

Software Engineering 1



Anna Prenzel

Einführung

Ziele der heutigen Veranstaltung:

- ✎ Kenntnisse zur Erstellung von Klassendiagrammen wiederholen und vertiefen.
- ✎ Erstellung und Interpretation von Klassendiagrammen üben.

Wiederholung

- ⌘ Was ist der Unterschied zwischen einem Analysemodell und einem Entwurfsmodell?
- ⌘ Nennen Sie zwei Aufgaben des Analysemodells
 - Validierung von Anforderungen mit Stakeholdern
 - Vorlage für den Softwareentwurf (Problemzerlegung)

Wiederholung

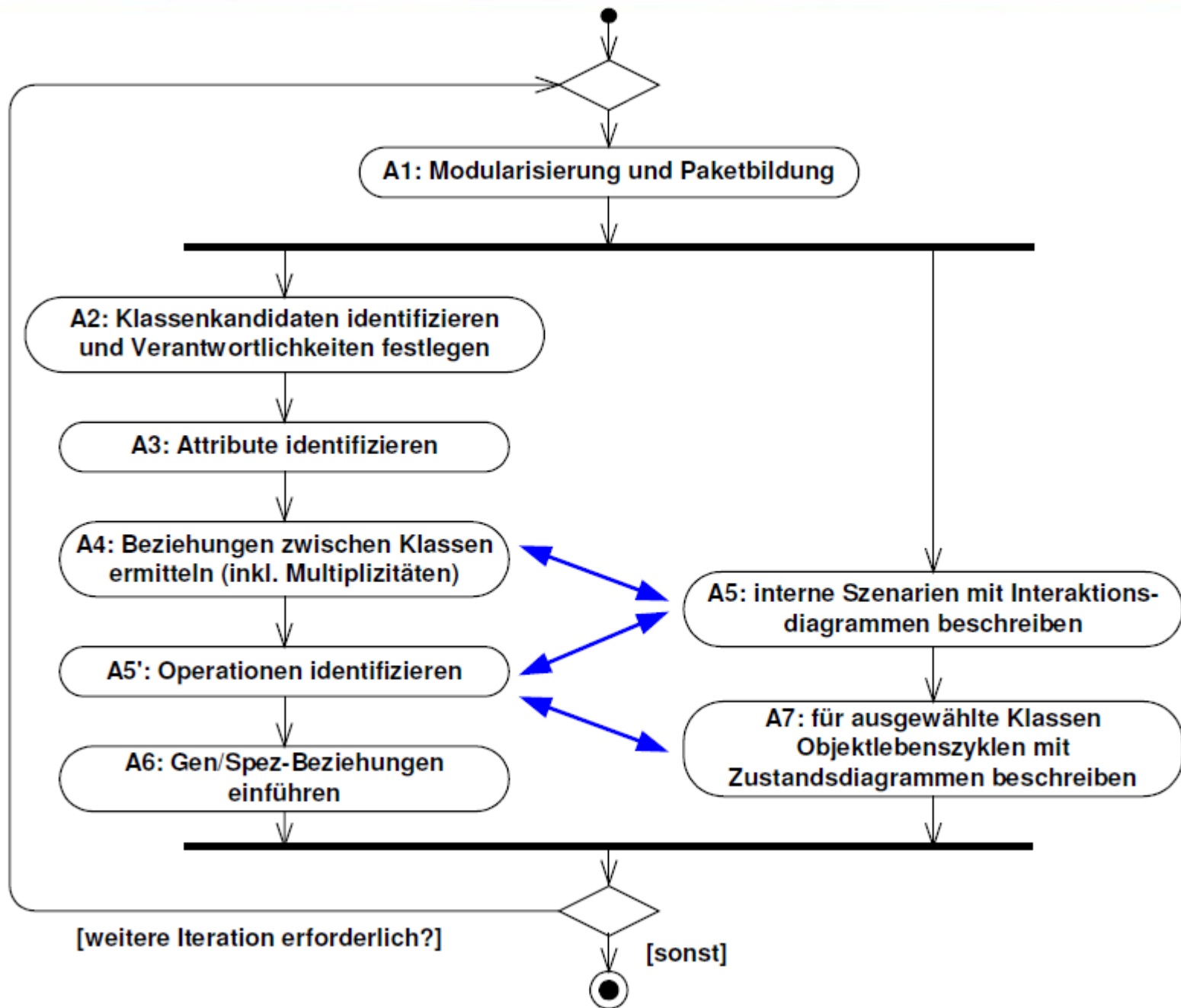
☞ Nennen Sie Vorteile des Einsatzes von Modellen in der Analyse

- Formale Notation ist eindeutig und kann nicht unterschiedlich ausgelegt werden (im Gegensatz zu natürlicher Sprache).
- Modelle geben vor, welche Zusammenhänge der Anwendungsdomäne analysiert werden müssen bzw. für die Lösung relevant sind (Abstraktion)
- Modelle erleichtern die Überprüfung der Vollständigkeit der Analyse (fehlende Multiplizitäten oder Assoziationen fallen auf)

Wiederholung

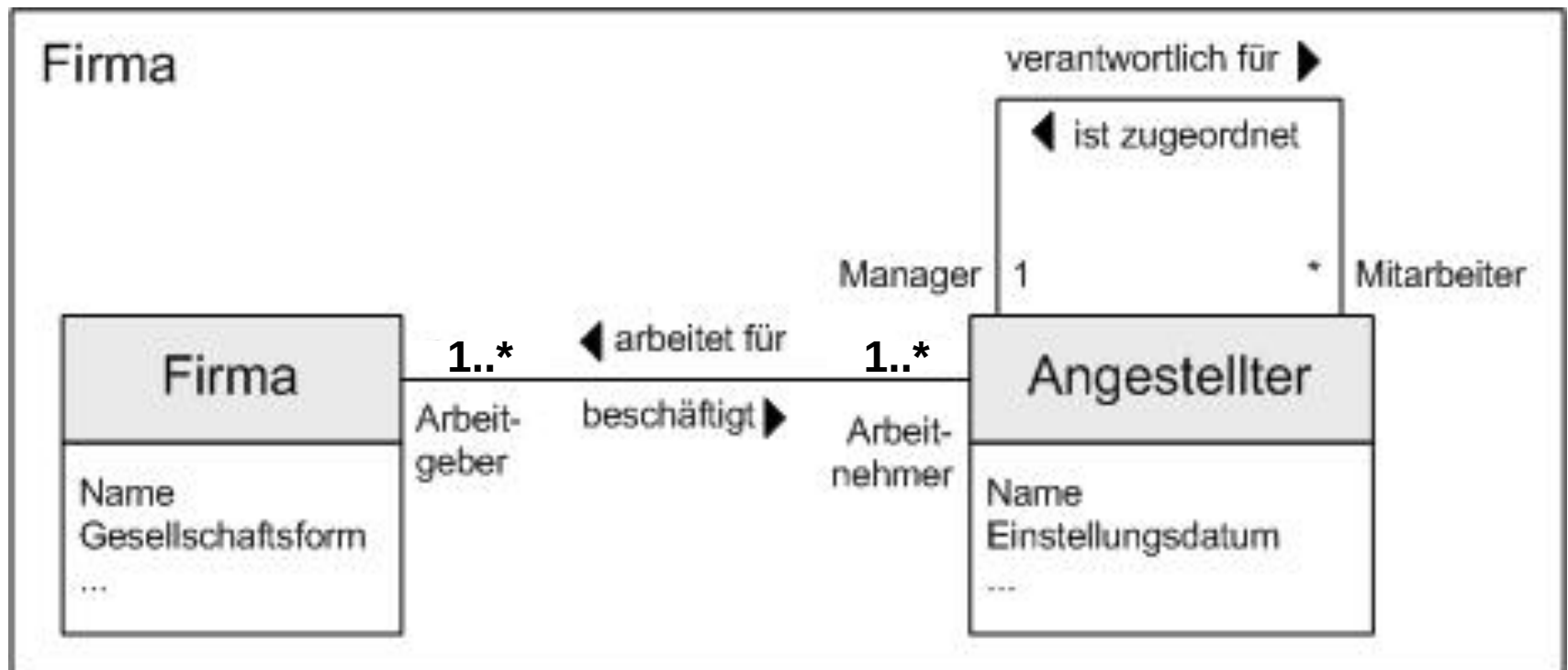
☞ Nennen Sie die Schritte A1 bis A4 des Vorgehensmodells für OOA

1. Modularisierung und Modellbildung
2. Klassenkandidaten identifizieren
3. Attribute identifizieren
4. Beziehungen zwischen Klassen ermitteln



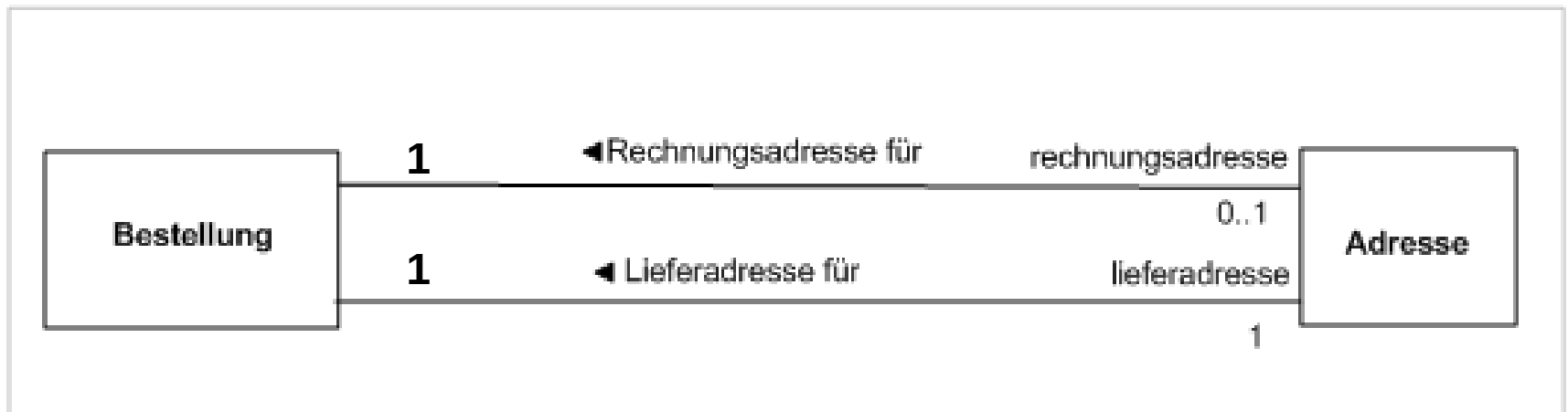
Klassendiagramme

☞ Erklären Sie die Modellelemente!

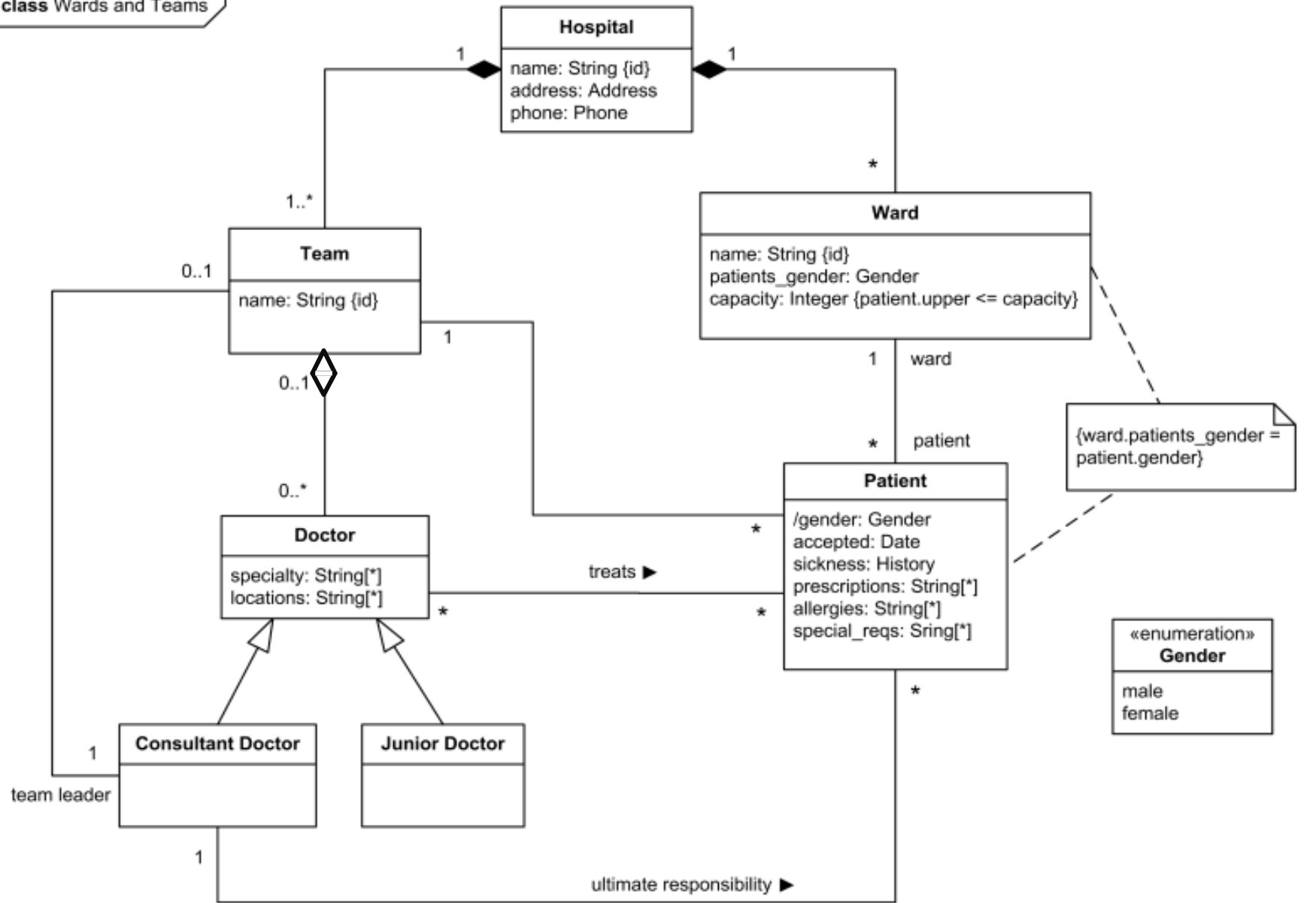


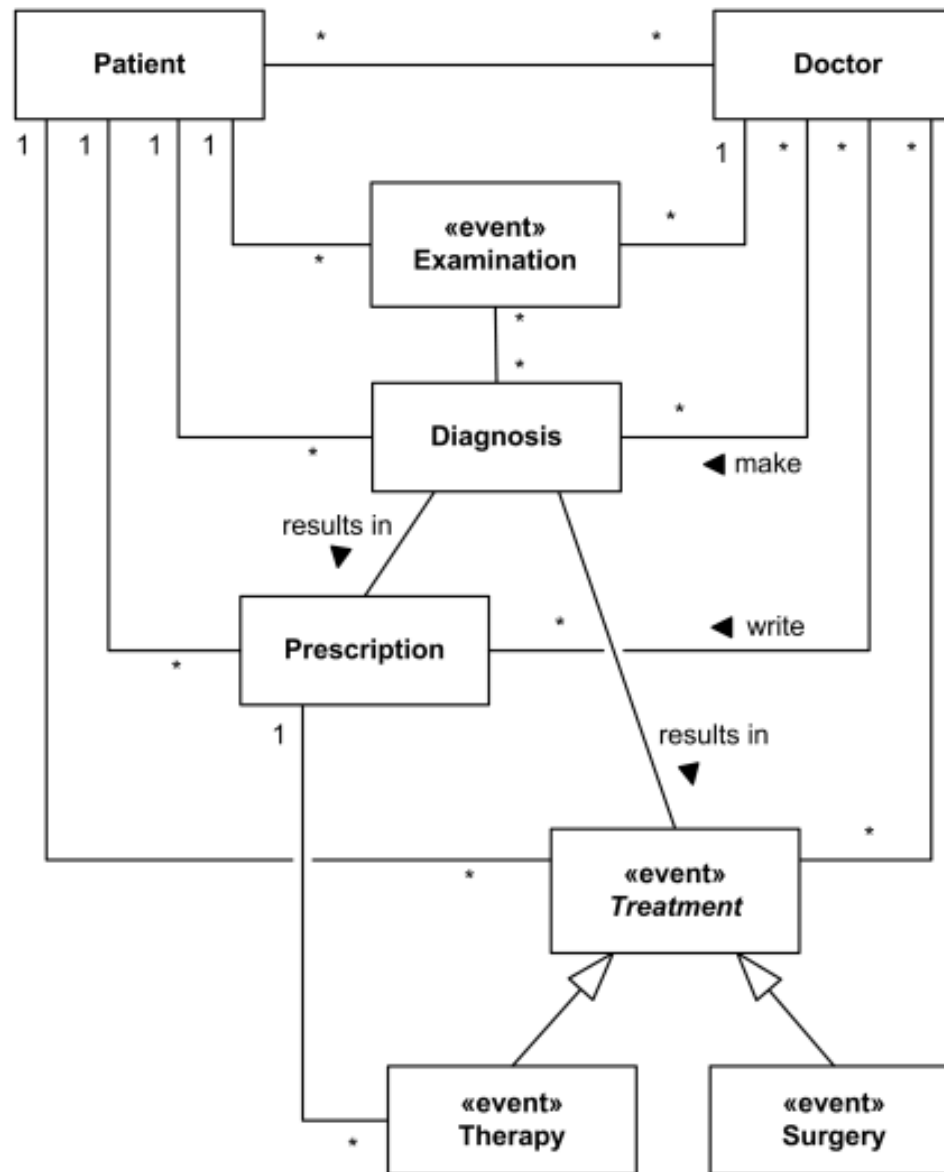
Klassendiagramme

☞ Erklären Sie das Modell!



class Wards and Teams





Klassendiagramme

Bitte beachten

- ✎ Assoziationen bestehen zwischen Objekten, und nicht zwischen Klassen
- ✎ Im Klassendiagramm werden mögliche Assoziationen dargestellt → Vorlage für Assoziationen
- ✎ Multiplizität legt die Grenzen für die Anzahl der Objekte in einer Beziehung fest, d.h. für die Kardinalität

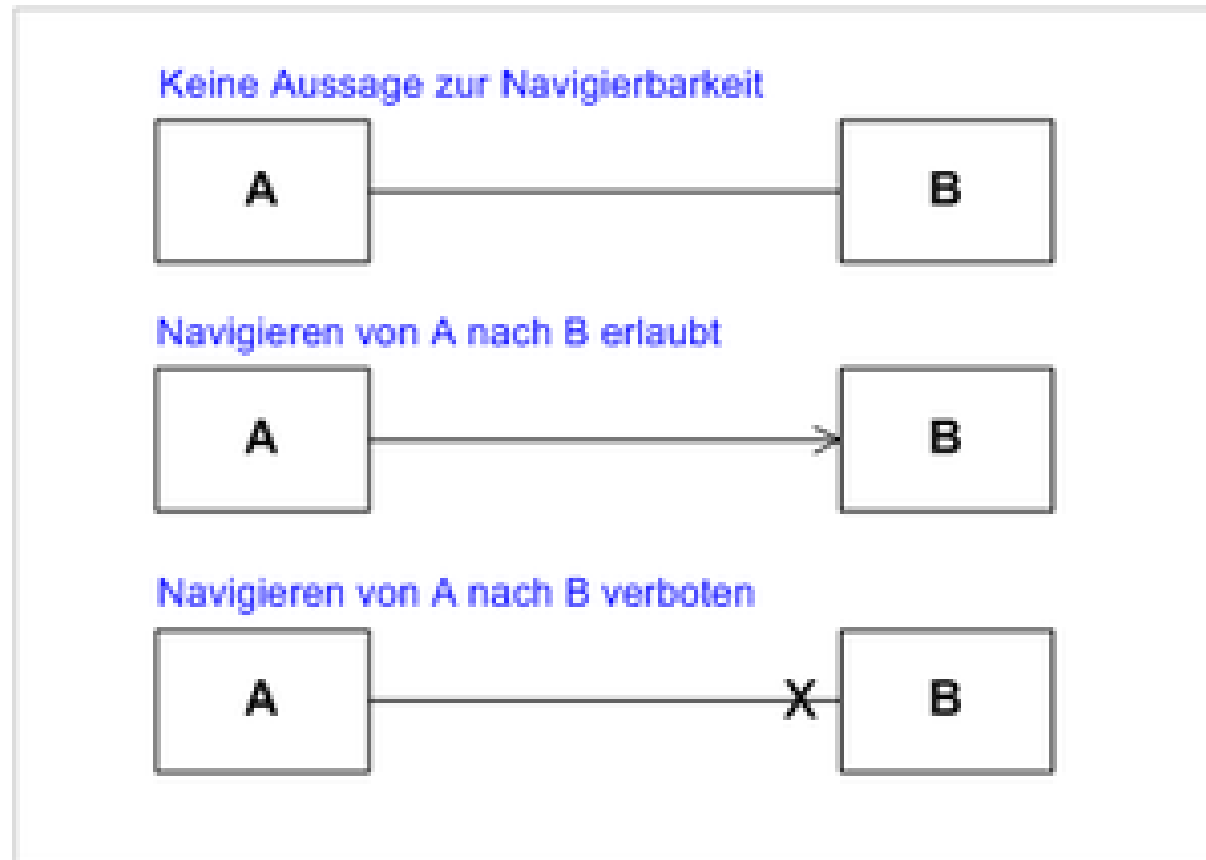
Klassendiagramme

Multiplizitäten

- ✎ **1** → genau ein Objekt ist an der Beziehung beteiligt
- ✎ **0..1** → max. ein Objekt ist an der Beziehung beteiligt
- ✎ **0..*** → beliebig viele Objekte sind an der Beziehung beteiligt
- ✎ **1..*** → min. ein Objekt ist an der Beziehung beteiligt
- ✎ **2..4** → zwischen zwei und vier Objekte sind an der Beziehung beteiligt

Klassendiagramme

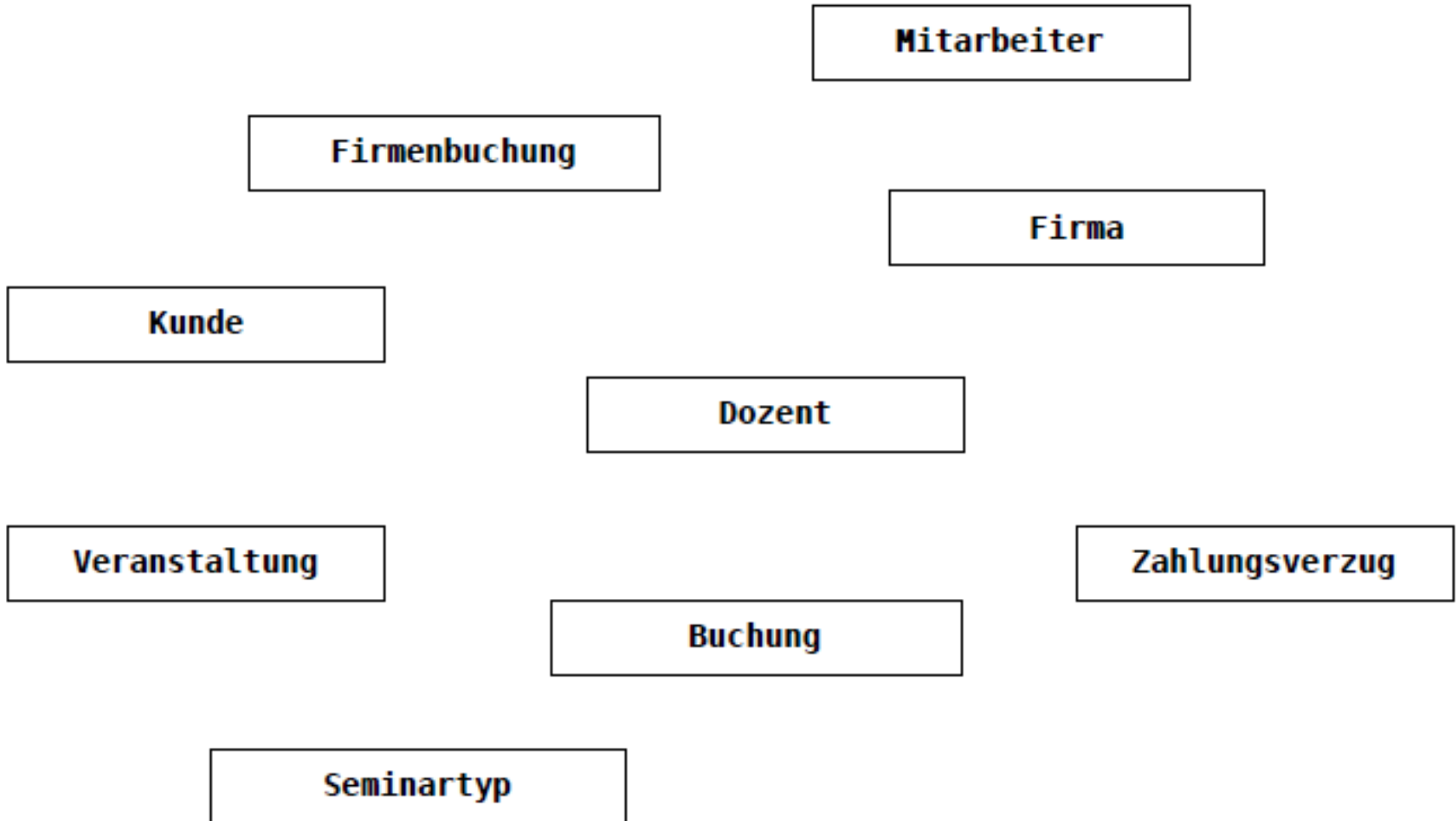
Navigierbarkeit



Klassendiagramme

**Setzen Sie das
dargestellte
Klassendiagramm in
Java-Code um!**

Übung SemOrg

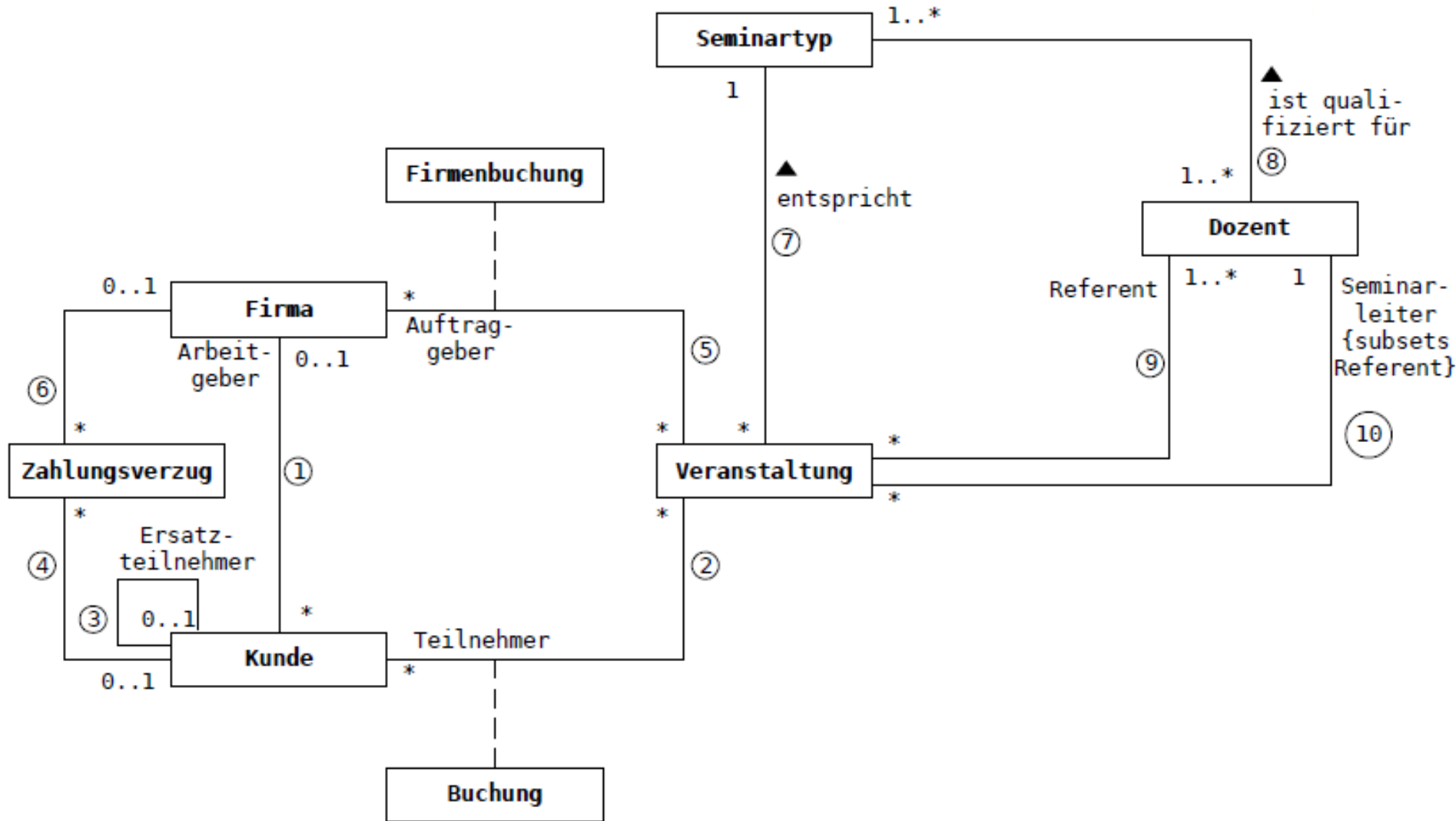


Klassendiagramme

Übung SemOrg

**Zeichnen Sie ein
Klassendiagramm mit
Assoziationen, Rollen,
Richtungen und
Multiplizitäten!**

Übung SemOrg



Klassendiagramme

Checkliste für Attribute

- ☞ sicherstellen, dass Attribute keine Objekte sind, die an einer Assoziation beteiligt sind
- ☞ prüfen, ob eine Menge von primitiven Attributen in eine eigene Klasse ausgelagert werden kann
- ☞ prüfen, ob ein Attribut nur im Zusammenhang mit einer Beziehung relevant ist → Assoziationsklasse bilden

Klassendiagramme

Attribute in eine separate Klasse auslagern - Varianten

- ∞ beschreibt die Klasse eine *Objektidentität* oder nur einen *strukturierten Datentyp*?

Klassendiagramme

Objektidentität

- ☞ jedes Objekt unterscheidet sich von allen anderen Objekten
- ☞ auch wenn zwei Objekte zufällig die gleichen Attributwerte besitzen, sind sie nicht identisch

Klassendiagramme

Objektidentität

Beispiel:

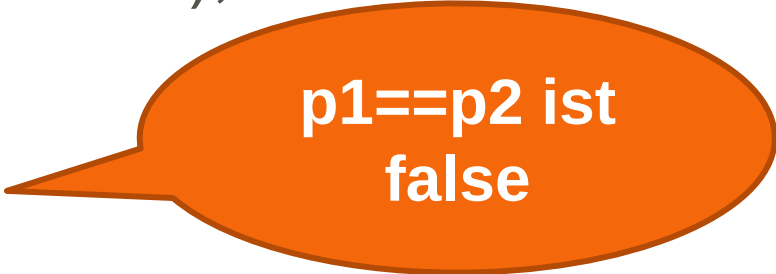
```
Person p1 = new Person();
```

```
p1.setName(„Hans“);
```

```
Person p2 = new Person();
```

```
p2.setName(„Hans“);
```

```
if(p1==p2){..}
```



**p1==p2 ist
false**

Klassendiagramme

Strukturierter Datentyp = zusammengesetzter Werttyp

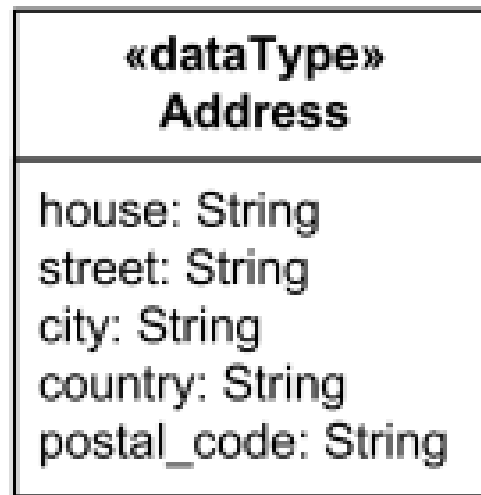
- ⌘ sein Wert ist kein Objekt bzw. besitzt keine Identität
- ⌘ Wert eines Datentyps existiert nur innerhalb des Objekts, in dem er definiert wurde
- ⌘ jeden Wert gibt es nur einmal

Klassendia

Stereotyp:
Modellelemente mit
anderer Semantik
versehen

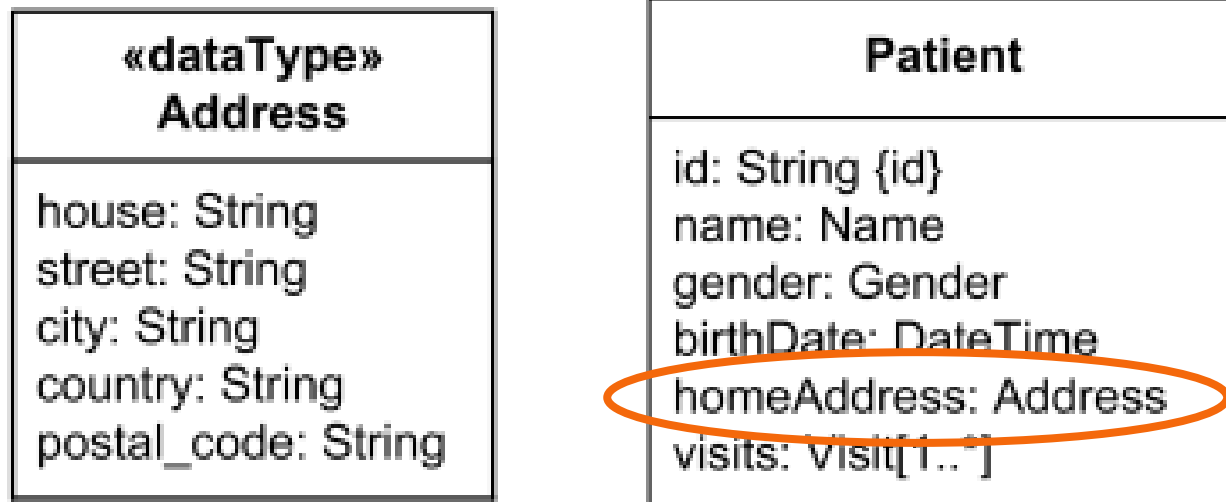
Strukturierter Datentyp

∞ in UML gibt es den Stereotyp
<<datatype>> zur Kennzeichnung
von Datentypen



Klassendiagramme

Verwendung des Datentyps für Attribute



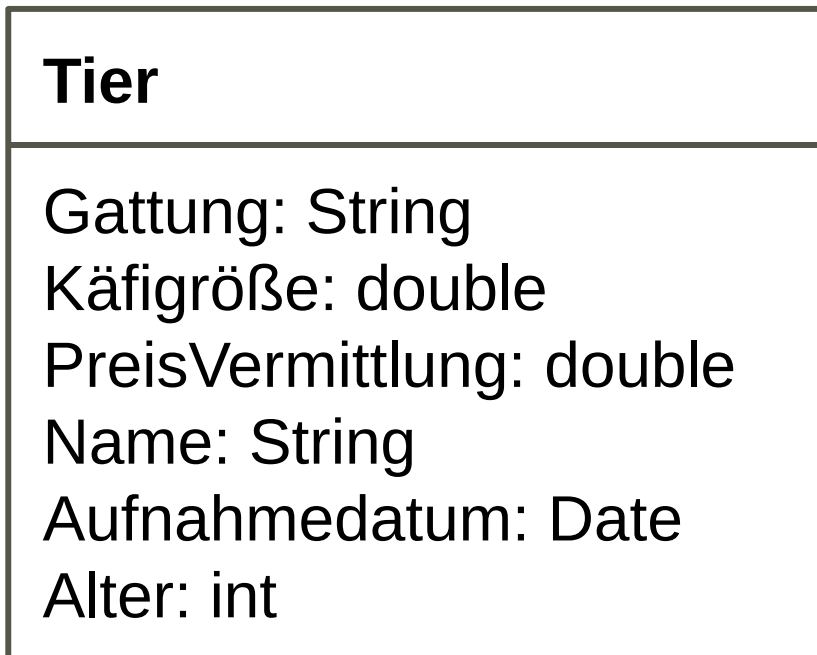
Klassendiagramme

Strukturierter Datentyp

- ⌘ Attribute des Datentyps werden in der Regel nicht einzeln geändert
- ⌘ stattdessen wird der Typ komplett neu instanziiert → neuer Wert
- ⌘ **C#**: Hier gibt es zusätzlich zu Klassen auch **struct**-Werttypen
- ⌘ [Vgl.: DateTime-struct und die Methode DateTime.Add](#)

Klassendiagramme

Beispielanwendung „Tierheim“



**Welche Attribute
könnten in eine
separate Klasse
ausgelagert werden?**

**Besitzt die neue Klasse
eine Objektidentität?**

Klassendiagramme

Anwendung in Ihrem Projekt

- ✎ Entwerfen Sie eine Paketstruktur für die Dienstplanung!
- ✎ 2 – 4 Pakete
- ✎ Erstellen Sie ein Analysemodell (Domänenmodell) für die Dienstplanung in Form eines Klassendiagramms!
- ✎ Ordnen Sie die Klassen den einzelnen Paketen zu!
- ✎ Ca. 10 Klassen