Assoziationen

- Logische Beziehungen zwischen Klassen („sich-kennen“)

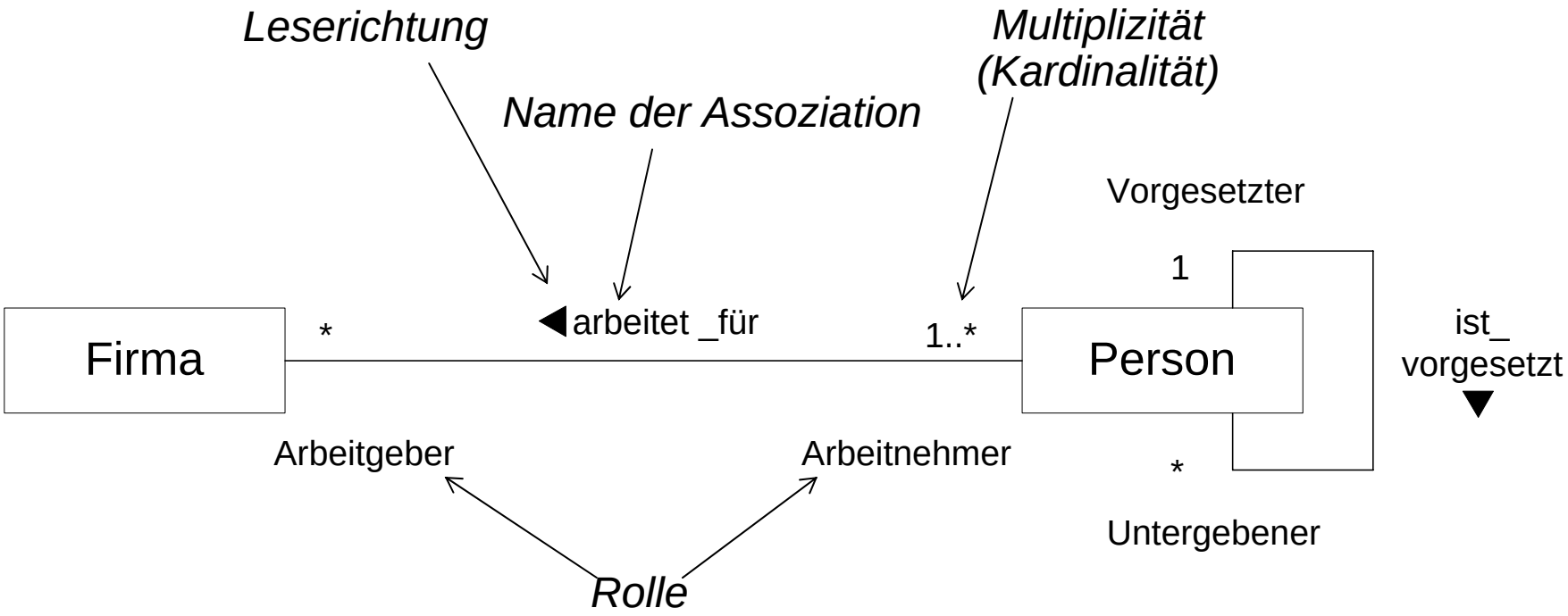
  ⇨ Verknüpfungsmöglichkeit zwischen Objekten dieser Klassen

- Definiert ebenso mögliche Kommunikationswege zwischen Objekten der Klassen.

- **Verbindung zwischen Objekten:**

  ⇨ Stets als Assoziation und nicht als Zeiger-Attribut modellieren! (Abstraktionsebenen)
  ⇨ Implementierung erst später auf niedrigerer Abstraktionsebene
  ⇨ Multiplizität kann angegeben werden
  ⇨ Leserichtung angeben, wenn sie von links⇨rechts abweicht

Assoziationen (Beispiel)



- *1 Firma (als Arbeitgeber) kennt 1..n Personen (als Arbeitnehmer)*
- *1 Person (als Arbeitnehmer) kennt bzw. arbeitet für 0..n Firmen (als Arbeitgeber)*
- *1 Person (als Vorgesetzter) kennt bzw. ist_vorgesetzt für 0..n Personen (als Untergebene)*
- *1 Person (als Untergebener) kennt 1 Person (als Vorgesetzten)*

Wie findet man Assoziationen?

a) Use Case Analyse und Dokumentenanalyse

Verben in Dokumenten (Beschreibungen, Dokumentationen, Spezifikationen Interview-Protokoll etc.)

Aber: Nicht alle Assoziationen sind für den zu modellierenden Problembereich wichtig!

b) Weitere Hinweise auf Assoziationen:

- Verwaltung von Dingen (Firma hat Kunden, Abteilung gliedert sich in Gruppen)
- Besitz (Motor besitzt Benzinpumpe, Auftrag besteht aus Positionen)
- Kommunikation zwischen Objekten

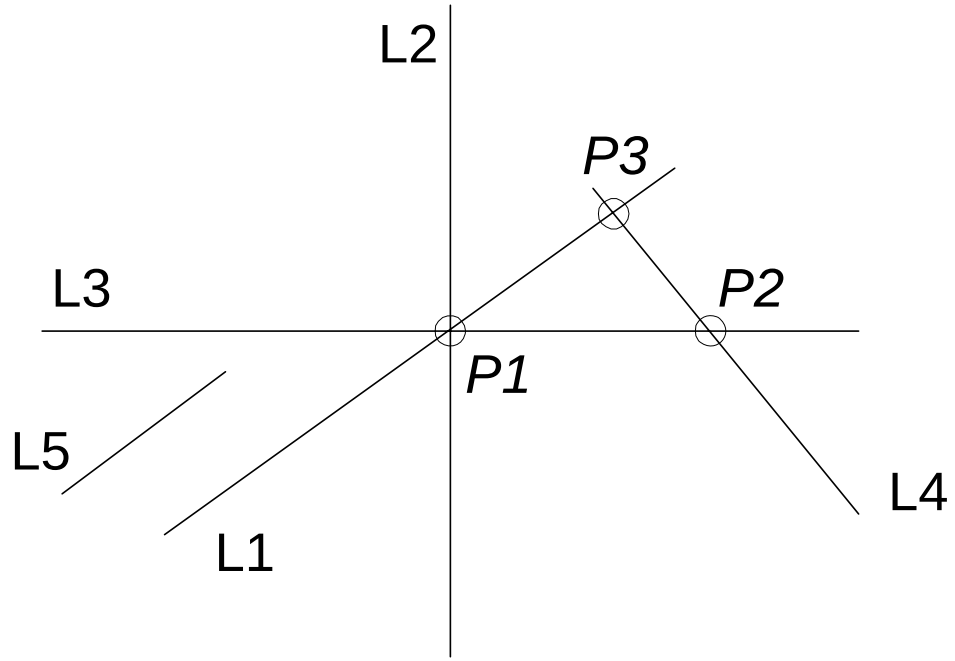
c) In späteren Schritten:

Bei dynamischer Modellierung wird Kommunikation zwischen Klassen dargestellt: Kommunikation läuft über vorhandene Assoziation – oder auch nicht

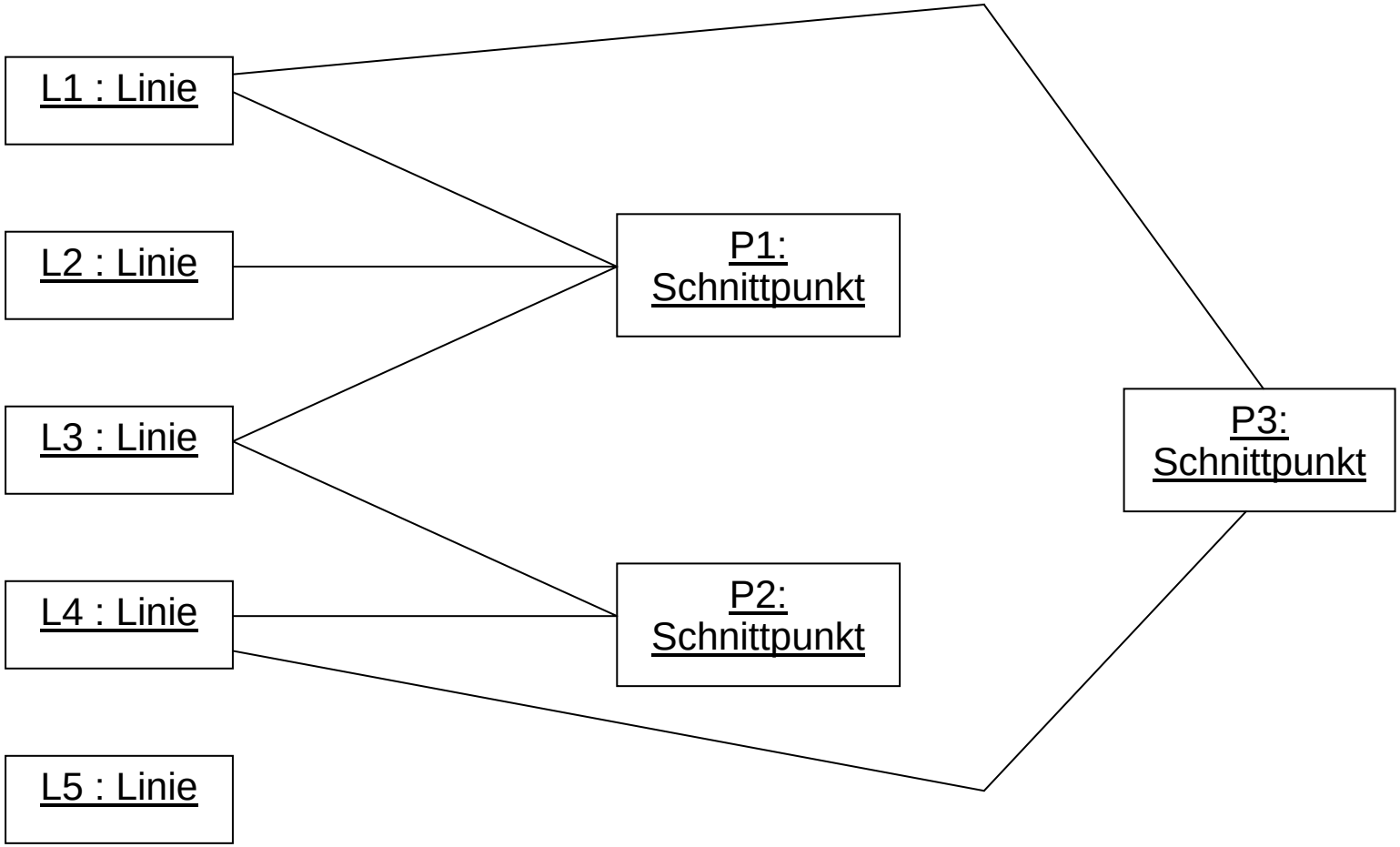
Falls nicht ⇨ zusätzliche Assoziation !

Assoziationen (Übung)

In der nachfolgenden Abbildung sind Linien gegeben, die bestimmte Schnittpunkte haben. Erstellen Sie ein Klassendiagramm mit den Klassen ´Schnittpunkt´ und ´Linie´ und einer Assoziation, die aussagt, dass Linien sich in einem Punkt schneiden.



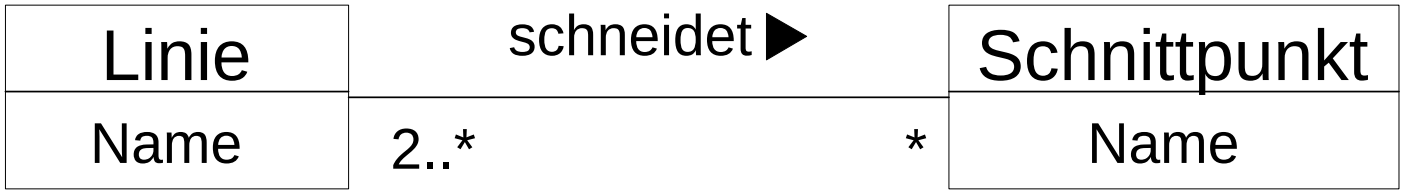
Assoziationen (Lösung – Vorarbeit durch Betrachtung der Objekte)



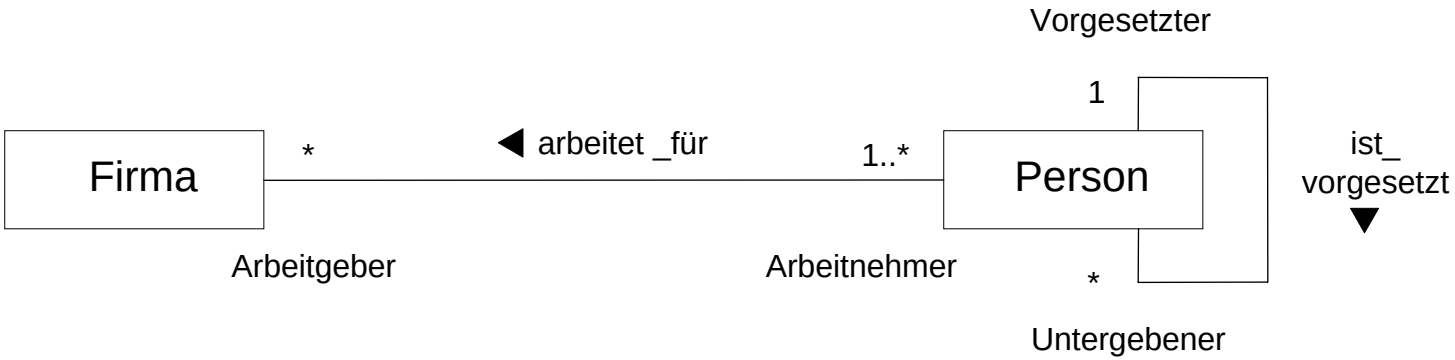
Assoziationen (Lösung - Klassendiagramm)

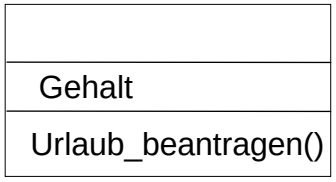
Erklärung:

Die Linien und Schnittpunkte sind Exemplare der Klassen ´Linie´ und ´Schnittpunkt´. Die Linien L1, L3 und L4 besitzen jeweils zwei Schnittpunkte mit anderen Linien (P1,P3 bzw. P1,P2 bzw. P3,P2), die Linie L2 schneidet zwei Linien im Punkt P1, die Linie L5 schneidet keinen Punkt.



Assoziationen





Betrachten Sie das Attribut *Gehalt* und die Methode *Urlaub_beantragen*

⇨ Wie und wo sollten diese Elemente modelliert werden?

Assoziationsklassen

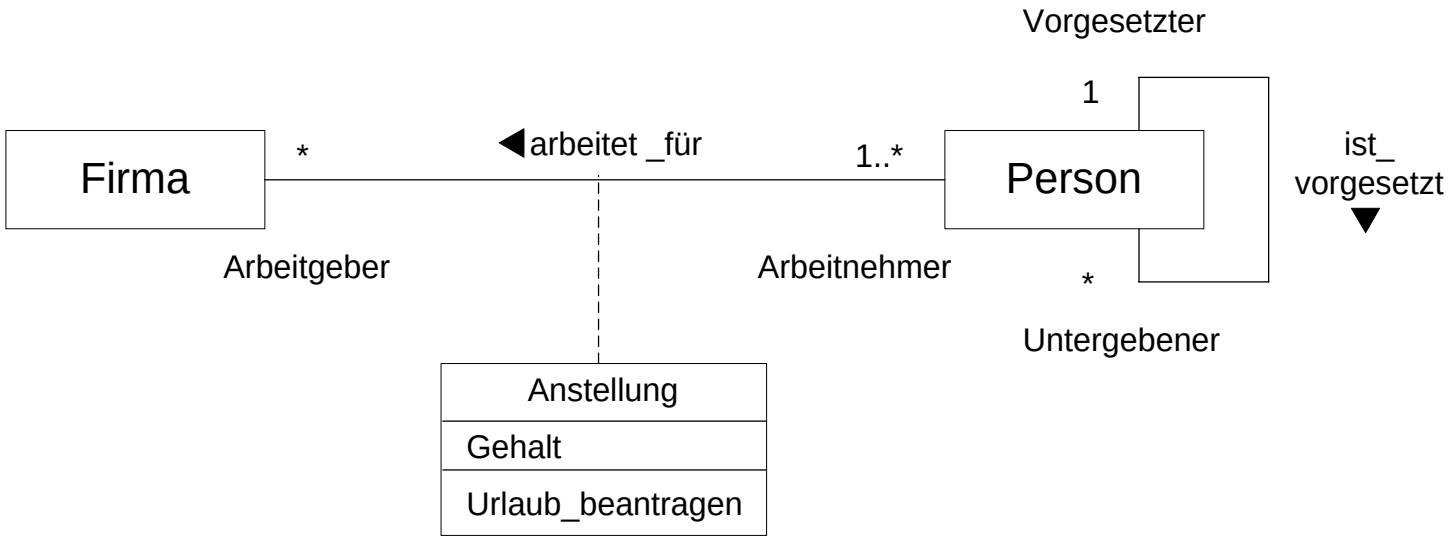
- auch Assoziationen können Merkmale, d.h. Attribute und Operationen besitzen.

- Diese Merkmale sind abhängig vom Vorhandensein der Assoziation. In anderem Kontext nicht nötig.

  $\Rightarrow$ Darstellung als Assoziationsklasse.

- Die Merkmale existieren für jede einzelne aufgebaute Verbindung zwischen zwei Objekten dieser Klassen genau einmal.

Beispiel für eine Assoziationsklasse



***Bem**.:* *Auch wenn jede Person in genau einer Firma arbeiten würde: Semantisch gehört Gehalt und Urlaub_beantragen zur Assoziation $\Rightarrow$ Assoziationsobjekt*

Durchgängiges Beispiel - Fortsetzung

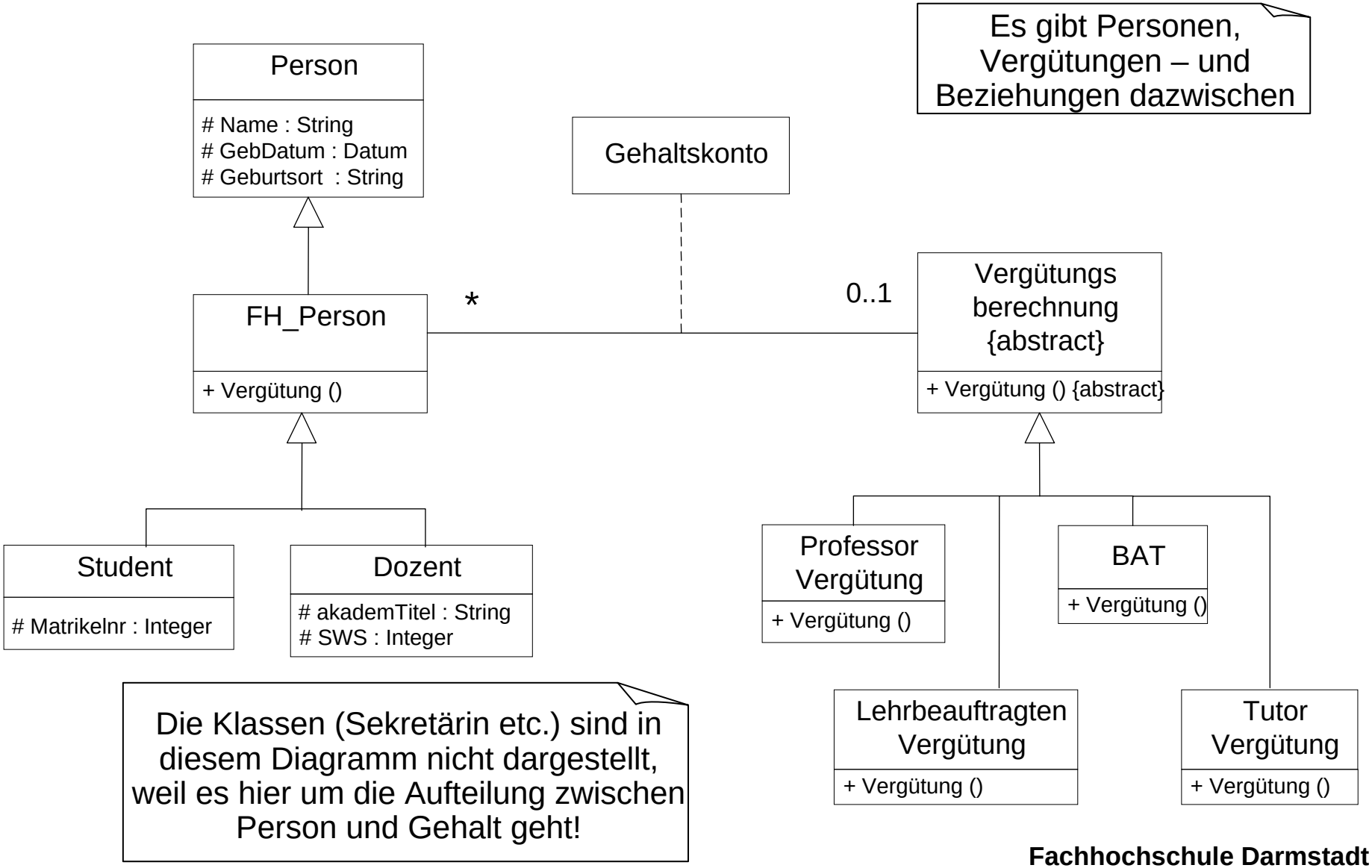
Die bisher modellierte Vererbungshierarchie ist korrekt, jedoch bezüglich Flexibilität und Wiederverwendung ungünstig modelliert:

- *Wird ein Objekt der Klasse Student erzeugt, so müsste dieses Objekt seine Klasse wechseln, sobald der Student Tutor wird. Nach der bisher modellierten Hierarchie müsste das Objekt zerstört und ein neues Objekt der Klasse StudentischerTutor erzeugt und mit denselben Daten belegt werden.*

- *Der Unterschied zwischen den Personengruppen (im Hinblick auf ein Hochschulverwaltungssystem) besteht in der Art der Vergütung.*
  Die Vergütung taucht aber nur "versteckt" als Methode auf.

⇨ *Nachdem wir Assoziationen und Assoziationsklassen kennengelernt haben, können wir durch ein geschicktes Design der bisherigen Klassenhierarchie dieses Problem beseitigen.*

Durchgängiges Beispiel (Lösung mit Assoziationen)

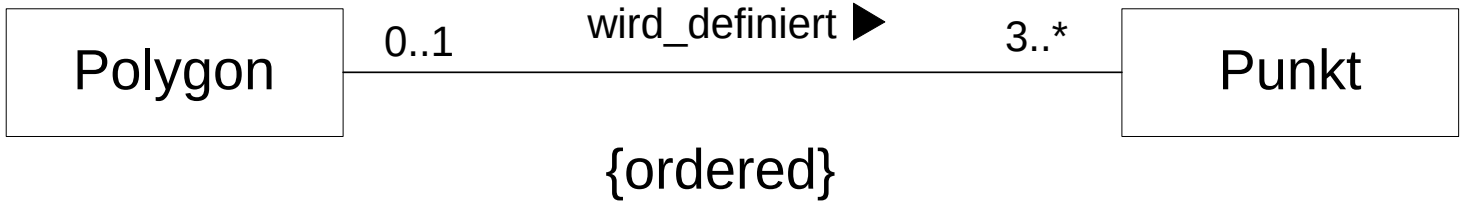


Randbedingungen für Assoziationen (Constraints)

Zu Assoziationen können Randbedingungen hinzudefiniert werden.

- Eine Randbedingung ist eine zusätzliche Information zum vorhandenen Modellelement, um es konsistent zu machen.
- Constraints geben Bedingungen für die Implementierungsphase.

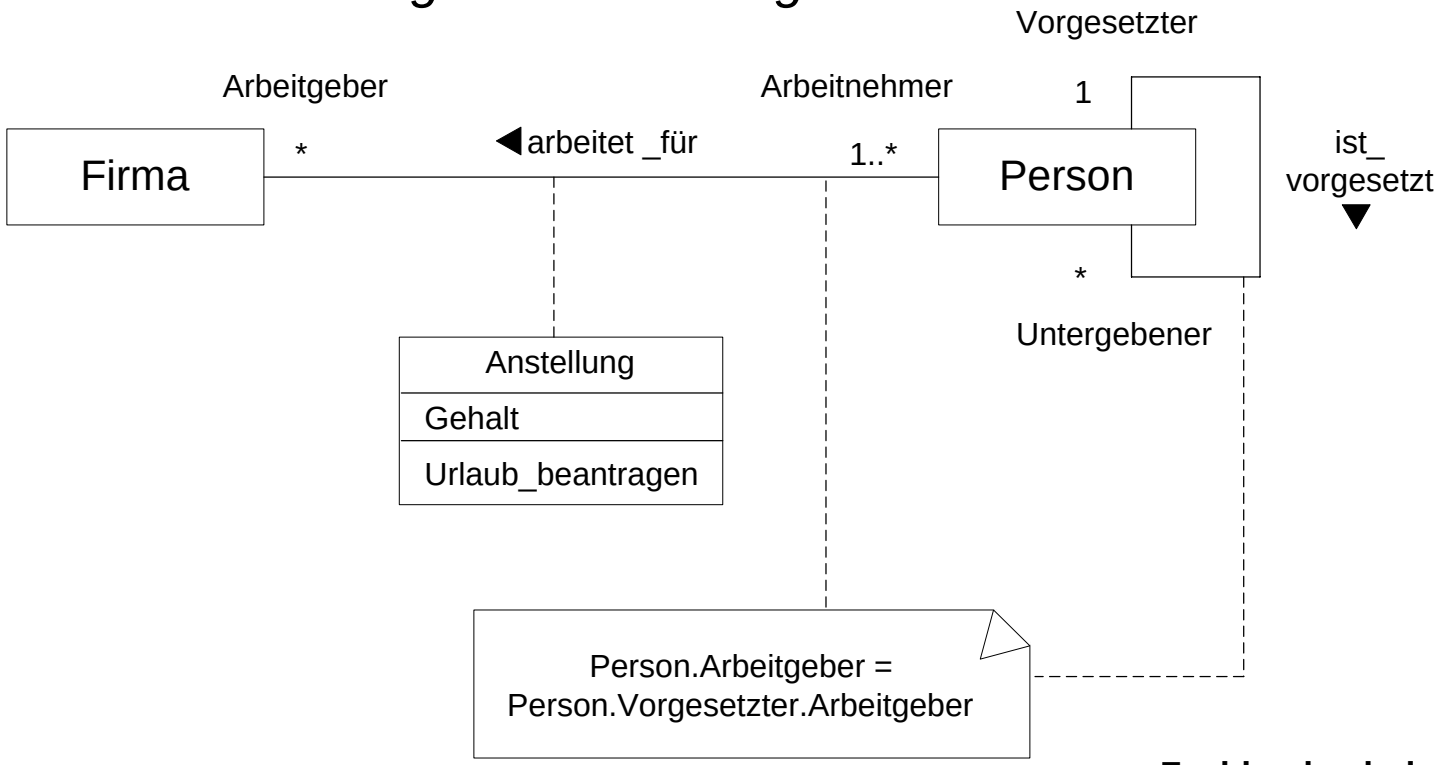
Beispiel 1:
Bei einem Polygon kommt es auf die Reihenfolge der Punkte an



Randbedingungen für Assoziationen (Beispiel)

Randbedingungen mit langem Text können als Kommentar angegeben sein.

z.B.: Der Vorgesetzte und dessen untergebener Arbeitnehmer müssen bei demselben Arbeitgeber beschäftigt sein.

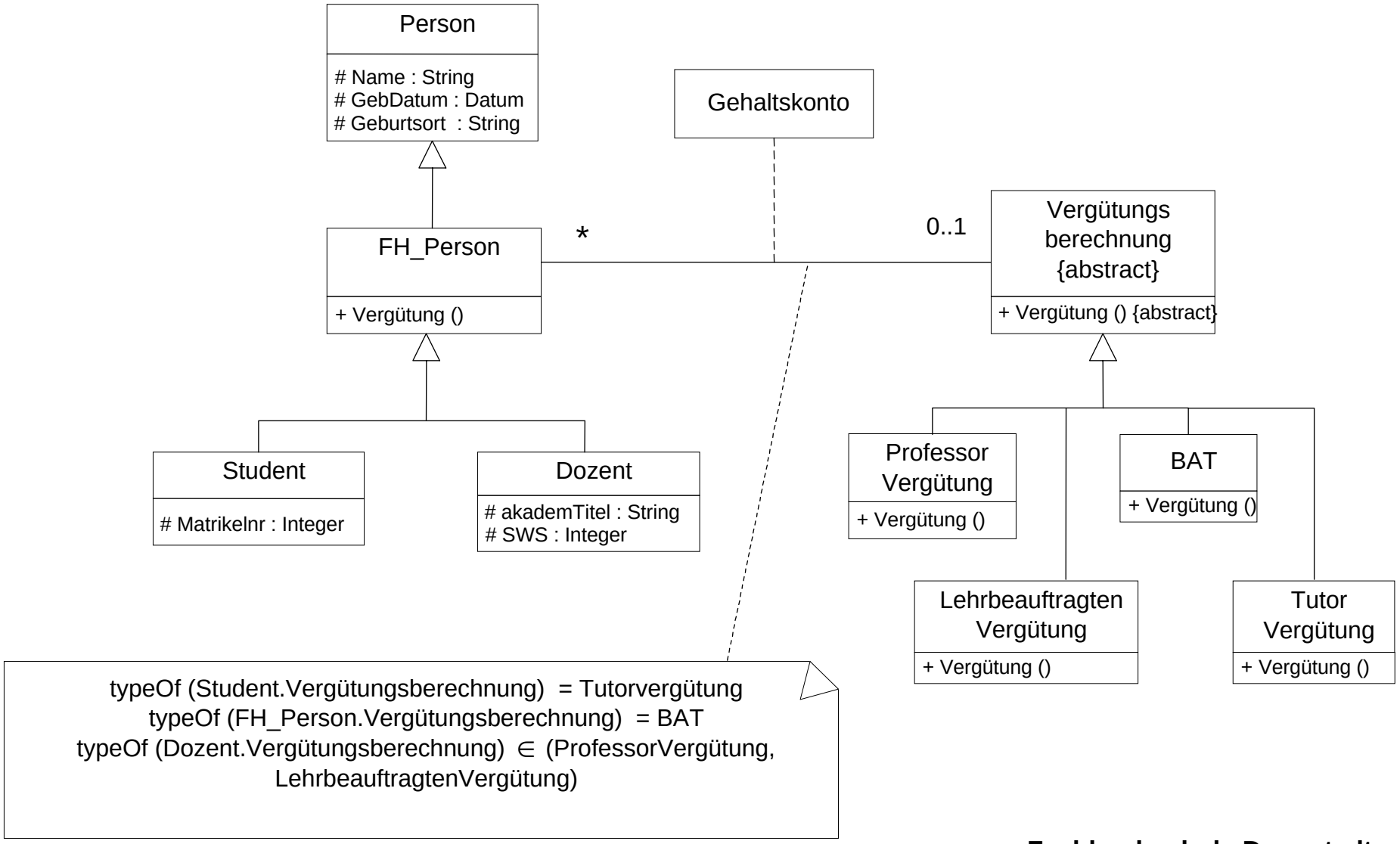


Durchgängiges Beispiel (mit Constraints)

- *Unsere bereits veränderte Vererbungshierarchie des ersten Klassenhierarchieentwurfs kann nun durch die Verwendung von selbstdefinierten Constraints vollständig und widerspruchsfrei modelliert werden.*

- *Es ist klar, dass nicht jedes Objekt willkürlich mit jedem Vergütungsalgorithmus kombiniert werden kann. Durch ein entsprechendes Constraint sind nun (gemäß der Aufgabe eines Modells) alle Regeln des Anwendungsbereichs widergespiegelt und hinreichende Vorgaben für die Implementierung gegeben.*

Durchgängiges Beispiel (Lösung mit Constraints)

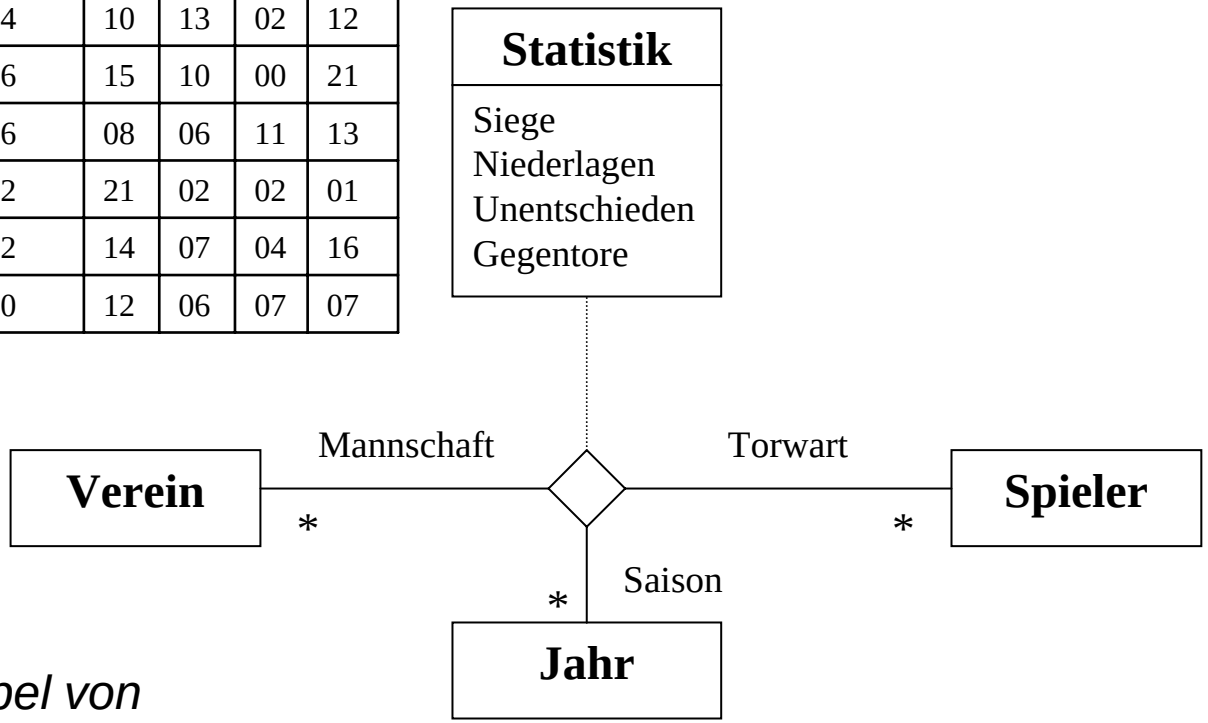


Ternäre und höherwertige Assoziationen: Beispiel

An einer Assoziation können auch mehr als zwei Klassen beteiligt sein.

Daten:

Torwart	Mannschaft	Saison	S	N	U	GT
Peter Gleichauf	FC Sandberg	1964	10	13	02	12
Peter Gleichauf	FC Sandberg	1966	15	10	00	21
Fred Hill	ASC Altheim	1966	08	06	11	13
Fred Hill	ASC Altheim	1972	21	02	02	01
Fred Hill	SV Rohrstock	1972	14	07	04	16
Fred Hill	FC Sandberg	1970	12	06	07	07



Zu einem assoziierten Tripel von

- *Spieler, Verein und Jahr gibt es genau 1 Statistik*
- *Statistik, Spieler und Verein gibt es 0..n Jahre*

Ternäre und höherwertige Assoziationen

- Oft gibt es Alternativen zur 3-er bzw. höherwertigen Assoziation (Einführung einer weiteren Klasse statt 3er-Beziehung)
- Oft sind vermeintliche 3-er Beziehungen in Wahrheit drei 2-er Beziehungen!

⇒ **3-er und höherwertige Assoziationen sollten, falls möglich, vermieden werden !**

Übung

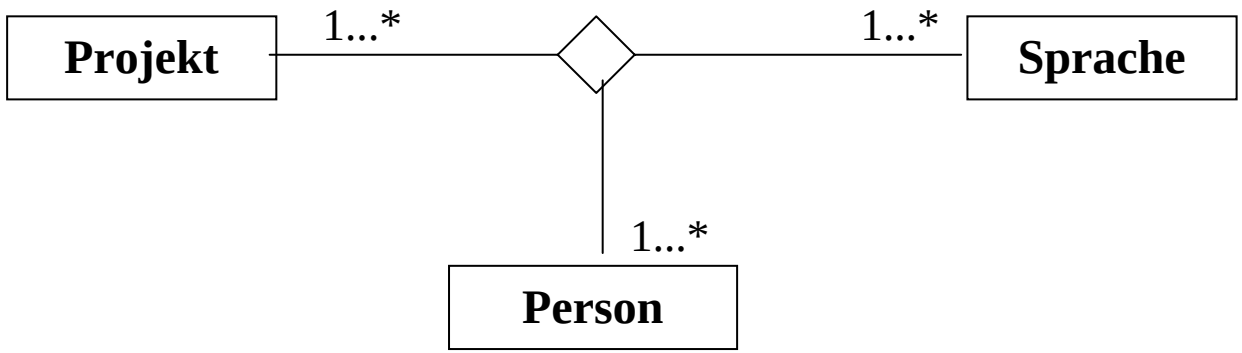
Anforderung:

- *In unserer Firma werden mehrere Projekte abgewickelt.*
- *In einem Projekt arbeiten mehrere Mitarbeiter mit.*
- *Das im Projekt zu erstellende Programmierer-System kann in mehreren Programmiersprachen zu schreiben sein.*
- *Ein Mitarbeiter arbeitet in diesem Projekt in einer oder mehreren Programmiersprachen.*
- *Der gleiche Mitarbeiter kann auch in anderen Projekten mitarbeiten, dort evtl. mit anderen Programmiersprachen.*

Aufgabe:

Erstellen sie das Klassendiagramm.

Übung: Klassendiagramm



- *zu einer Person in einem Projekt gibt es 1..n verwendete Sprachen*
- *zu einer Sprache in einem Projekt gibt es 1..n Personen*
- *zu einer Person, die in einer Sprache arbeitet, gibt es 1..n Projekte*

Qualifizierte Assoziationen

Zweck:

auf logischer Ebene einen Zugriffsschlüssel auf Objekte definieren

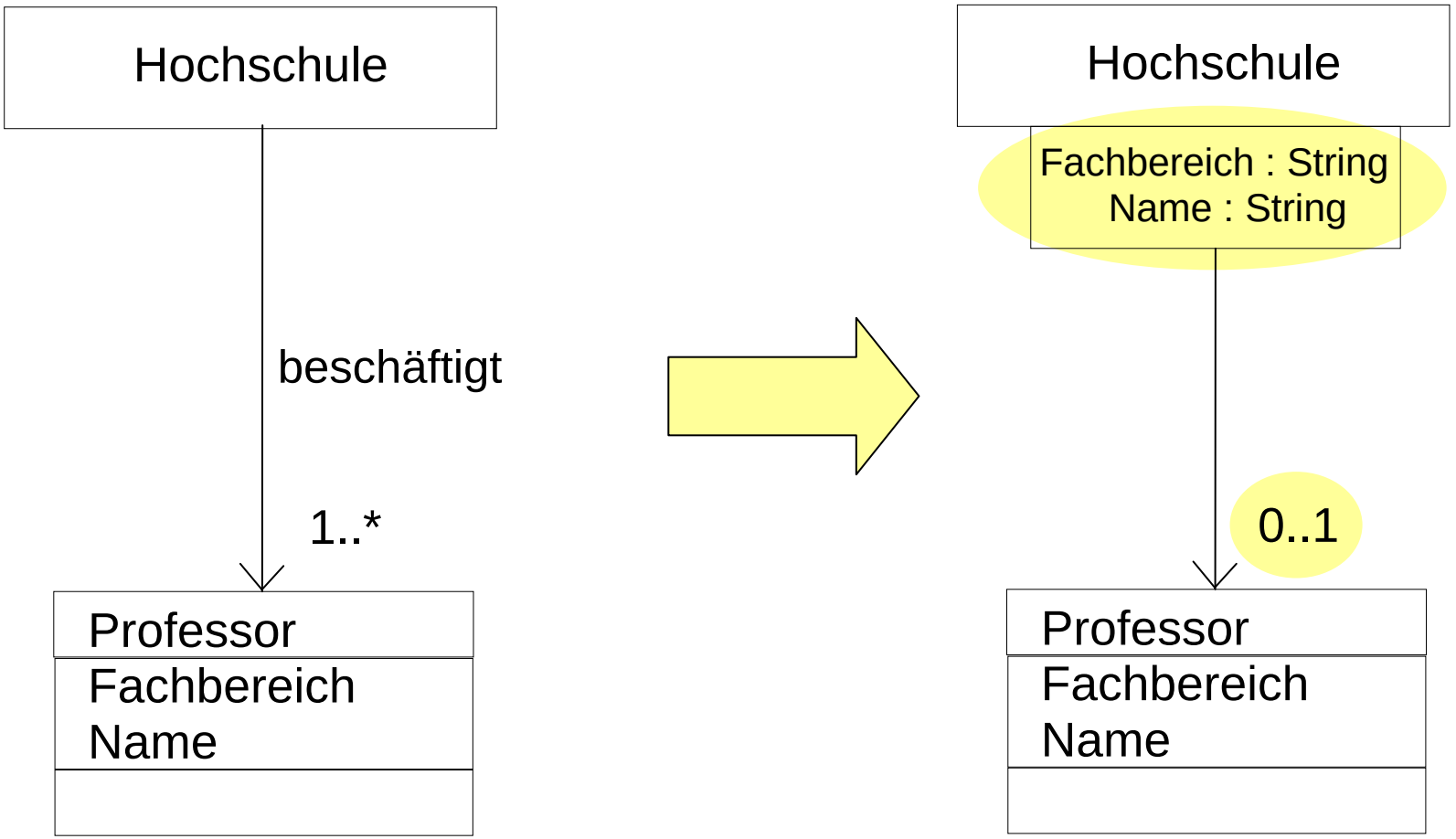
⇨ direkter Zugriff auf assoziierte Objekte
⇨ spezifiziert über ein oder mehrere Attribute (**Qualifier**) der Zielklasse
⇨ reduzierte Multiplizität für die Objekte, die über Qualifier festgelegt sind

Kardinalität:
- bezieht sich auf den Qualifier

⇨ „wieviele Objekte liefert ein spezifischer Wert des Qualifiers?“

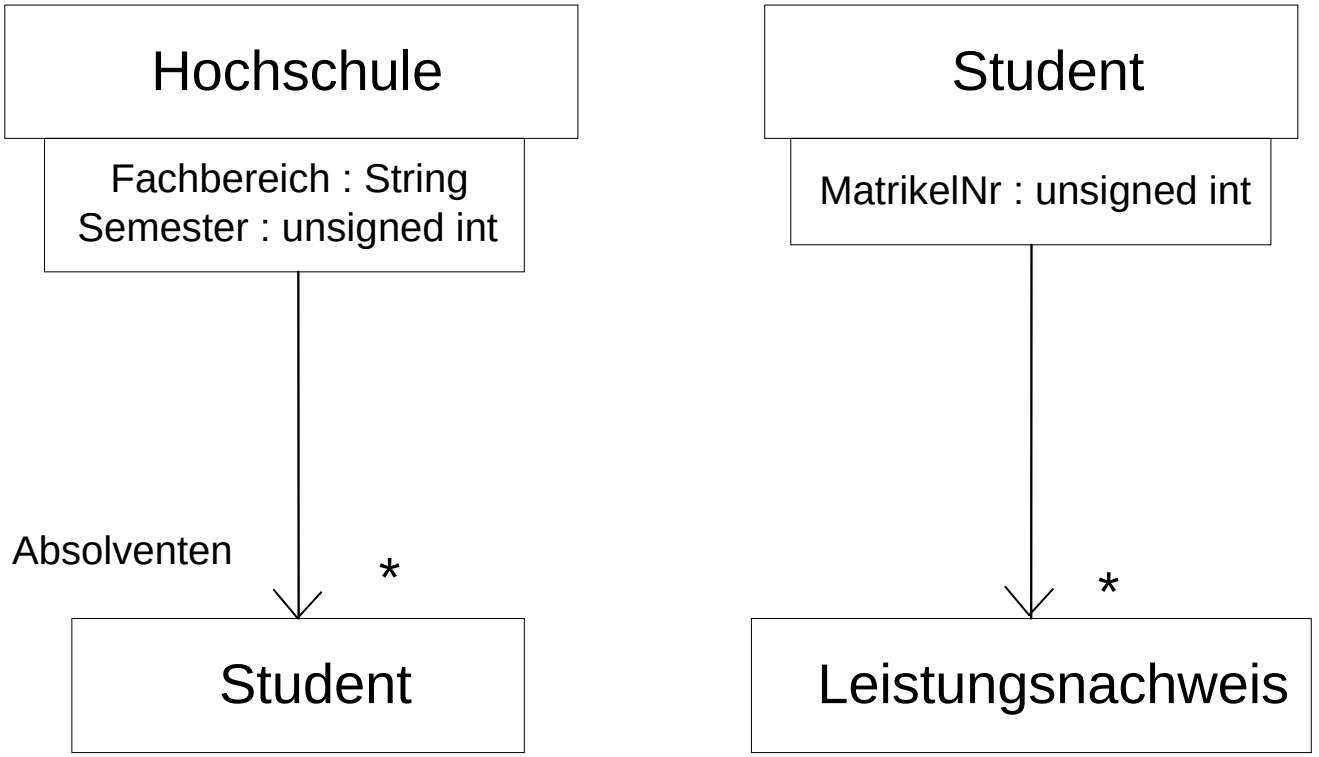
Qualifizierte Assoziationen (Beispiel)

Beispiel: Normale Assoziation und zweifach qualifizierte Assoziation



Qualifizierte Assoziationen (Beispiel)

Beispiel: Einfach und zweifach qualifizierte Assoziation:



Durchgängiges Beispiel

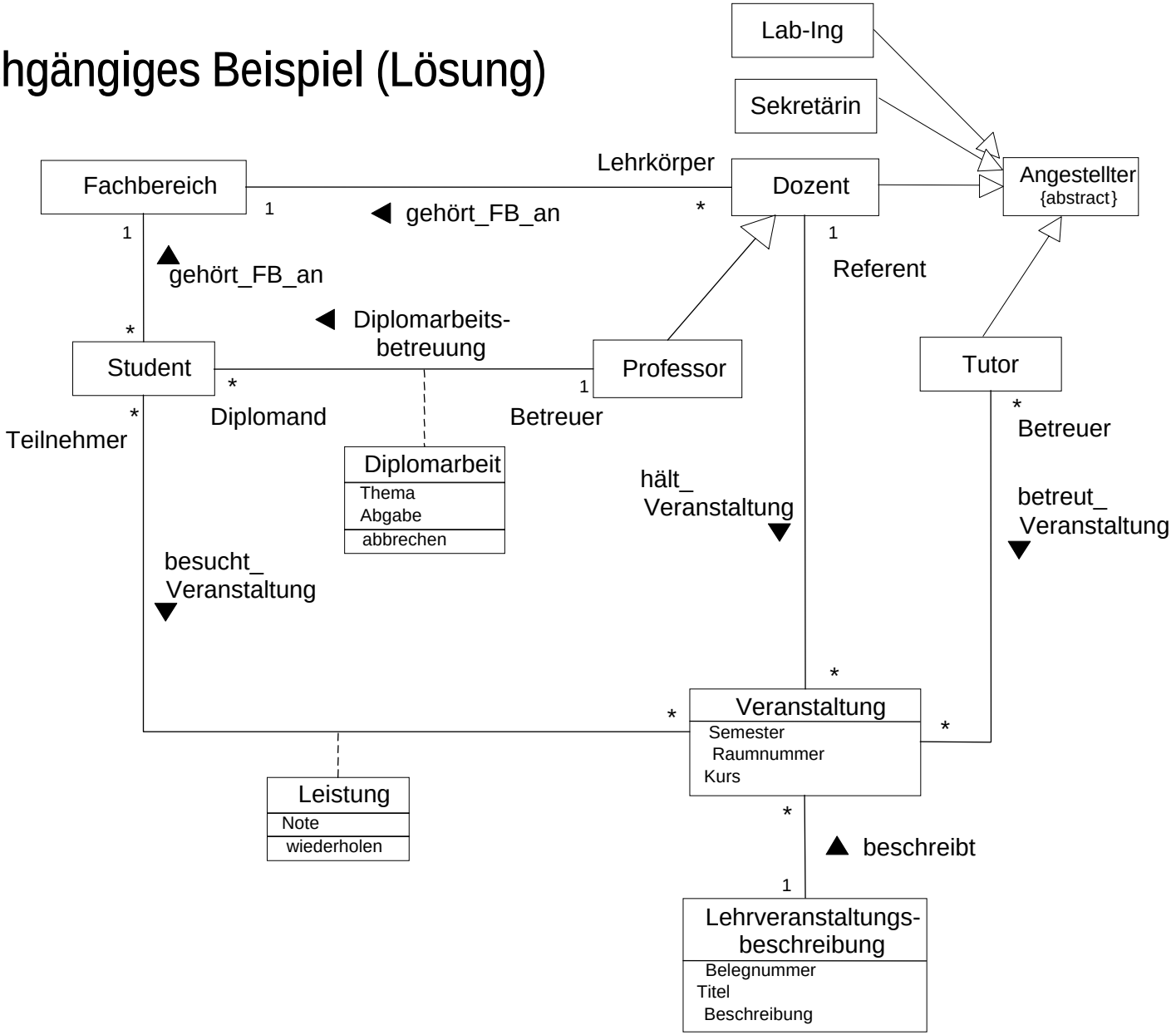
***Aufgabe**:*

- *Erweitern Sie das bisherige Klassendiagramm unseres Hochschulverwaltungssystems so, dass es die nachfolgenden zusätzlichen Klassen und Assoziationen integriert.*
- *Analysieren Sie zur Identifikation der Klassen und Assoziationen den Text. Bestimmen Sie die Kardinalitäten der Assoziationen und benennen Sie die Assoziationen und/oder die Rollen, die die beteiligten Klassen in einer Assoziation spielen, wenn dadurch die Verständlichkeit des Modells verbessert wird.*

***Anforderungen**:*

- *Die Dozenten und die Studenten gehören einem Fachbereich an.*
- *Studenten nehmen an Lehrveranstaltungen teil, die von Dozenten gehalten werden.*
- *Die Lehrveranstaltungen werden mit einem Leistungsnachweis abgeschlossen. Bestehen Studenten die Klausur nicht, können sie an einer Wiederholungsklausur teilnehmen.*
- *Studenten werden während der Diplomarbeit von Professoren betreut. Thema und Abgabetermin der Diplomarbeit sind zu erfassen. Studenten können natürlich eine Diplomarbeit auch unvollendet abbrechen.*
- *Lehrveranstaltungen im Semester können von Tutoren mitbetreut werden.*

Durchgängiges Beispiel (Lösung)



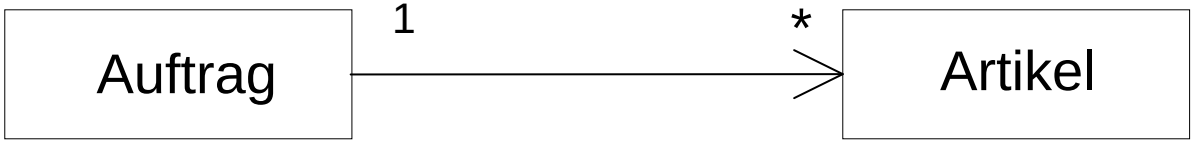
Navigationsrichtungen bei Assoziationen

- Bei der Modellierung des dynamischen Verhaltens (Entwurf der Kommunikation zwischen Klassen) wird festgestellt, in welche Richtung eine Assoziationsbeziehung durchlaufen wird.

  $\Rightarrow$ Diese Navigationsrichtung kann angegeben werden (offene Pfeilspitze)

- Die Navigationsrichtung gibt beim späteren Programmentwurf Auskunft darüber, in welchen Klassen Verweise auf Objekte anderer Klassen angelegt werden müssen.

  Beispiel:



  ⇨ *Ein Auftrag muss seine assoziierten Artikel kennen (z.B. Gesamtsumme berechnen, Nachschauen was disponiert werden muss).*

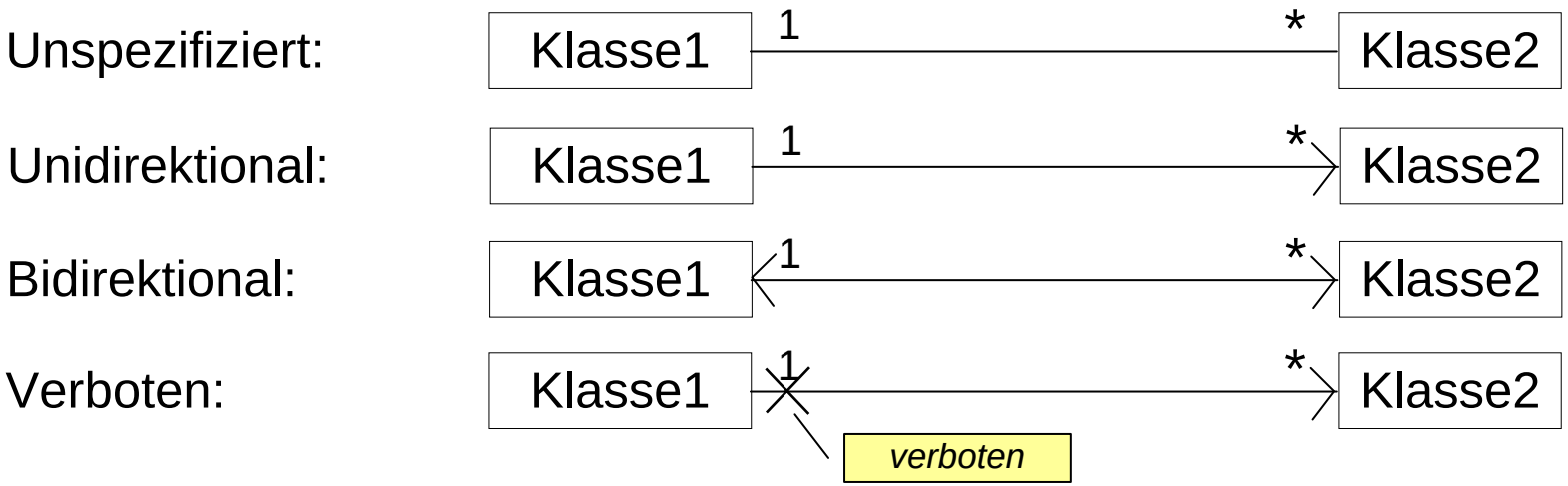
  ⇨ *Ein einzelner Artikel muss jedoch nicht wissen, welche Kunden ihn jemals bestellt haben.*

Notation bei Navigationsrichtungen

UML 2 !!
geändert ggü. UML 1.x

- In UML 1.x stand die Assoziation in Linienform für eine bidirektionale Beziehung
  - ⇨ es gab keine Möglichkeit auszudrücken, dass man sich über die Richtung "noch keine Gedanken gemacht" hat

- In UML 2.0:
  - ⇨ Notation für Navigierbarkeit und "unspezifiziert" (default)



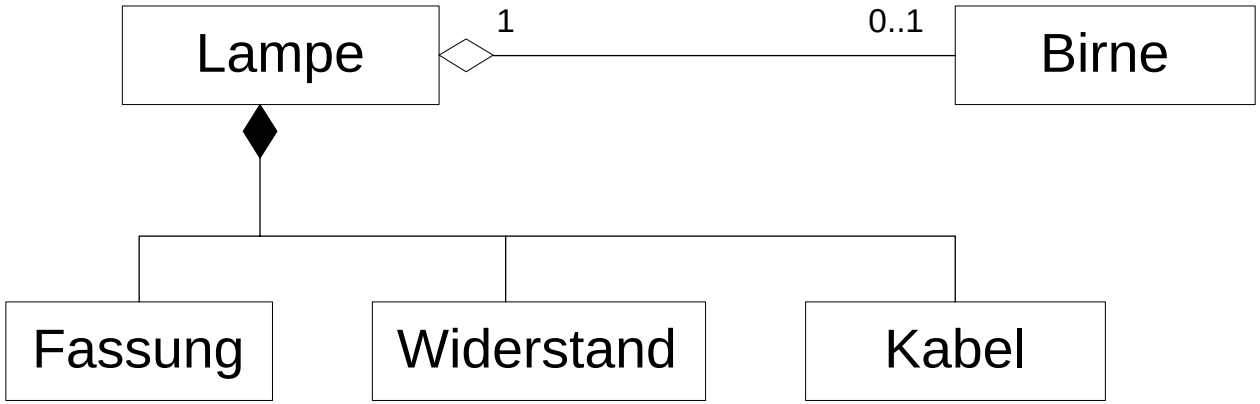
Aggregation (= spezielle Assoziation)

⇨ Aggregation bedeutet „Teil-von-Beziehung“ (oder „besteht-aus-Beziehung“) d.h. ein Objekt ist Bestandteil eines anderen.

Aggregation ist

transitiv: A ist Teil von B, B ist Teil von C. Somit ist A Teil von C.

asymetrisch: A ist Teil von B, aber B ist nicht Teil von A.

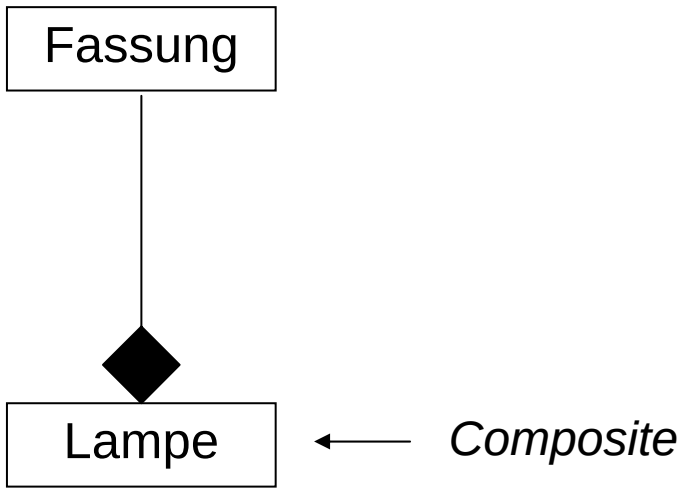


Arten von Aggregationen: Komposition

Komposition

- Symbol: ausgefüllte Raute
- Ist die Verbindung aufgebaut, kann sie nicht mehr verändert werden.
- Ein aggregiertes Objekt existiert während der Lebenszeit des aggregierenden Objekts (Composite-Objekt), aber nicht darüber hinaus.
- Ein aggregiertes Objekt gehört zu genau einem aggregierenden Objekt

⇨ "unshared" Aggregation

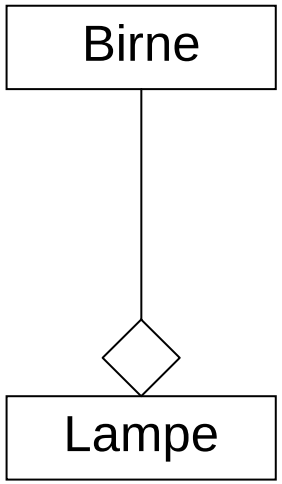


Arten von Aggregationen: Aggregation

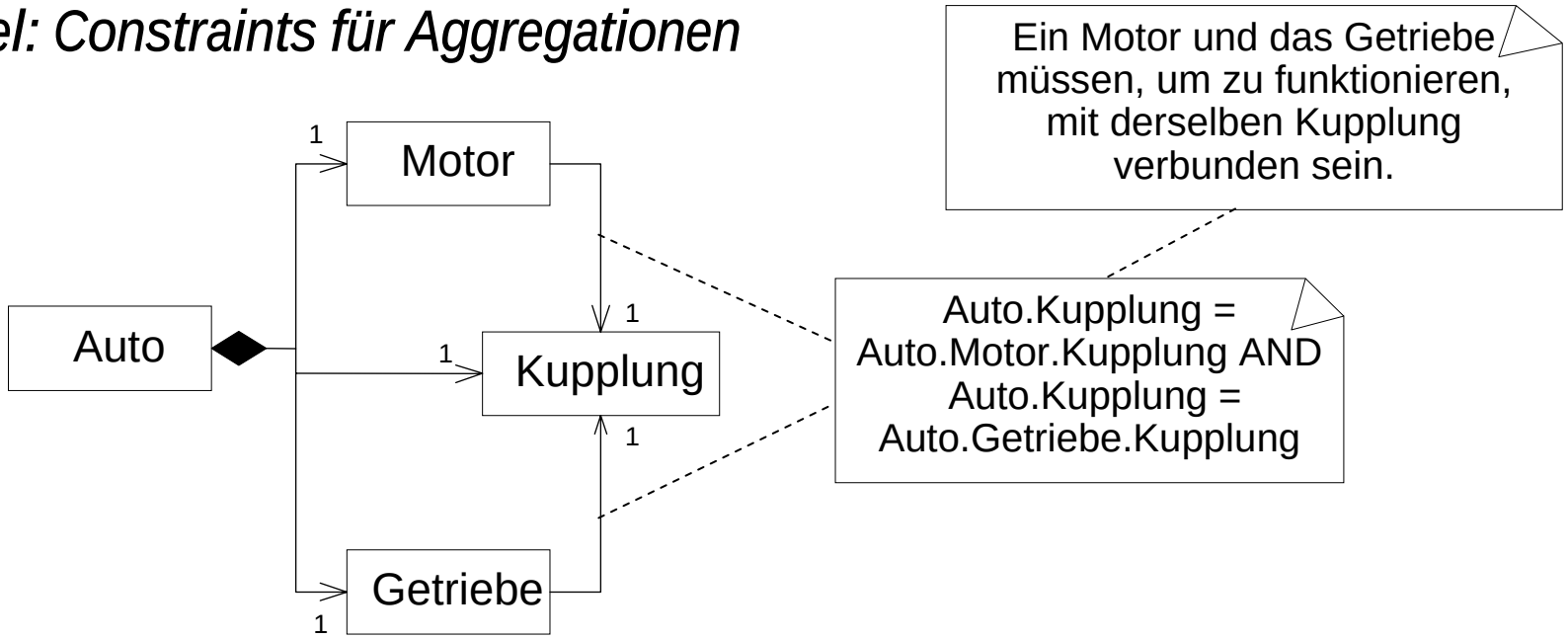
Aggregation:

- Symbol: leere Raute
- spezifiziert Referenz auf das aggregierte Objekt.
- Aggregiertes Objekt ist zwar Bestandteil, existiert aber unabhängig vom aggregierenden Objekt.

⇨ "shared" Aggregation



Beispiel: Constraints für Aggregationen



- *Ohne ein Constraint könnte laut Modell der Motor sowie das Getriebe ein anderes Kupplungsexemplar assoziieren als das übergeordnete Auto selbst*

  $\Rightarrow$ *Inkonsistenz im Modell !*

- *Das hinzudefinierte Constraint schließt diese Lücke*

Durchgängiges Beispiel mit Aggregationen

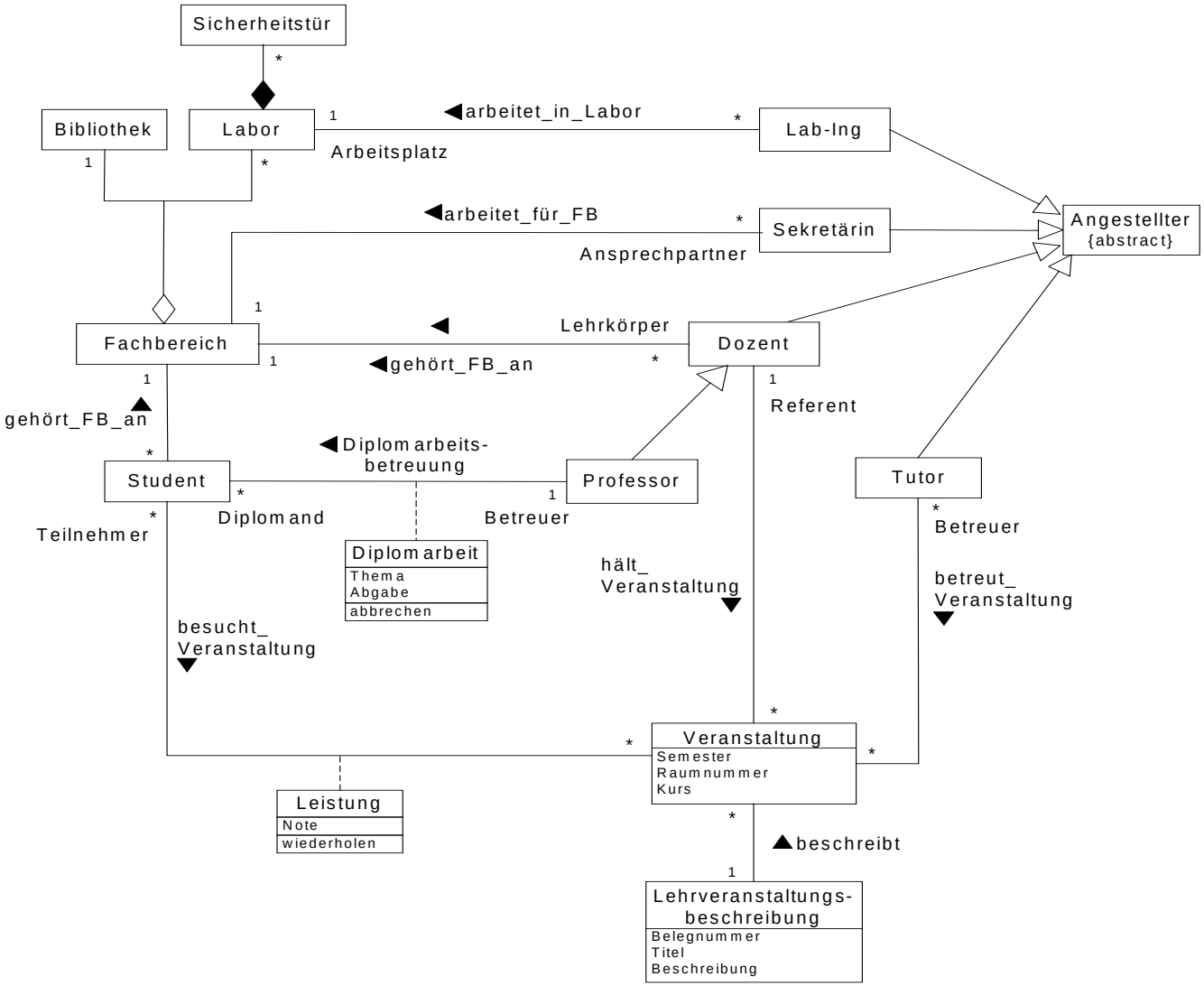
Aufgabe:

Erweitern Sie das Klassendiagramm der Hochschulverwaltung um die nachfolgenden Inhalte.

***Anforderungen**:*

- *Ein Fachbereich kann eine Bibliothek und mehrere Labore besitzen, in denen Laboringenieure arbeiten.*
- *Jeder Fachbereich beschäftigt in der Verwaltung Sekretärinnen.*
- *Die Laborräume sind zur Vermeidung von Diebstählen mit Sicherheitstüren ausgestattet.*
- *Labor-Ingenieure arbeiten in bestimmten Labors.*

Durchgängiges Beispiel mit Aggregationen (Lösung)



Durchgängiges Beispiel (Erklärung)

- *Teile eines Fachbereichs sind Labors und eine Bibliothek, die aber jederzeit einem anderen Fachbereich zugewiesen werden können. Daher ist die Teil-von-Beziehung eine shared-Aggregation.*

- *Eine Sicherheitstür ist fester Bestandteil des Labors und ist deswegen eine unshared-Aggregation.*

Zusammenfassung Klassen(diagramme)

Klassendiagramme enthalten

- Klassen mit Attributen und Methoden
- Vererbung / Generalisierung / Spezialisierung
- Abstrakte Klasse
- Parametrisierte Klasse
- Diskriminator
- Assoziationsklassen
- Constraints
- Assoziationen
- Höherwertige Assoziationen
- Qualifizierte Assoziationen
- Navigationsrichtung
- Aggregation

⇨ Das sind die Elemente im "Baukasten"

Wert von Klassendiagrammen

- Klassendiagramme sind eines der wichtigsten Beschreibungsmittel in der Modellierung
- Zur Beschreibung eines (komplexen) Systems werden viele Klassendiagramme erstellt

- Hinweise:
  - niemals versuchen gleich alles perfekt zu machen; auch Klassendiagramme werden iterativ entwickelt
  - Details wie Navigationsrichtungen und Multiplizität erst eintragen, wenn die Modellierung hinreichend stabil scheint
  - Bei der Modellierung an die Systemgrenze denken! Nur das System modellieren – nicht die komplette Außenwelt!
  - immer nur einen Aspekt auf einem Diagramm darstellen; wenn das Diagramm nicht mehr auf eine Seite passt – aufteilen!
  - die Begriffe für alle Elemente mit großer Sorgfalt wählen
  - Assoziationen so konkret wie möglich spezifizieren
  - ein Pointer als Attribut einer Klasse ist meist eine falsch modellierte Assoziation
  - keine Pseudoprogramme in Constraints schreiben – es geht um Verständlichkeit