach-
folger
MatrNr
Name
Semester
Studenten
hören
(0,*)
(3,*)
Vorlesungen
VorlNr
SWS
Titel
(0,*)
(0,*)
(1,1)
Note
prüfen
lesen
(0,*)
(0,*)
Fachgebiet
(1,1)
arbeitenFür
(0,*)
Assistenten
Professoren
Rang
Raum
is-a
PersNr
Name
Angestellte

Aggregation



Aggregation und Generalisierung



HPI MOOC

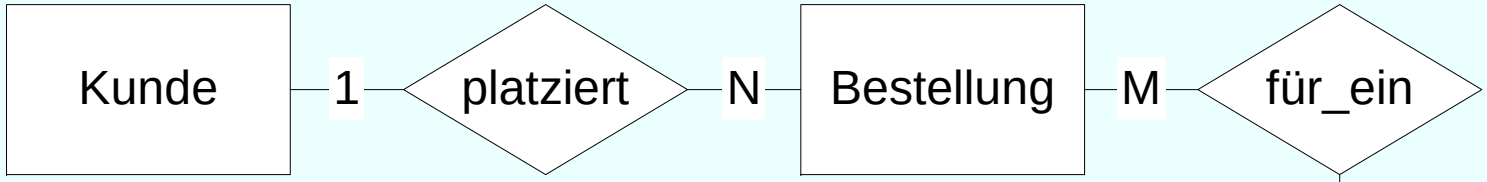
- n-äre Relationships
- Rollen von Relationships
- Konvertierung in binäre Relationships
- Attribute an Relationships

Entwurfskriterien

- Regeln zur Klassifikation von Entities und Attributen:
  - Entities sollten deskriptive Informationen enthalten.
  - Mehrwertige Attribute sollten als Entities klassifiziert werden.
  - Attribute sollten der Entity zugeordnet werden, die sie am direktesten beschreibt.
- Redundante Relationships sollten vermieden werden.
- Wie eine Informationseinheit repräsentiert wird, ist *anwendungsabhängig*.

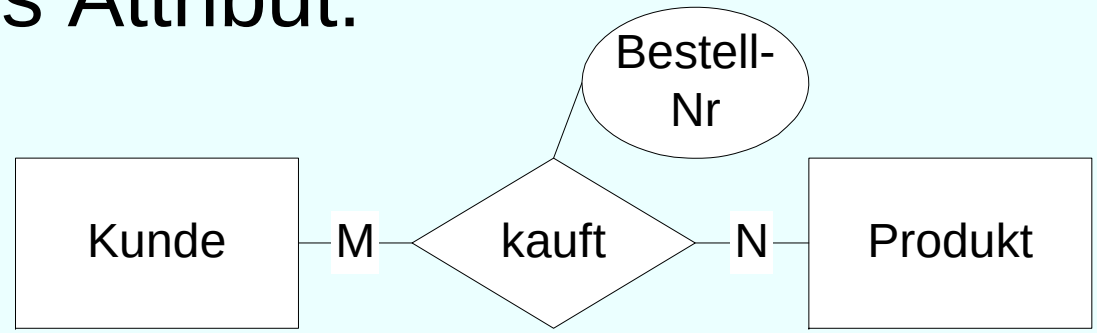
Beispiel: Bestellung

Als Entity:



Als Relationship:

Als Attribut:



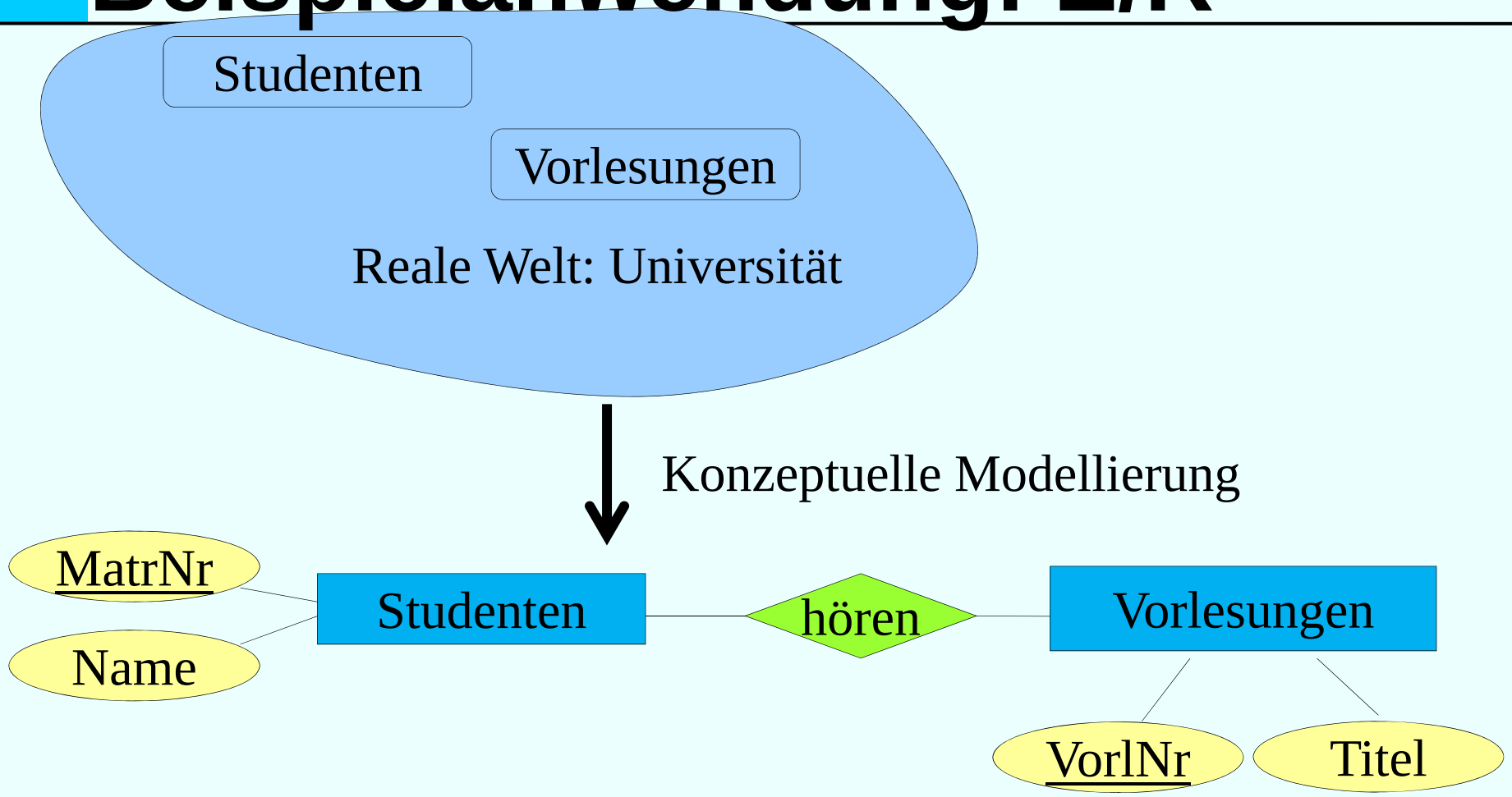
Aufgabe für nächste Woche

Lesen und versuchen Sie zu verstehen aus
Datenmanagement mit SQL, HPI
open.hpi.de/courses/sql:

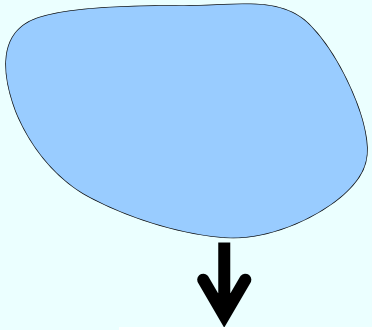
Woche 2: 2.08 –
Designprinzipien

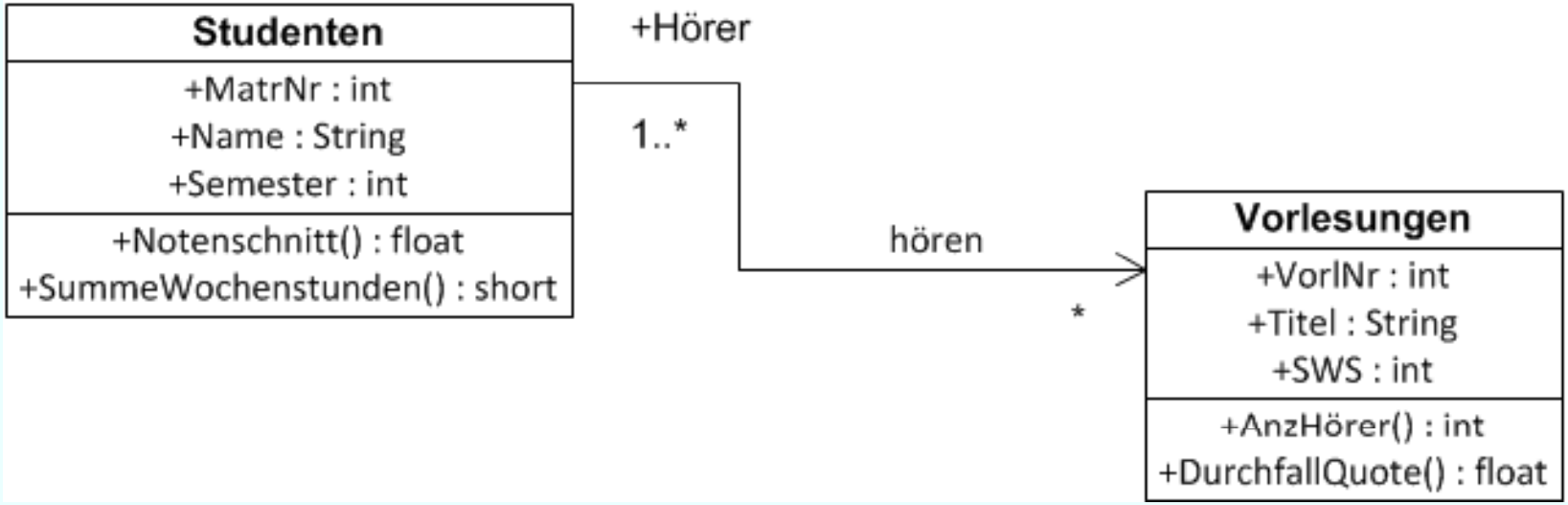
Fragen dazu bitte nächste Woche!

Modellierung einer kleinen Beispielanwendung: E/R



Modellierung einer kleinen Beispielanwendung: UML





Datenmodellierung mit UML

UML: Unified Modelling Language

De-facto Standard für den objekt-orientierten Software-Entwurf

Zentrales Konstrukt: Klasse (class),
modelliert gleichartige Objekte hinsichtlich

- Struktur (~Attribute)
- Verhalten (~Operationen/Methoden)

Assoziationen zwischen Klassen entsprechen Beziehungstypen

Generalisierungshierarchien

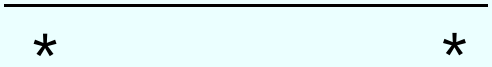
Aggregation

Cheat sheet Class Diagram:

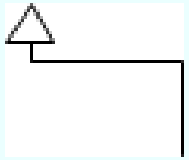
http://www.code-meets-design.de/wp-content/uploads/2013/07/uml-classdiagram-cheat-sheet.pdf

UML Notation

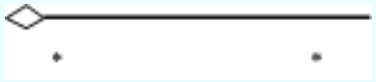
Assoziation:



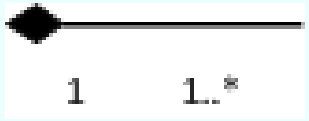
Generalisierung:



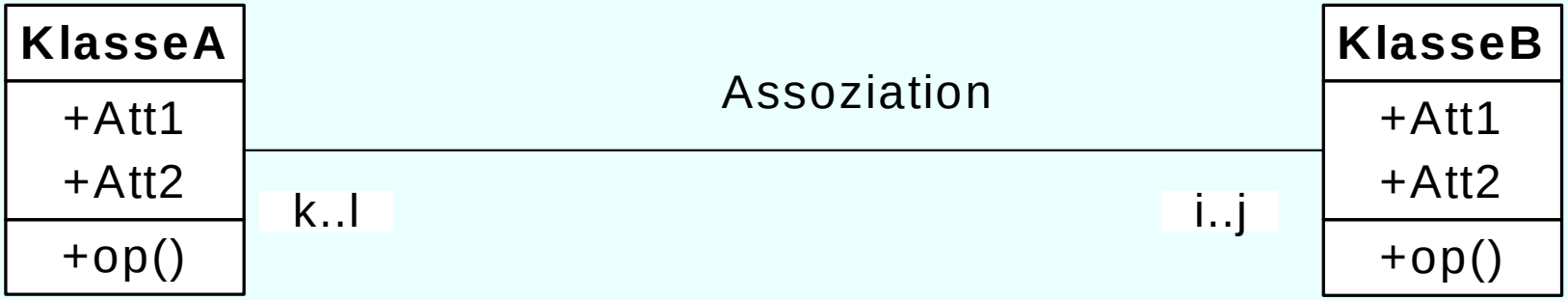
Aggregation:
(Teil-von)



Komposition:
(Spezialfall von Aggregation)



Multiplizität



Jedes Element von KlasseA steht mit mindestens i Elementen der KlasseB in Beziehung

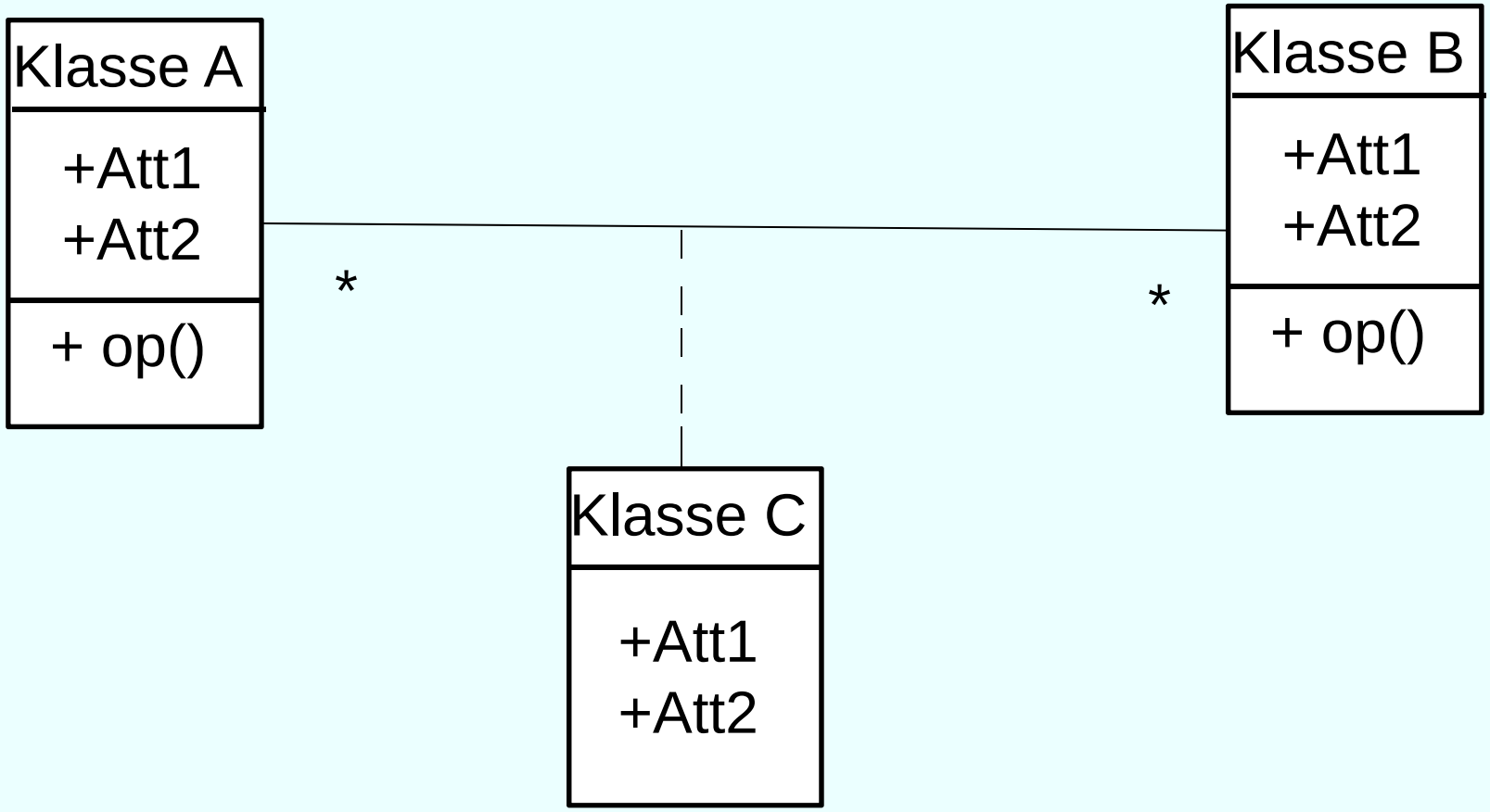
... und mit maximal j vielen KlasseB-Elementen

Analoges gilt für das Intervall k..l

Multiplizitätsangabe ist analog zur einfachen Funktionalitätsangabe im ER-Modell

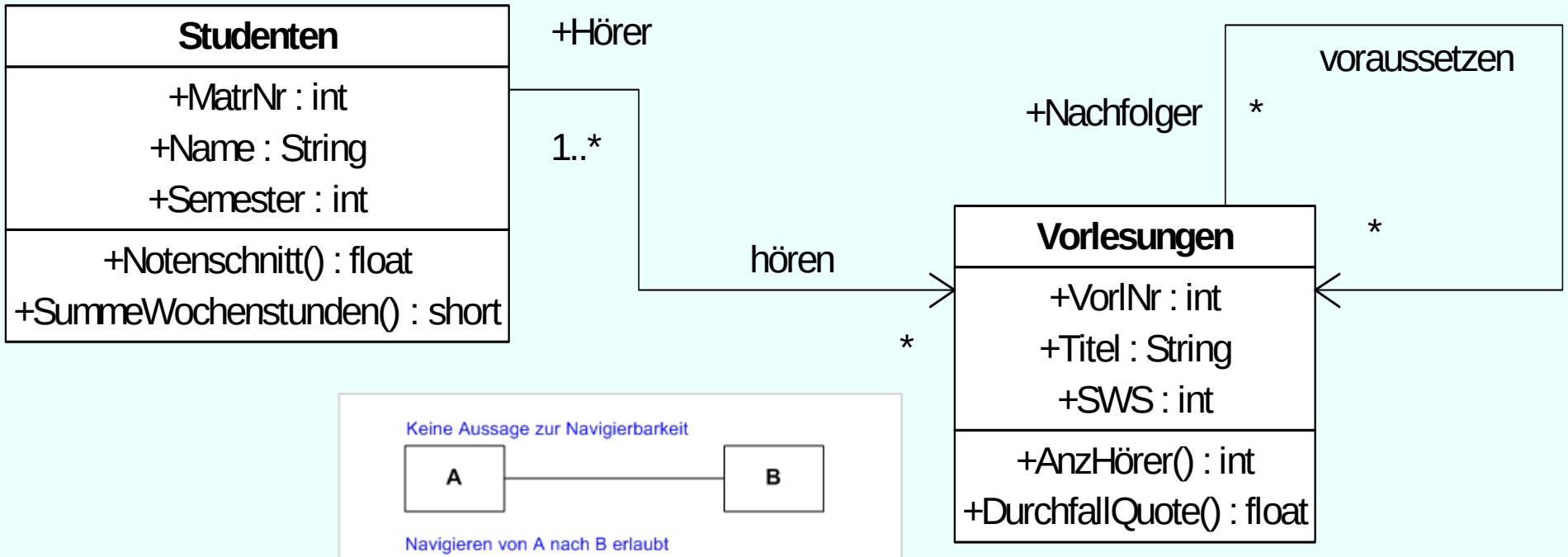
Nicht zur (min,max)-Angabe: **Vorsicht!**

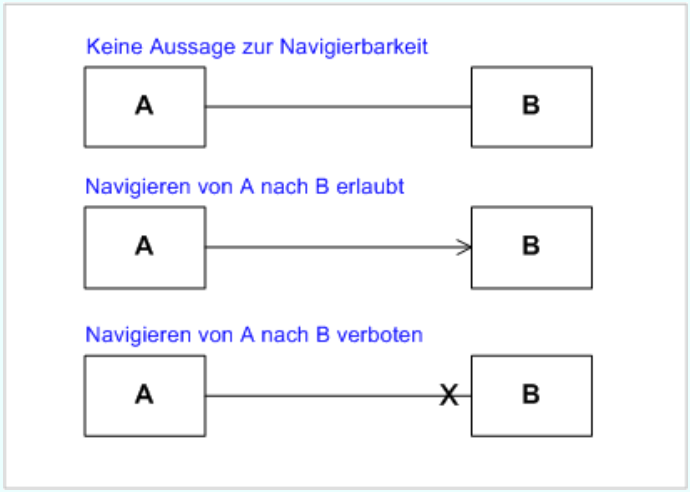
Assoziationsklasse



… für Attribute der Assoziation

Klassen und Assoziationen

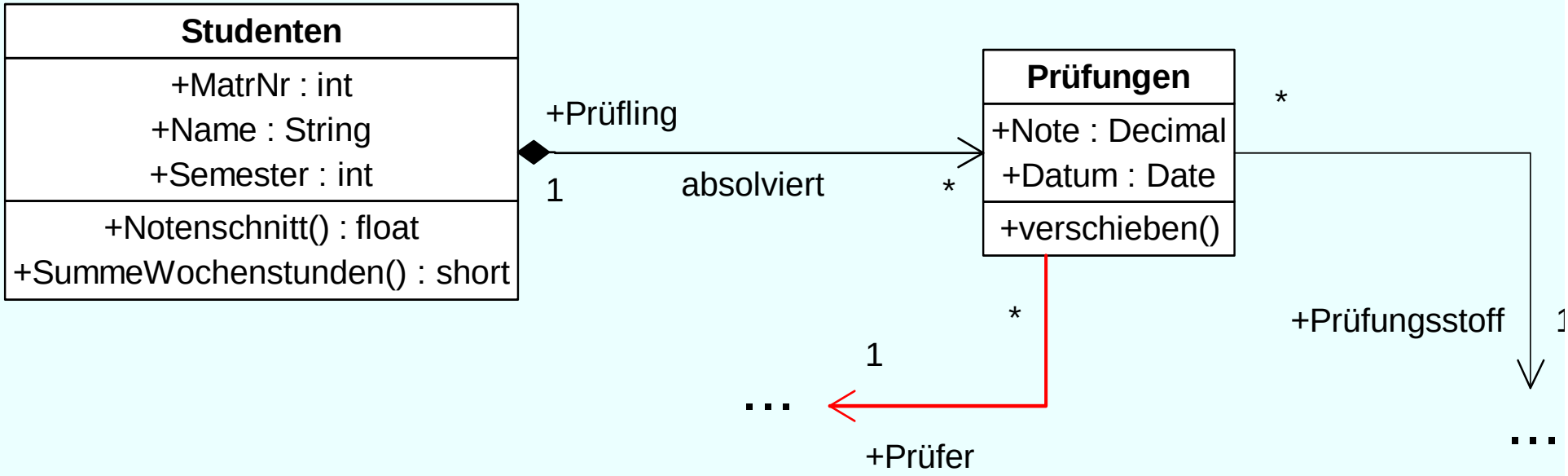




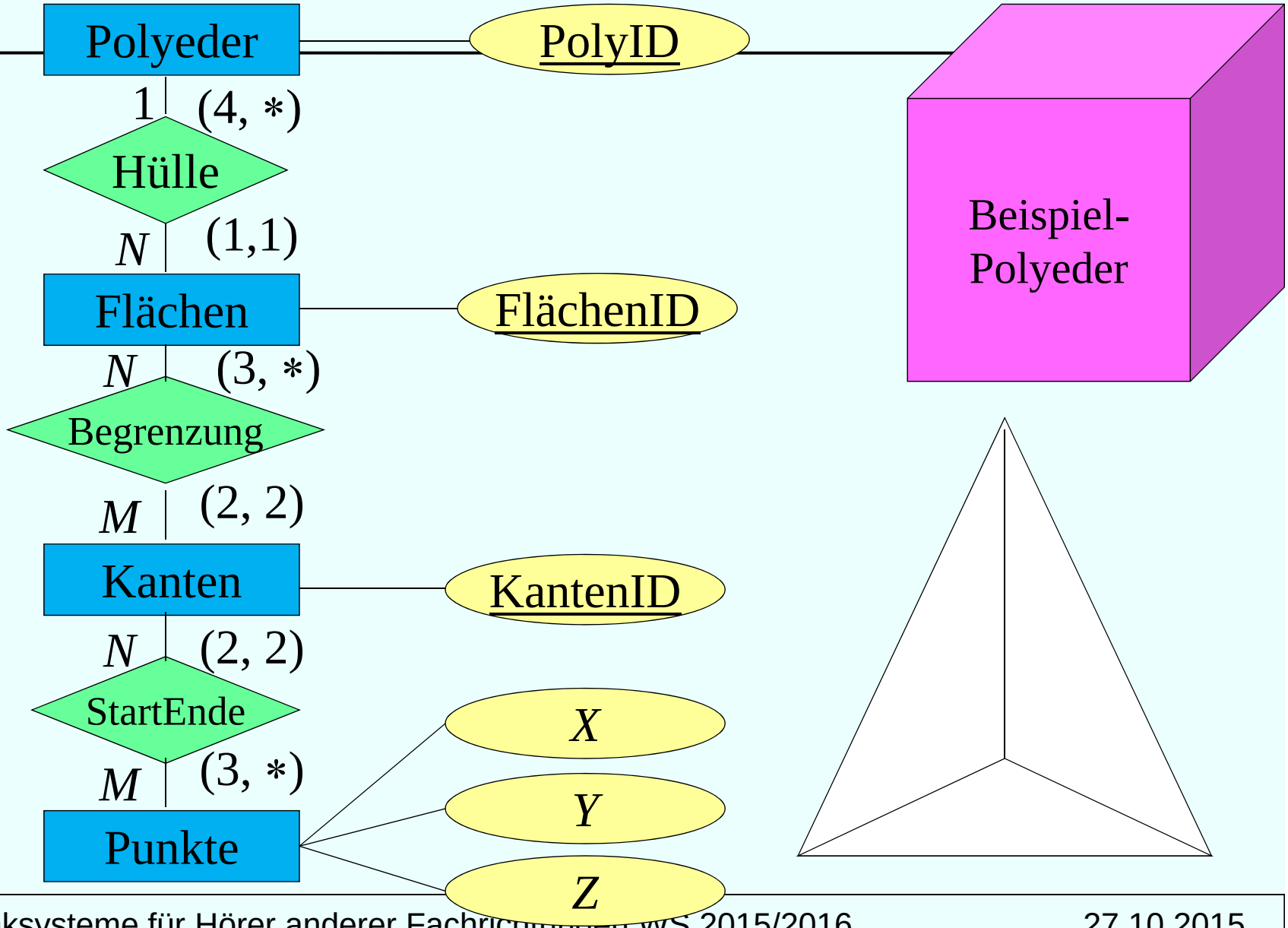
Legende:

+: public

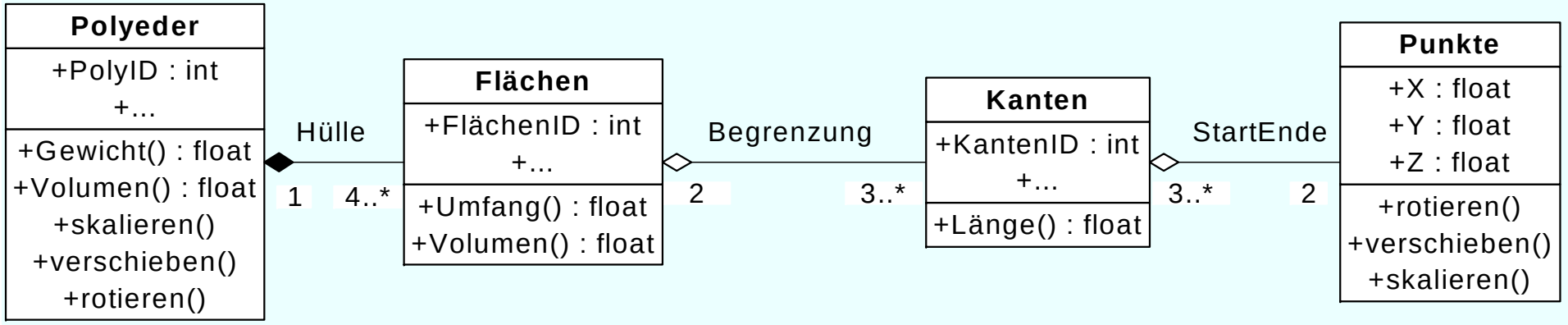
Komposition

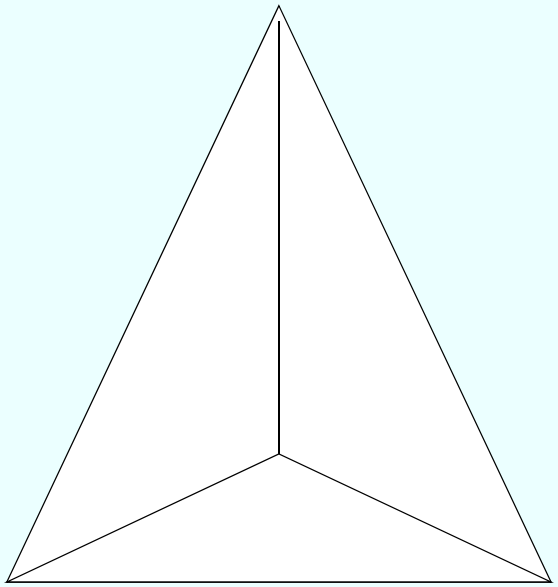


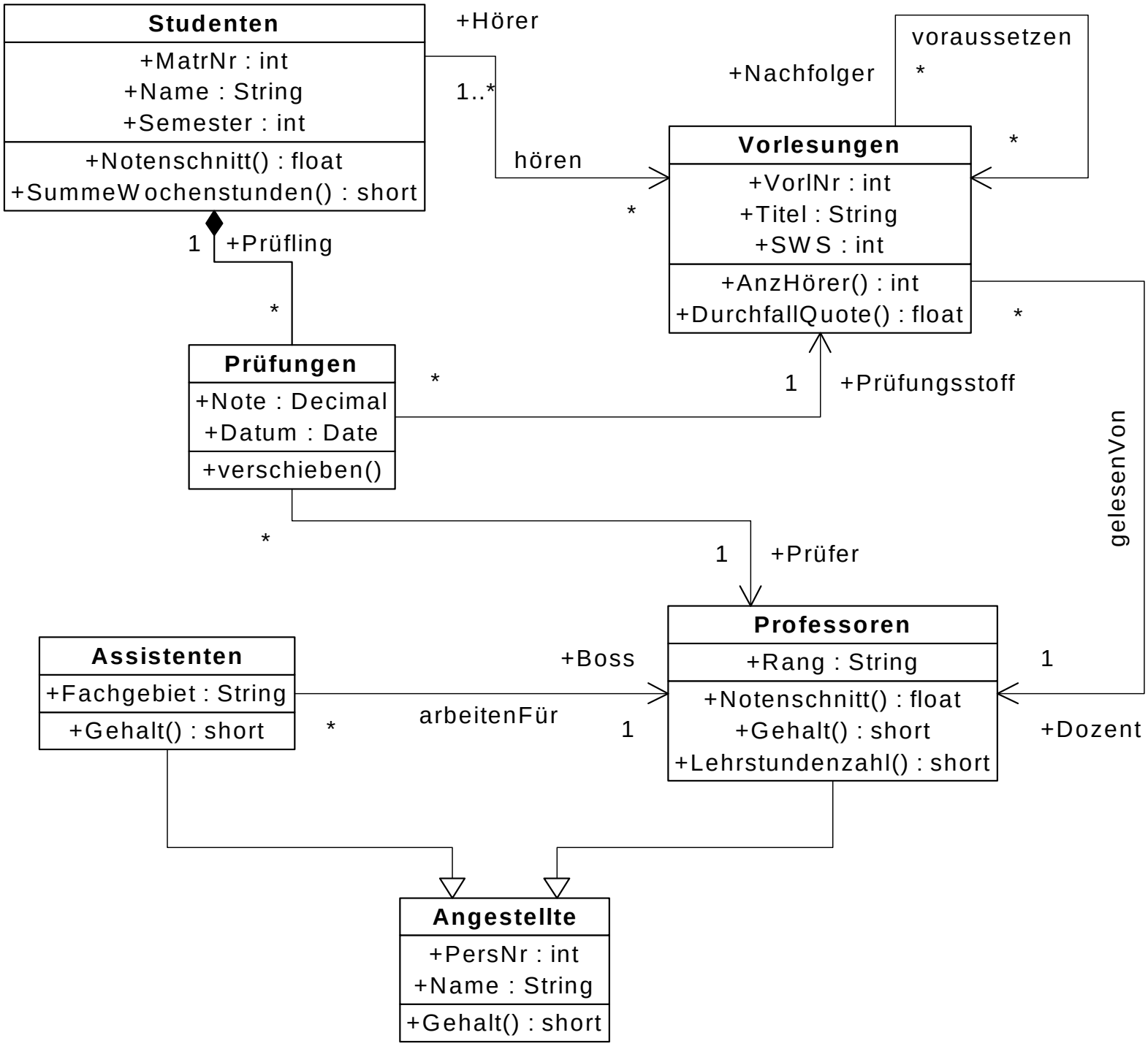
Begrenzungsflächendarstellung



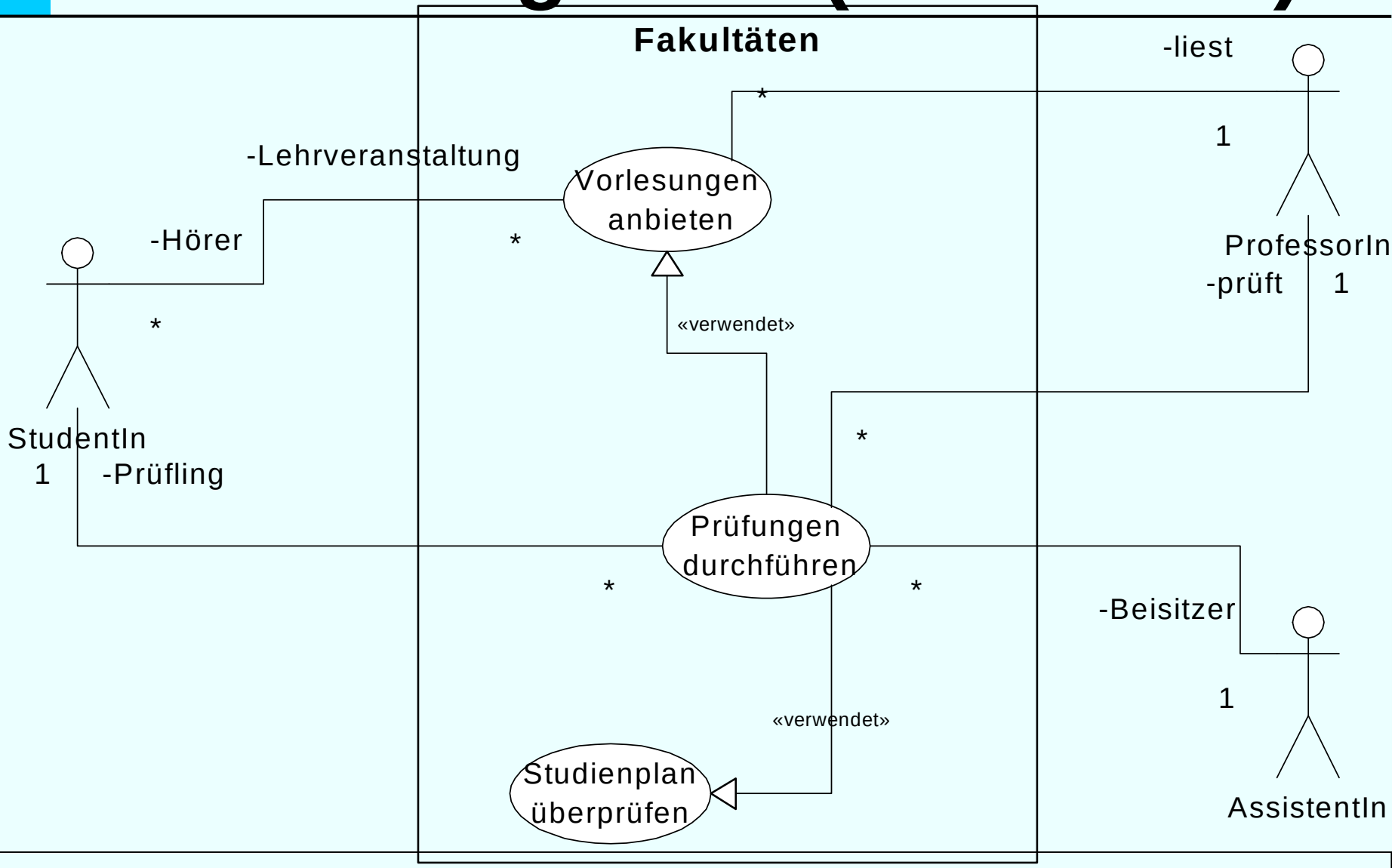
Begrenzungsflächenmodellierung von Polyedern in UML



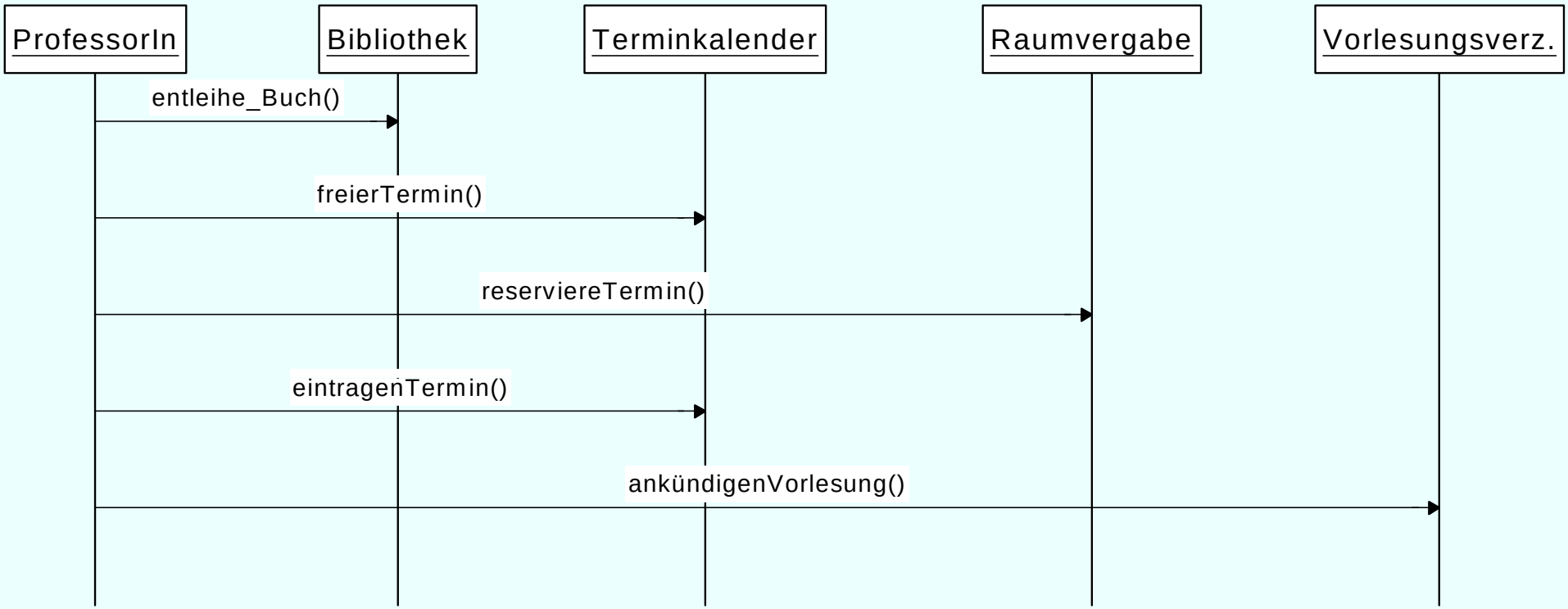



Studenten
+MatrNr : int
+Name : String
+Semester : int
+Notenschnitt() : float
+SummeWochenstunden() : short
+Hörer
1..*
hören
*
Vorlesungen
+VorlNr : int
+Titel : String
+SWS : int
+AnzHörer() : int
+DurchfallQuote() : float
+Nachfolger
voraussetzen
*
*
1
+Prüfling
*
Prüfungen
+Note : Decimal
+Datum : Date
+verschieben()
*
1
+Prüfungsstoff
*
gelesenVon
*
1
+Prüfer
Professoren
+Rang : String
+Notenschnitt() : float
+Gehalt() : short
+Lehrstundenzahl() : short
1
+Dozent
Assistenten
+Fachgebiet : String
+Gehalt() : short
+Boss
*
arbeitenFür
1
Angestellte
+PersNr : int
+Name : String
+Gehalt() : short

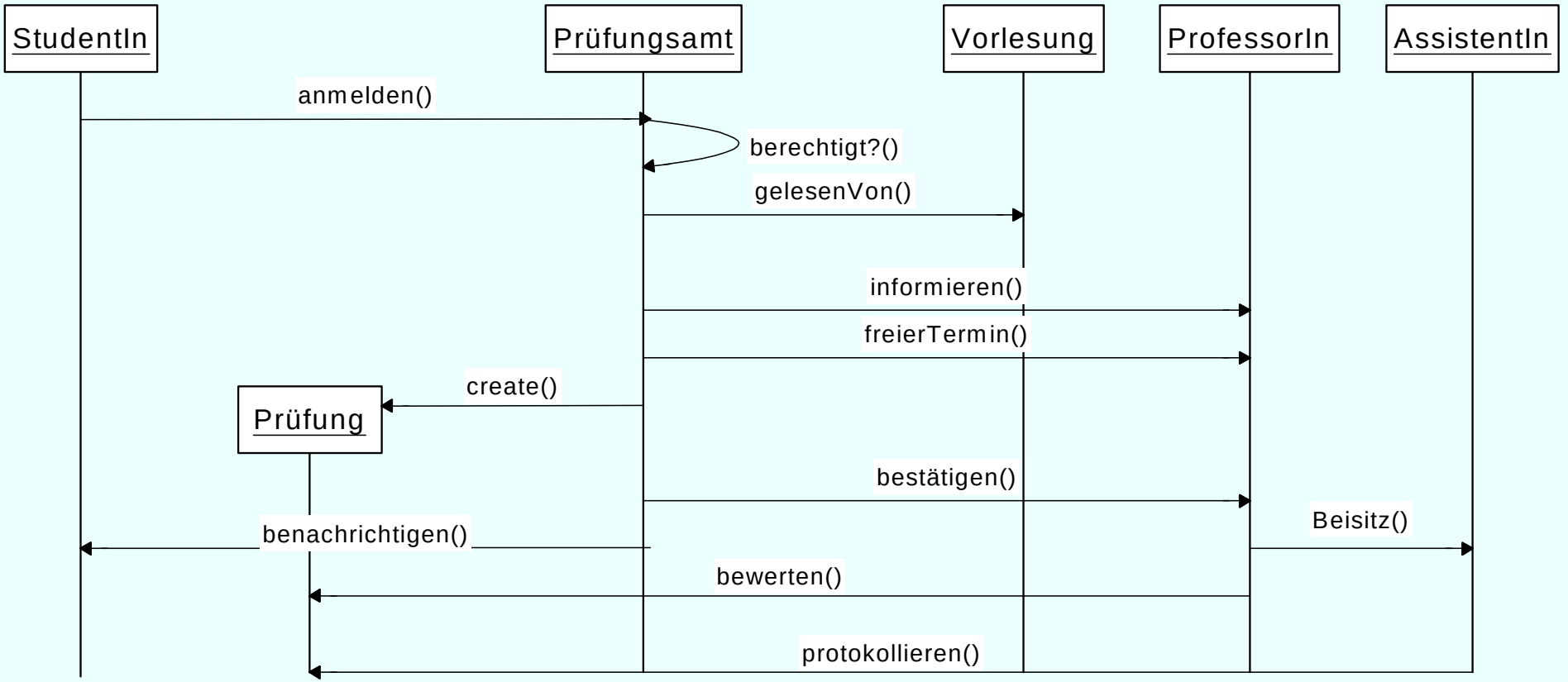
Anwendungsfälle (use cases)



Interaktions-Diagramm:
Modellierung komplexer Anwendungen



Interaktions-Diagramm: *Prüfungsdurchführung*



UML Notationsübersicht:
http://www.oose.de/nuetzliches/fachliches/uml/uml-notationsubersicht/