

Software Engineering 1



Anna Prenzel

Einführung

Ziele der heutigen Veranstaltung:

- ✎ Die Rolle von Modellen im SW-Entwicklungsprozess verstehen.
- ✎ Methoden zur Modellbildung kennen lernen.
- ✎ Einführung in die Methode der objektorientierten Analyse (OOA).

Wiederholung

- ⌘ Was ist der Unterschied zwischen einem Business Use Case und einem System Use Case?
- ⌘ Nennen Sie drei Dinge, die man beim Schreiben von Use Cases beachten sollte!

Requirements Engineering

Ermitteln

- Sammeln einer vollständigen Menge von Anforderungen von den Stakeholdern.

Dokumentieren (Prüfen I)

- Beseitigung redundanter, konfliktärer Anforderungen
- Priorisierung der Anforderungen
- Verhandlungen, Abstimmungen mit Stakeholdern

Spezifizieren

- Festlegung der Schnittstelle und Modellierung der Interaktion zwischen System und Umgebung

Validieren (Prüfen II)

- Abstimmung der Spezifikation mit den Stakeholdern
- Inspektion der Modelle
- Überprüfung der Nachverfolgbarkeit

RE – und jetzt?

Die Anforderungen können jetzt in Modelle umgesetzt werden.

Definition (Modell)

Ein Modell ist eine Abstraktion eines beliebigen Originals. Es beschreibt die für den Einsatzzweck des Modells relevanten Aspekte des Originals.

RE – und jetzt?

Was wird modelliert?

- ⌘ Zusammenhänge der Anwendungsdomäne – das Problem
- ⌘ das zukünftige System – die Lösung
- ⌘ wir unterscheiden zwischen Analyse- und Entwurfsmodellen

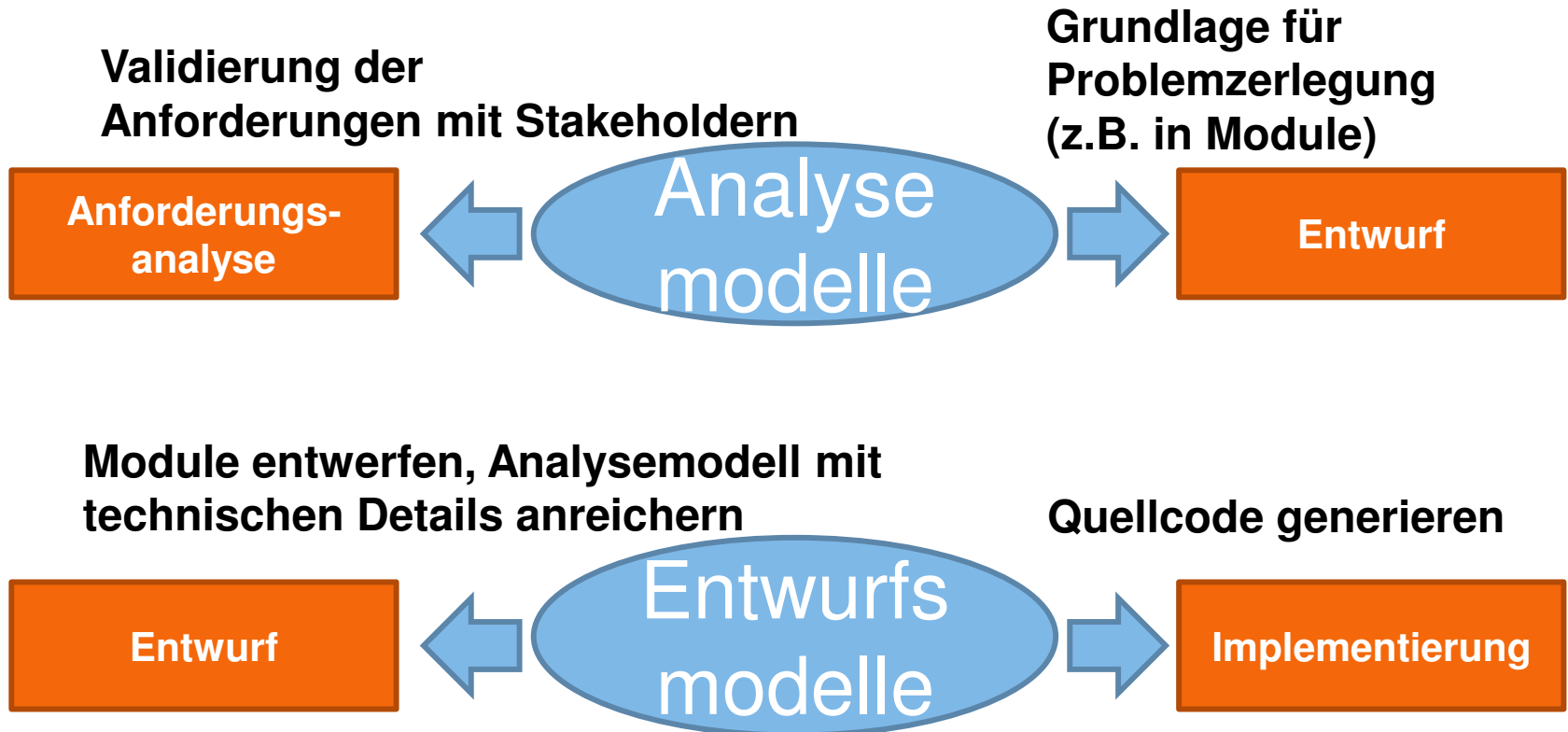
RE – und jetzt?

Wie wird modelliert?

- ✎ Verwendung einer formalen oder semiformalen Modellierungssprache
- ✎ Semiformal: Semantik der Sprachelemente nicht vollständig spezifiziert
- ✎ textuelle oder grafische Notation(Diagramme)
- ✎ wir verwenden die Unified Modeling Language (UML)¹

1) www.uml.org

Warum Modelle?



Warum Modelle?

Analysemodelle

- ✎ es wird ausschließlich mit den Begriffen der Anwendungswelt gearbeitet
- ✎ man sagt, es wird eine fachliche (anwendungsbezogene) Problemlösung erarbeitet
- ✎ andere Begriffe: Produktmodell, Fachkonzept

Warum Modelle?

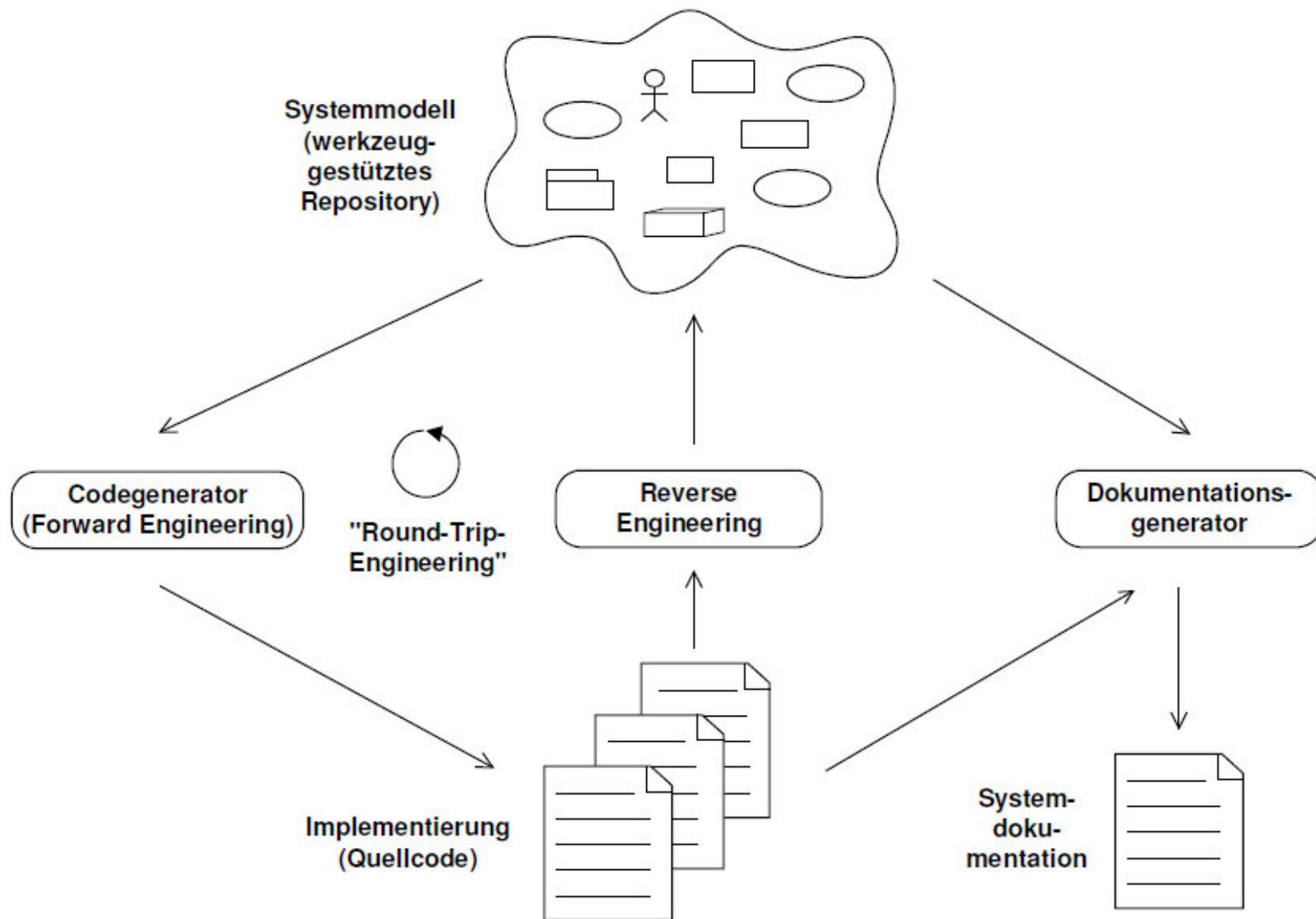
Entwurfsmodelle

- ∞ Konkretisierung des Analysemodells
- ∞ Idealfall:
 - Entwurfsmodelle werkzeuggestützt erstellen
 - Automatisierte Erzeugung von Code aus Modellen
 - Automatische Generierung von Dokumentation

Warum Modelle?

Entwurfsmodelle - Idealfall

- ✎ Änderungen ausschließlich auf Modellebene durchführen
- ✎ Code und Dokumentation bleiben automatisch konsistent und aktuell



Warum Modelle?

Warum Trennung in Analyse- und Entwurfsmodelle?

- ⌘ Software bildet Geschäftsprozesse aus der realen Welt ab
- ⌘ Zusätzlich enthält Software Konzepte, die nichts mit den Geschäftsprozessen zu tun haben → *Sicherheit, Datenaustausch, Persistenz, GUI, Datenstrukturen, ...*

 **Geschäftsprozesse separat modellieren**

Warum Modelle?

Vorteile der Trennung

- ∞ Modellierung der Geschäftsprozesse kann mit Stakeholdern diskutiert werden
- ∞ Entwurf der technischen Konzepte wird hinausgezögert, bis die Geschäftsprozesse vollständig verstanden werden.

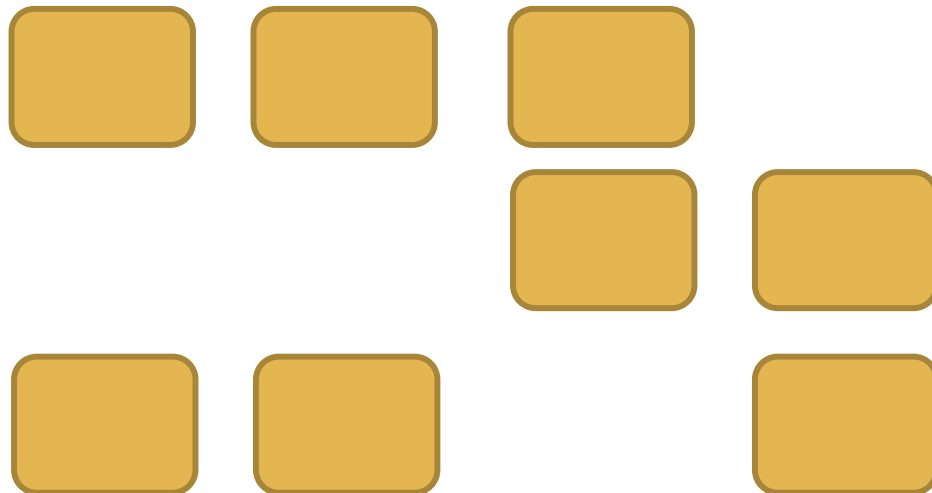
 **Trotzdem kein Strukturbruch, wenn in Analyse und Entwurf dieselbe Modellierungssprache verwendet wird.**

Warum Modelle?

Analyse



Entwurf



Orange: Konzept ist relevant für Geschäftsprozess
Braun: technisches Konzept

Methoden der Modellierung

Eine Modellierungsmethode definiert...

- ☞ eine Kombination von Basiskonzepten (und ggf. Notationen), mit denen einzelne Modellierungsgegenstände¹ abgebildet werden
- ☞ ein Vorgehensmodell, durch das festgelegt wird
 - welche Aktivitäten
 - in welchen Phasen und Teilschritten
 - ggf. mit welchen Hilfsmitteln und Werkzeugen auszuführen sind,

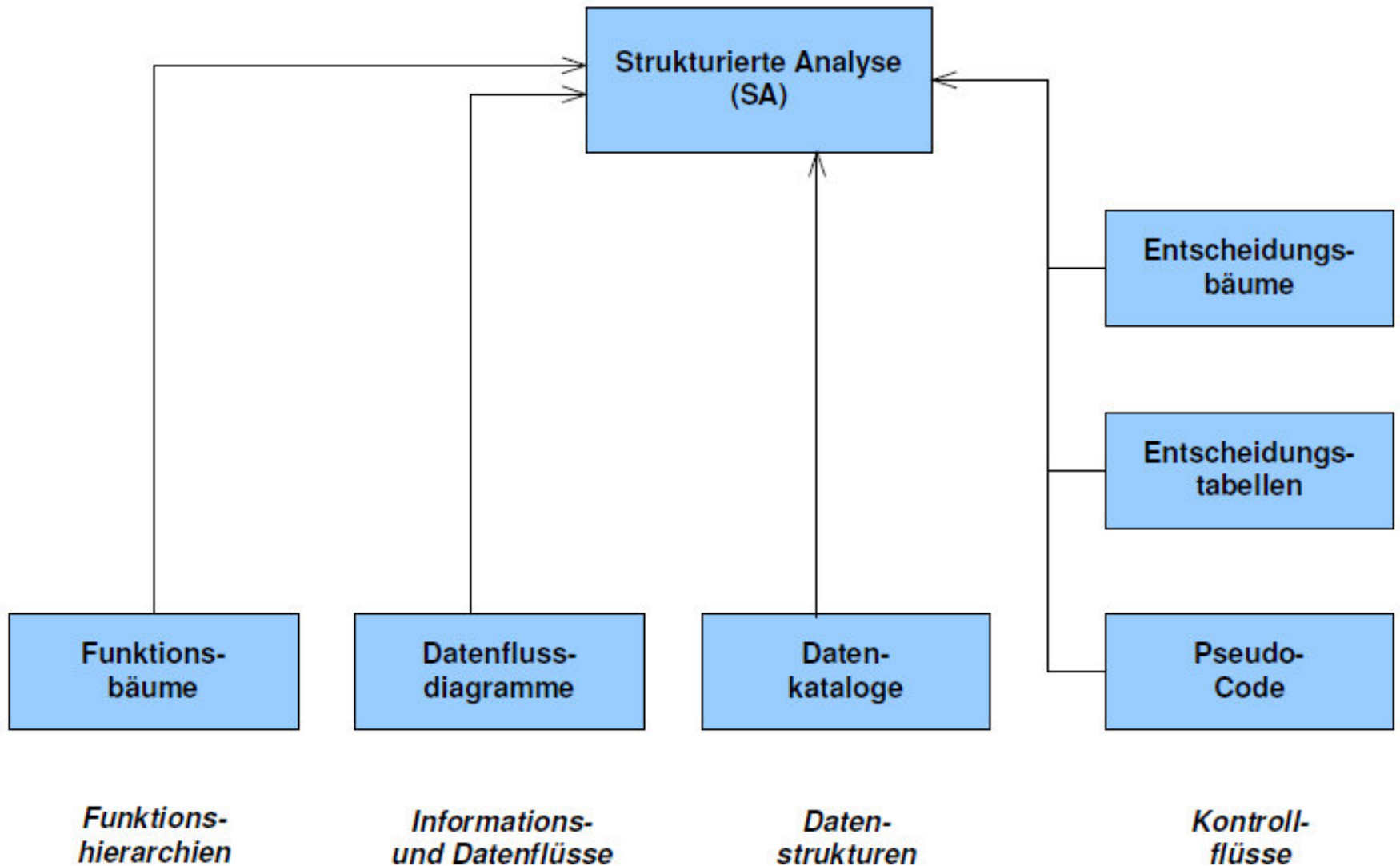
um die Teilmodelle und das integrierte Systemmodell zu erstellen.

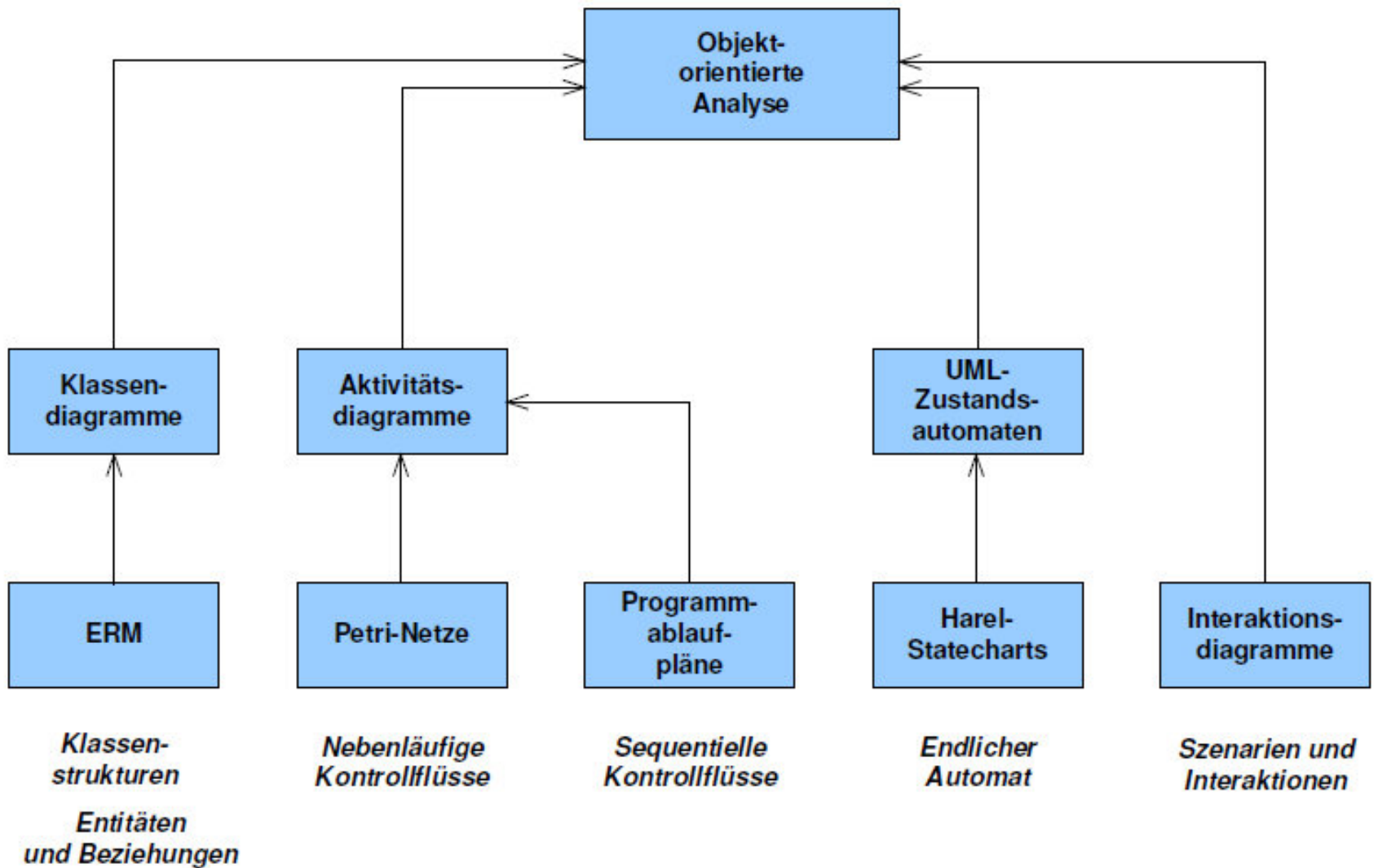
1) Daten, Funktionen, Algorithmen, Dynamik, Architektur ...

Methoden der Modellierung

Zwei mögliche Methoden

- ✎ Strukturierte Analyse/Design (SA/SD)
- ✎ Objektorientierte Analyse / Design (OOA / OOD)

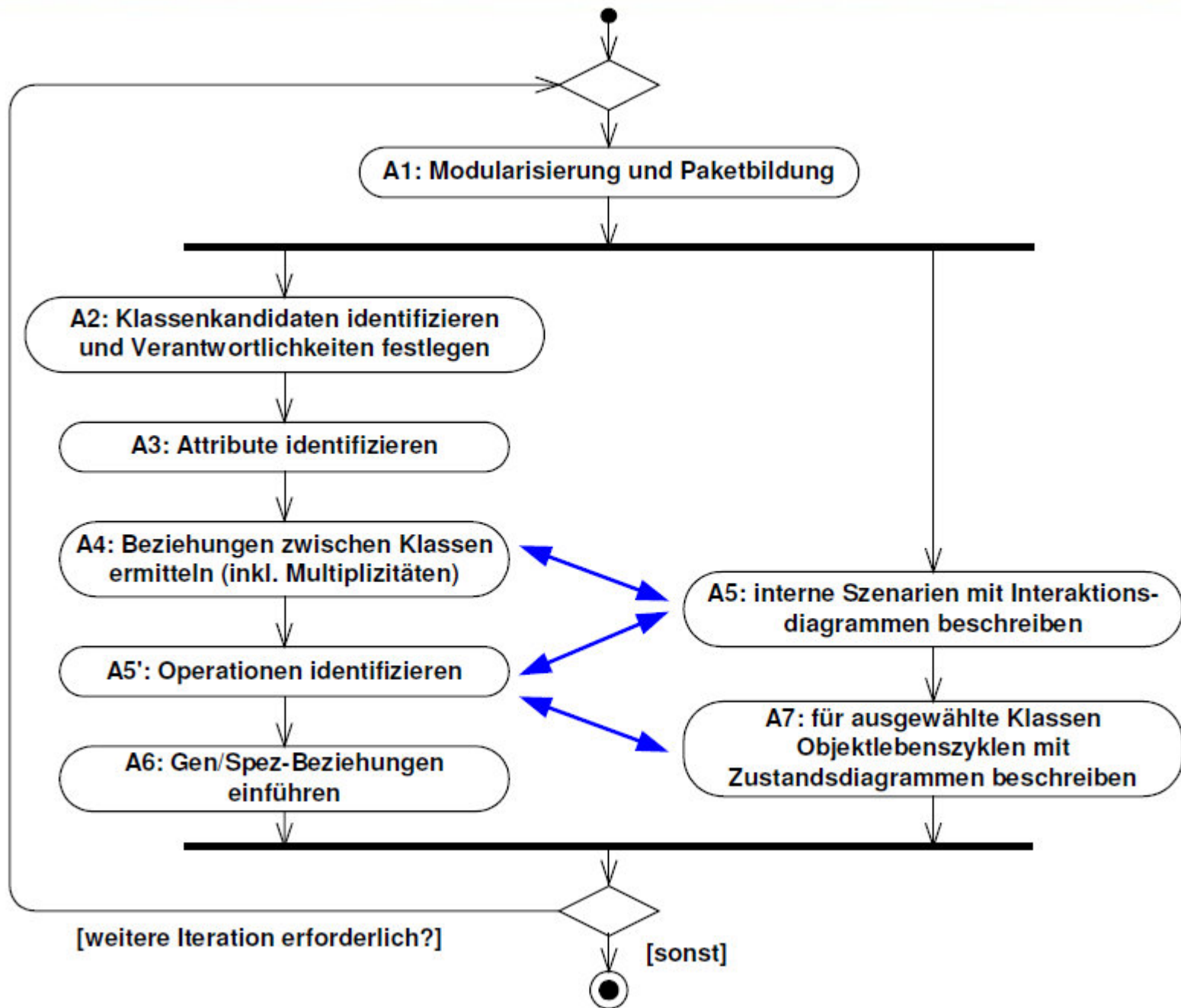




Objektorientierte Analyse

- ⌘ Zentrales Basiskonzept: das objektorientierte Klassenmodell
- ⌘ Klassen als gemeinsame Kapselungseinheit von Daten und Funktionen
- ⌘ Abbildung von Struktur und Verhalten einer Klasse durch Attribute und Operationen

 **Es folgt ein einfaches OOA-Vorgehensmodell**

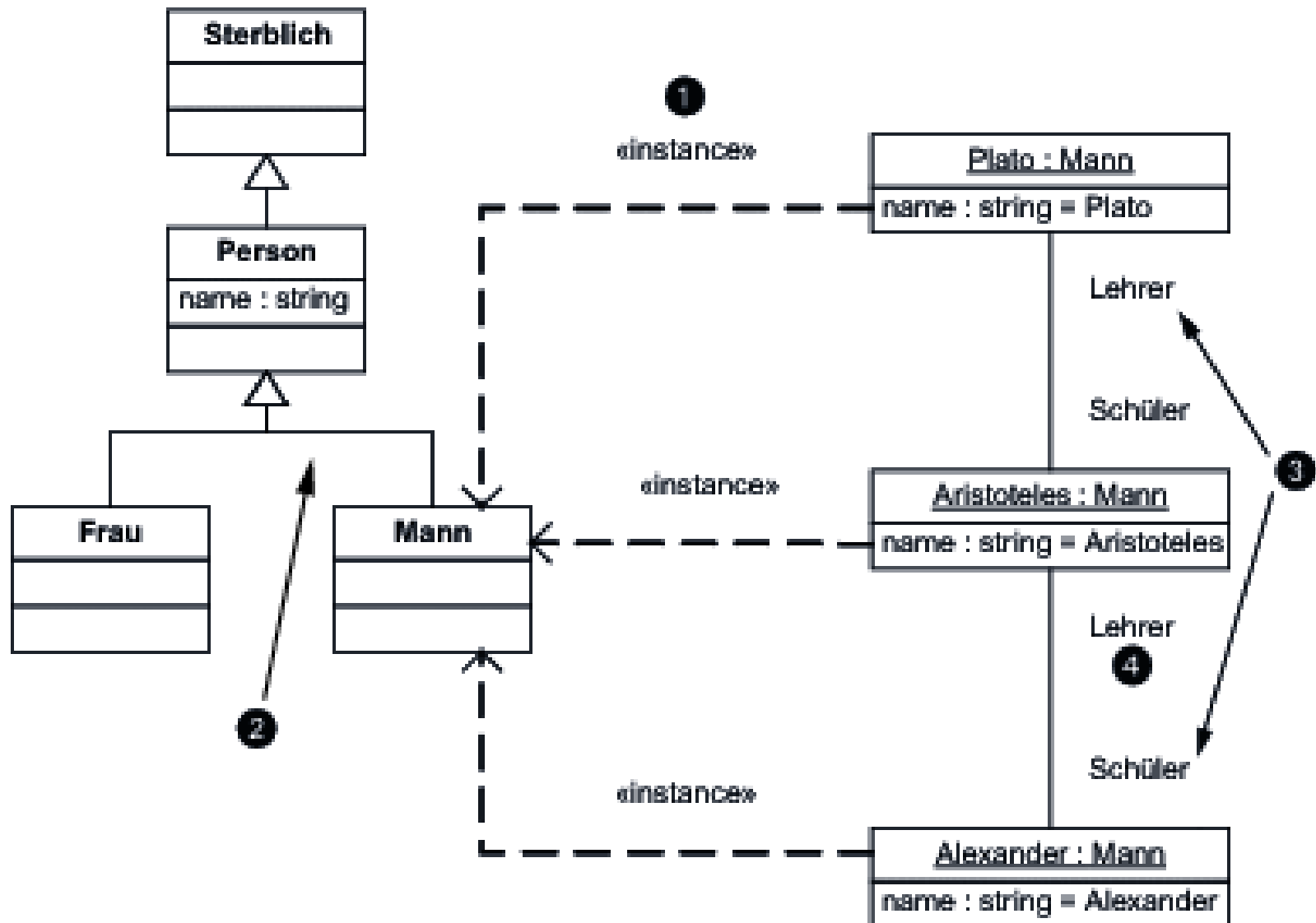


Grundlagen: UML + Beziehungen zwischen Objekten

- ∞ Beziehungen zwischen Klassen
- ∞ Beziehungen zwischen Objekten

Klassen

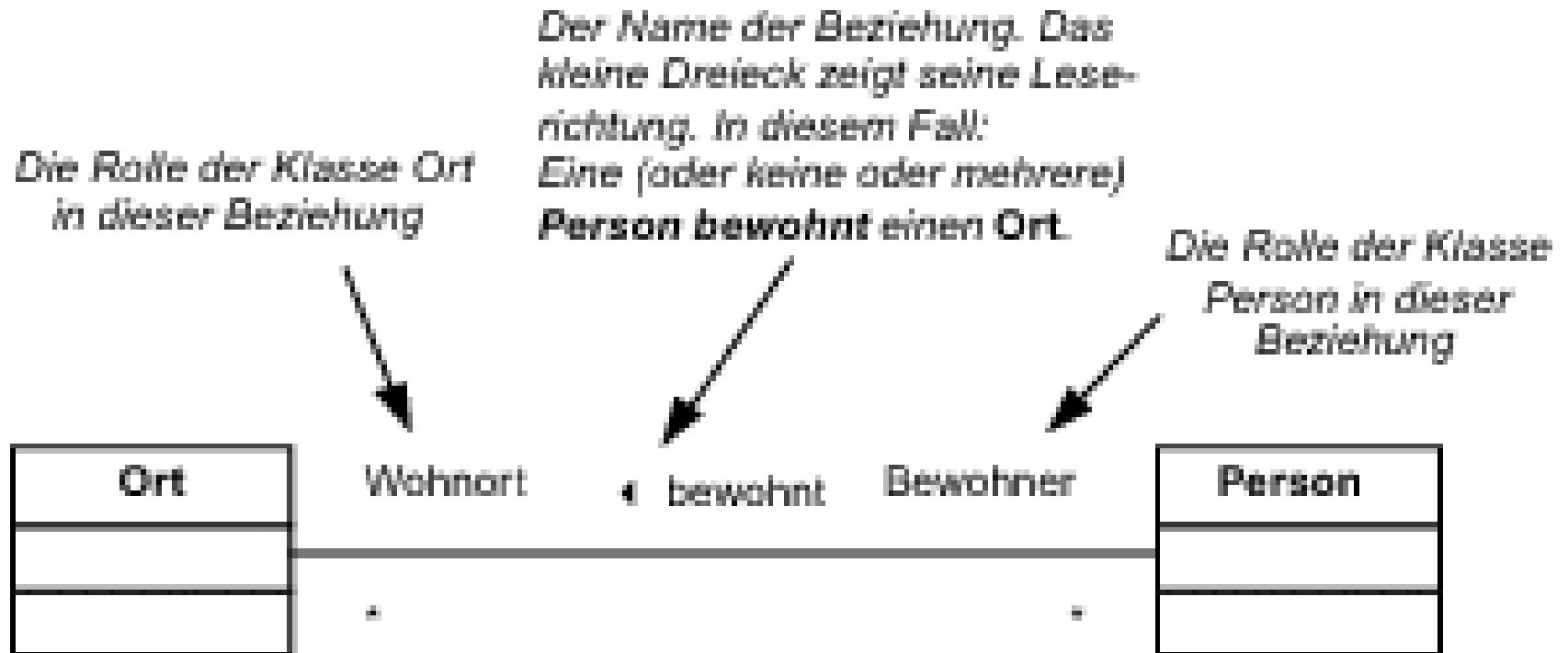
Objekte



Grundlagen: UML + Beziehungen zwischen Objekten

Assoziationen

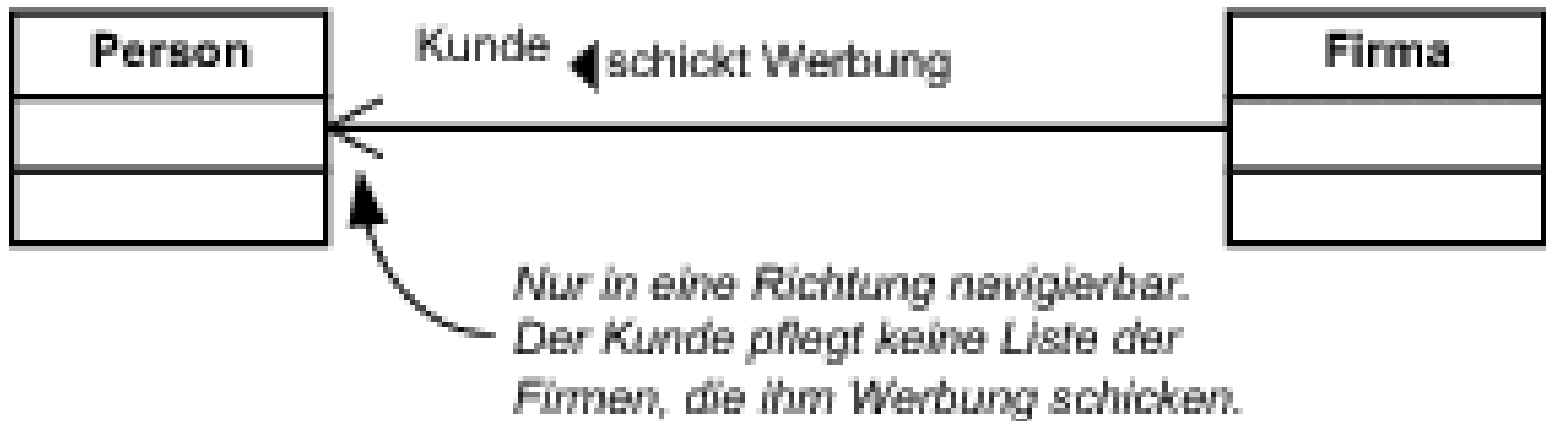
- ∞ Zwei oder mehrere Objekte können zueinander in verschiedenen Beziehungen stehen
- ∞ Jedes Objekte nimmt eine bestimmte **Rolle** in der Beziehung ein.



Grundlagen: UML + Beziehungen zwischen Objekten

Navigierbarkeit

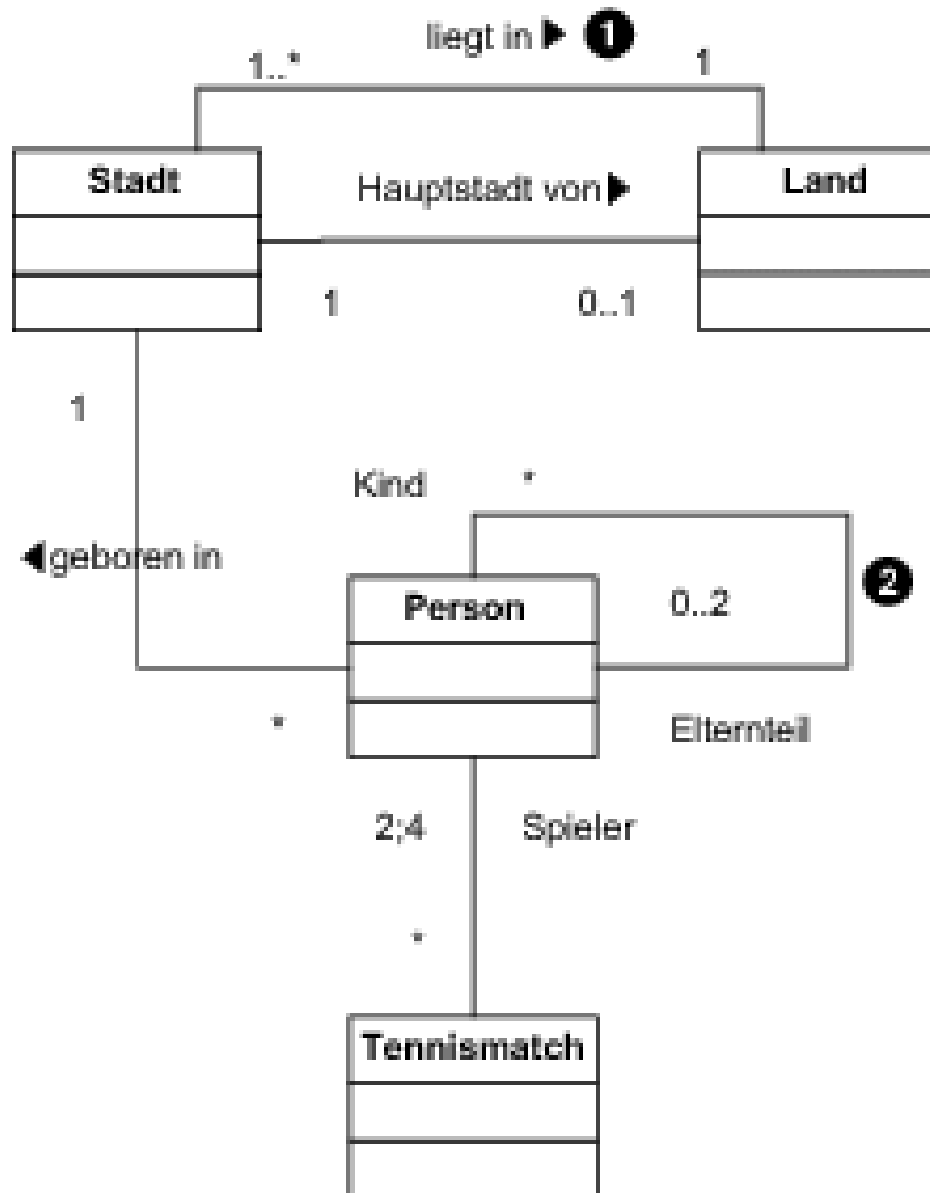
∞ Richtung einer Assoziation



Grundlagen: UML + Beziehungen zwischen Objekten

Multiplizität

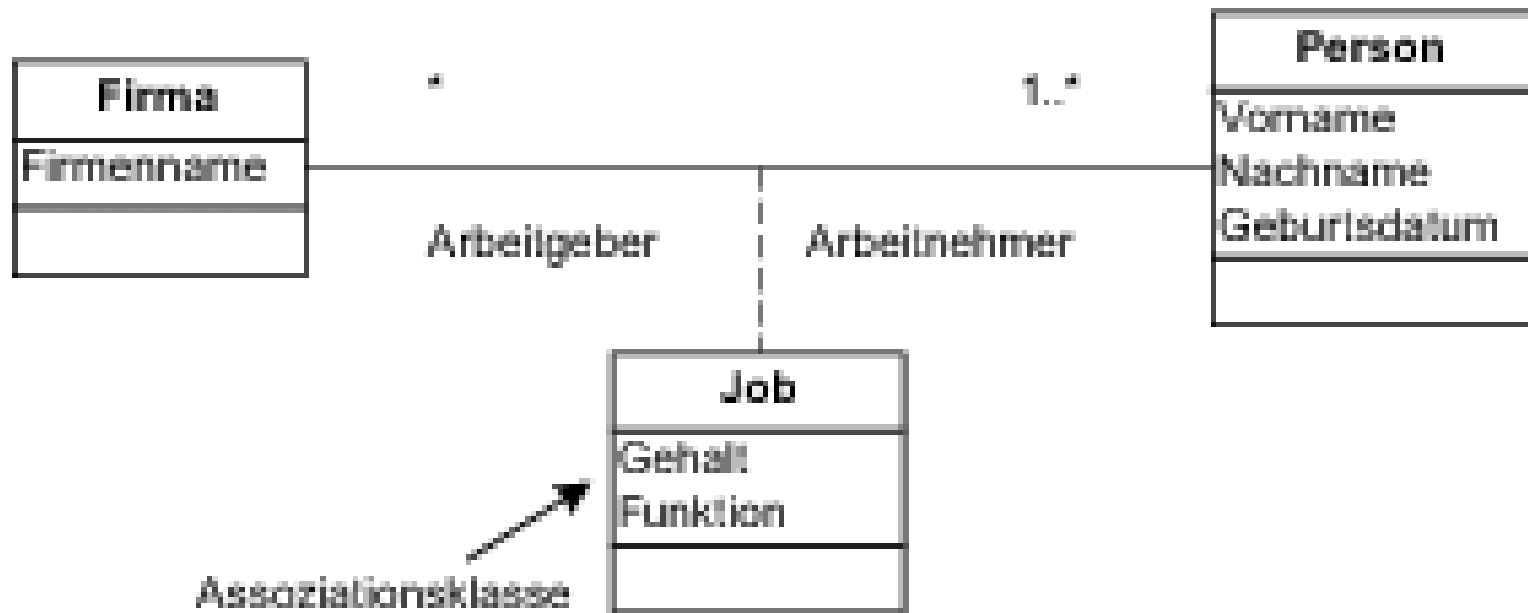
∞ legt die mögliche Anzahl der an der Beziehung beteiligten Exemplare fest.



Grundlagen: UML + Beziehungen zwischen Objekten

Assoziationsklassen

∞ Beschreibt Attribute und Operationen einer Assoziation zwischen anderen Klassen



Grundlagen: UML + Beziehungen zwischen Objekten

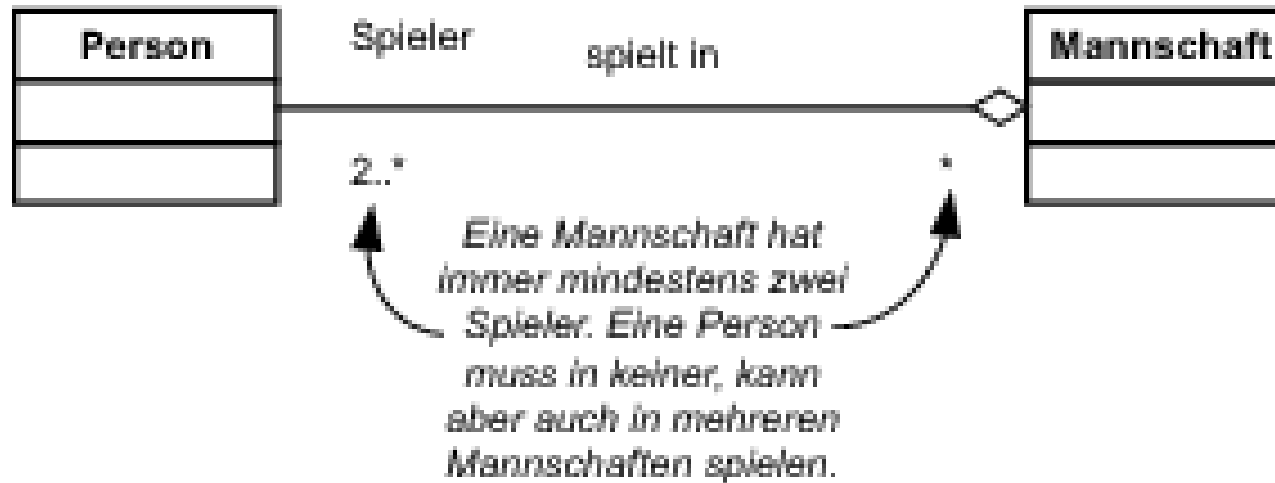
Aggregation

- ∞ Beziehung zwischen einem Objekt und seinen Teilen.
- ∞ Objekt kann ein Teil von mehreren zusammengesetzten Objekten sein.
- ∞ Die Lebensdauer der Teile kann länger sein als die Lebensdauer der Aggregate.

Grundlagen: UML + Beziehungen zwischen Objekten

Komposition

- ⌘ Beziehung zwischen einem Objekt und seinen Teilen.
- ⌘ ein Teil kann immer nur in genau einem zusammengesetzten Objekt enthalten sein
- ⌘ die Lebensdauer des zusammengesetzten Objekts entspricht immer der Lebensdauer seiner Komponenten.



Grundlagen: UML + Beziehungen zwischen Objekten

Abgeleitete Assoziation

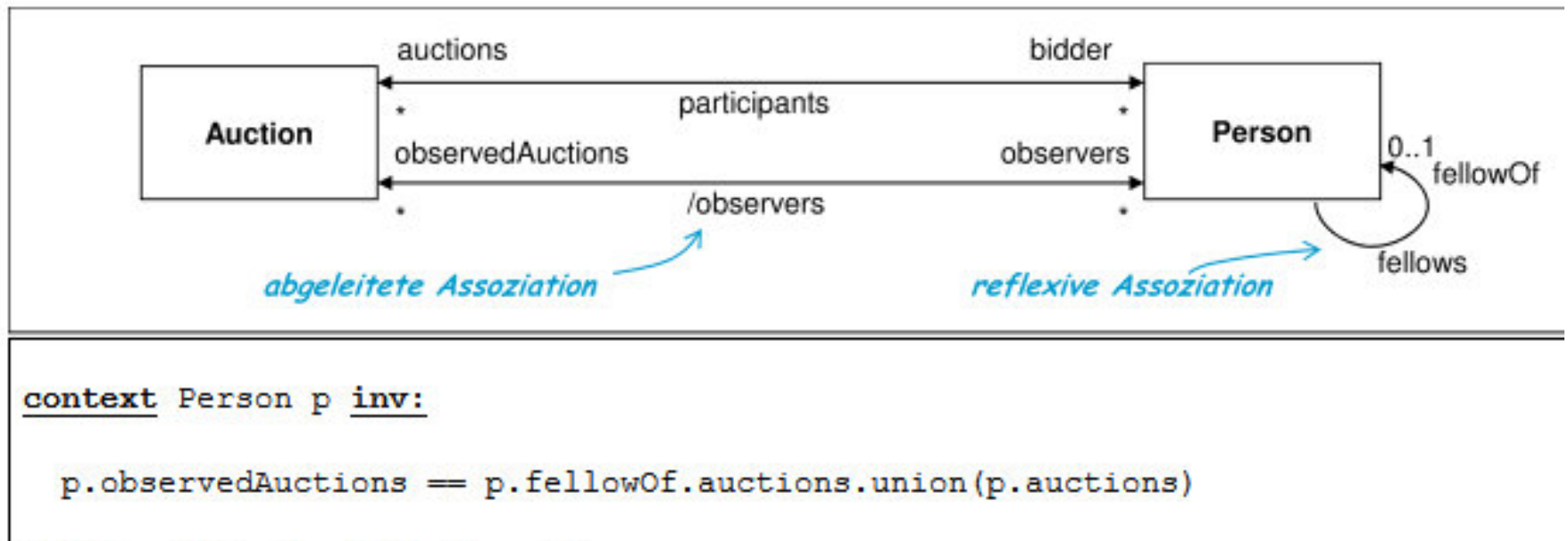
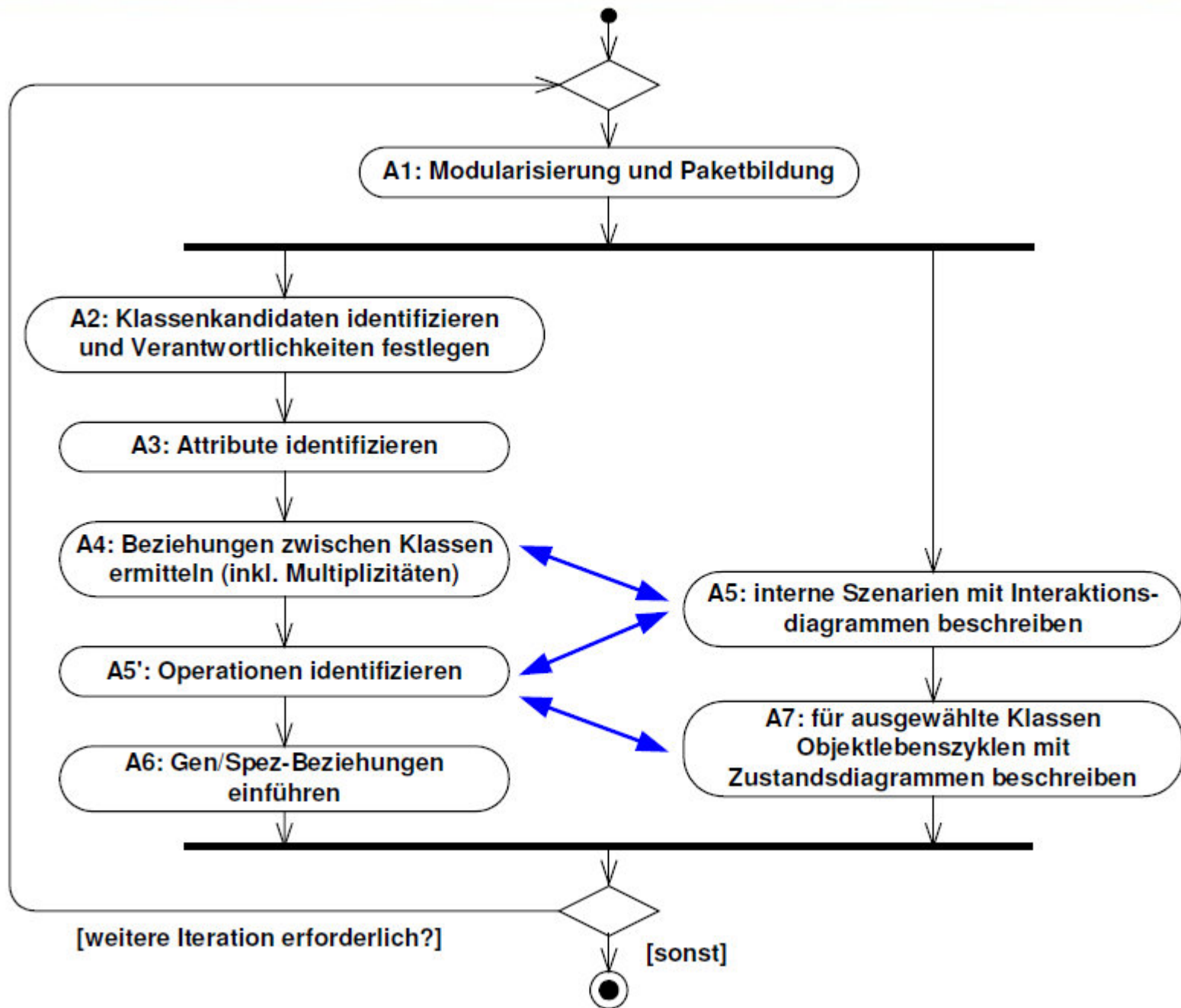


Abbildung 2.10: Abgeleitete Assoziation



Fragestellung:

- Welche *Vorgehensweisen* führen zu einer sinnvollen Paketstruktur?

Mögliche Vorgehensweisen:

▪ ***Top-Down-Ansatz***

- Eine grobe Partitionierung des Systemmodells in Pakete erfolgt bereits *zu Beginn der Systemanalyse*, d.h. vor der Identifikation der einzelnen Klassen.
- Ausgehend von der dabei gefundenen groben Paketaufteilung wird die Systemanalyse für die einzelnen Pakete separat durchgeführt.

▪ ***Bottom-Up-Ansatz***

- In einem Systemmodell, das eine kritische Größe bzw. Komplexität überschritten hat, wird durch Partitionierung in Pakete eine zusätzliche Abstraktionsebene eingeführt.
- Modellelemente, die in mehreren Paketen benötigt werden, werden dabei in separaten Paketen „ausgeklammert“ (→ „Faktorisierung“).

Fragestellung:

- Nach welchen *Kriterien* sollen die Klassen des Zielsystems zu Paketen *gruppiert* werden?

Mögliche Kriterien:

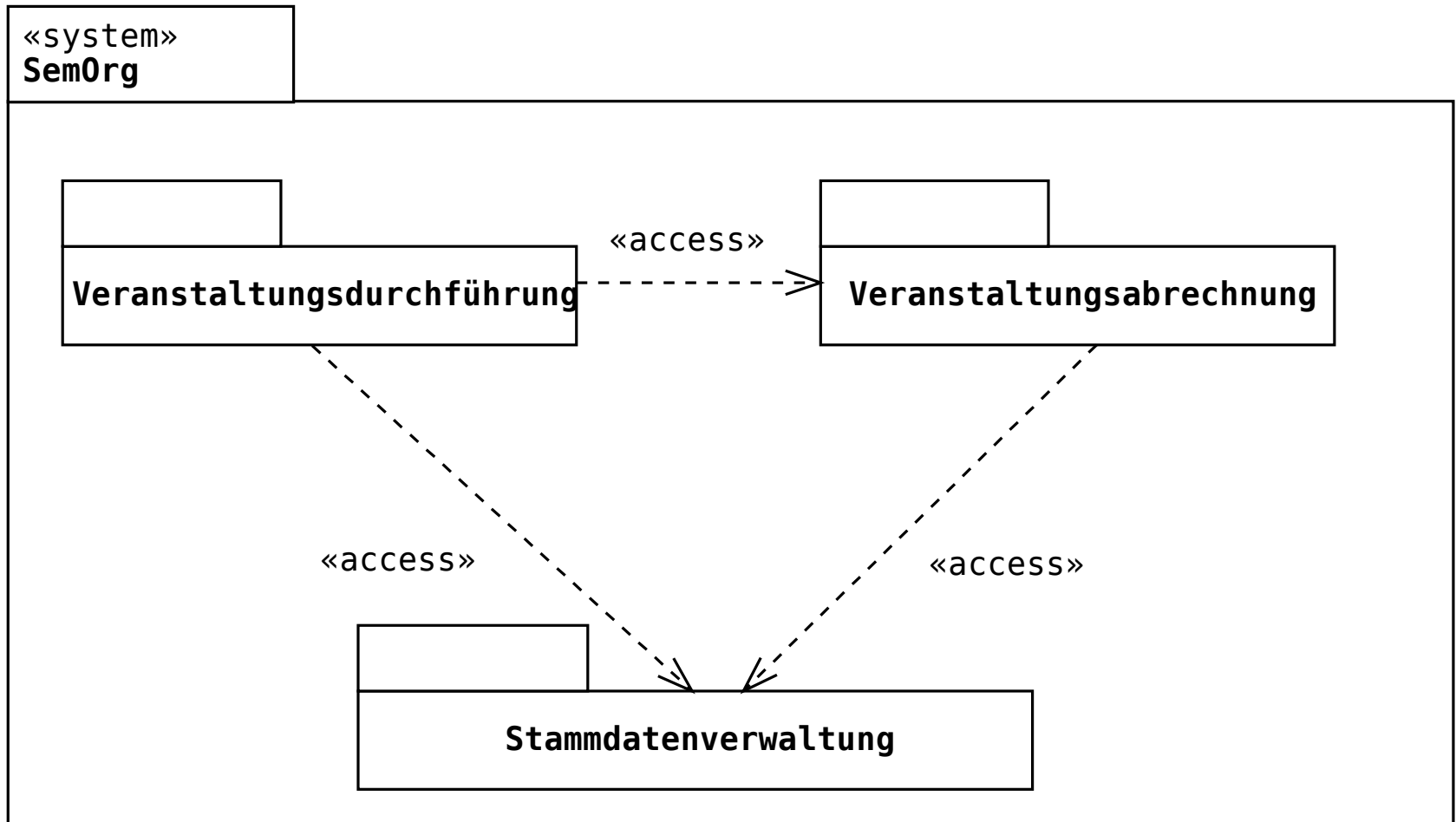
- *Maximierung der funktionalen Kohäsion innerhalb eines Paketes*
 - d.h. alle Klassen, die einem gemeinsamen funktionalen Zweck dienen, werden in einem Paket zusammengefasst.
- *Minimierung der Kopplung zwischen Paketen*
 - d.h. Klassen, die untereinander stark gekoppelt sind, aber wenige Abhängigkeiten zu Klassen anderer Pakete aufweisen, werden in einem Paket zusammengefasst.
- *Abbildung der Verteilungsstruktur des Systems auf die Paketstruktur*
 - d.h. alle Klassen, die im Rahmen der physischen System-architektur der gleichen Verteilungseinheit zugeordnet sind, werden in einem Paket zusammengefasst.

OOA-Vorgehensmodell: Checkliste für Pakete

Überprüfung der Paketstruktur

- Jedes Paket soll eine logisch weitgehend *abgeschlossene Einheit* bilden.
 - Die im Paket enthaltenen Klassen sollten sich ohne größere Schwierigkeiten unter einem *fachlogischen Oberbegriff* zusammenfassen lassen.
 - Die internen Strukturen des Pakets sollten weitgehend "autonom" (d.h. ohne Betrachtung anderer Pakete) verstanden werden können.
- *Paketübergreifende Beziehungen* sollten möglichst vermieden werden.
 - keine paketübergreifenden Aggregations- bzw. Kompositionsbeziehungen
 - möglichst wenig paketübergreifende Assoziationen
 - möglichst wenig paketübergreifende Gen/Spez-Beziehungen
- Die *Abhängigkeiten* zwischen den identifizierten Paketen müssen, soweit schon erkennbar, durch entsprechende Abhängigkeitsbeziehungen dokumentiert sein.

OOA-Vorgehensmodell: Paketbildung im Fallbeispiel Seminarorganisation



OOA-Vorgehensmodell / Aktivität 2:

Identifikation von Klassen und Verantwortlichkeiten

- Die Ermittlung geeigneter Klassen und die Festlegung von deren Verantwortlichkeiten ...
 - ...gehört zu den wichtigsten und schwierigsten Aufgaben jeder objektorientierten Analyse
 - ... kann, je nach Größe und Komplexität des Zielsystems, entweder *systemglobal* oder für einzelne *Pakete* separat durchgeführt werden
- Die Aktivitäten zur Klassenermittlung gliedern sich zumeist grob in folgende Teilschritte:
 - Identifikation von Klassenkandidaten
 - Reduktion und Stabilisierung der Klassenkandidaten
 - Festlegung von Verantwortlichkeiten für die einzelnen Klassen
 - Erstellung eines initialen Klassendiagramms

OOA-Vorgehensmodell / Aktivität 2 (Forts.): Identifikation von Klassenkandidaten

- Ausgangspunkt der Analyse sind die Dokumente der Anforderungsspezifikation:
 - Lastenheft
 - Pflichtenheft
 - Systemglossar
 - Anwendungsfallbeschreibungen
- Die *initiale* Identifikation von Klassenkandidaten kann z.B. erfolgen durch
 - *Brainstorming* des Analyseteams anhand der Anforderungsdokumente
 - Lexikalisch-syntaktische Analyse der Anforderungsdokumente („*Substantiv-Methode*“)
- Bei der „Substantiv-Methode“ werden aus den vorliegenden Anforderungsdokumenten alle geeigneten *Substantiva* selektiert:
 - Substantiva im Nominativ Singular
 - ohne substantivierte Tätigkeiten
 - ohne „offensichtlich“ ungeeignete Begriffe (Abstrakta, Floskelkonstituenten ...)

OOA-Vorgehensmodell: Checkliste für Klassen (1)

▪ **Indizien für "gute" Klassenkandidaten**

- Der Klassenkandidat ist eine logische Zusammenfassung von mehreren relevanten *Informationen/Daten* (= spätere Attribute)
 - ❖ z.B. *Auftrag, Katalog, Rechnung, Dokument, ...*
- Der Klassenkandidat repräsentiert einen *Aktor*, über den sich das Zielsystem diverse Informationen "merken" muss
 - ❖ z.B. *Kunde, Mitarbeiter, Abteilung, Motor, Roboter, ...*
- Der Klassenkandidat repräsentiert ein "typisches" Objekt der Anwendungswelt, das seiner Umgebung bestimmte Dienstleistungen (*Services*) anbietet.
 - ❖ z.B. *Bankkonto, Terminkalender, Lagerplatz, Archiv, ...*
- Liegt bereits ein ER-Modell vor, so ist jeder *Entitätstyp* und jeder *Beziehungstyp* mit Attributen ein potentieller Klassenkandidat.

OOA-Vorgehensmodell: Checkliste für Klassen (2)

- Der **Klassenname** sollte ...
 - ... ein Substantiv im Singular sein,
 - ... so konkret wie möglich gewählt werden,
 - ... der Terminologie der Anwendungswelt entsprechen,
 - ... *nicht* die Rolle von Objekten dieser Klasse in ihrer Beziehung zu den Objekten einer anderen Klasse beschreiben (→ Assoziationsklassen!)
- Als Analyseklassen sind i.a. **ungeeignet** ...
 - Klassen, die nur Datenwerte repräsentieren (z.B. *Kundenadresse*, *Überweisungsbetrag*, ...)
 - Klassen, die nur Funktionen repräsentieren (z.B. *Kundenlistendruck*, *Buchungsstorno*, ...)
 - Klassen, die zur Modellierung von Entwurfs- oder Implementierungsdetails dienen (z.B. *Transaktion*, *Menü-Eintrag*, *Datenpuffer*, ...)

OOA-Vorgehensmodell: Verantwortlichkeiten von Klassen

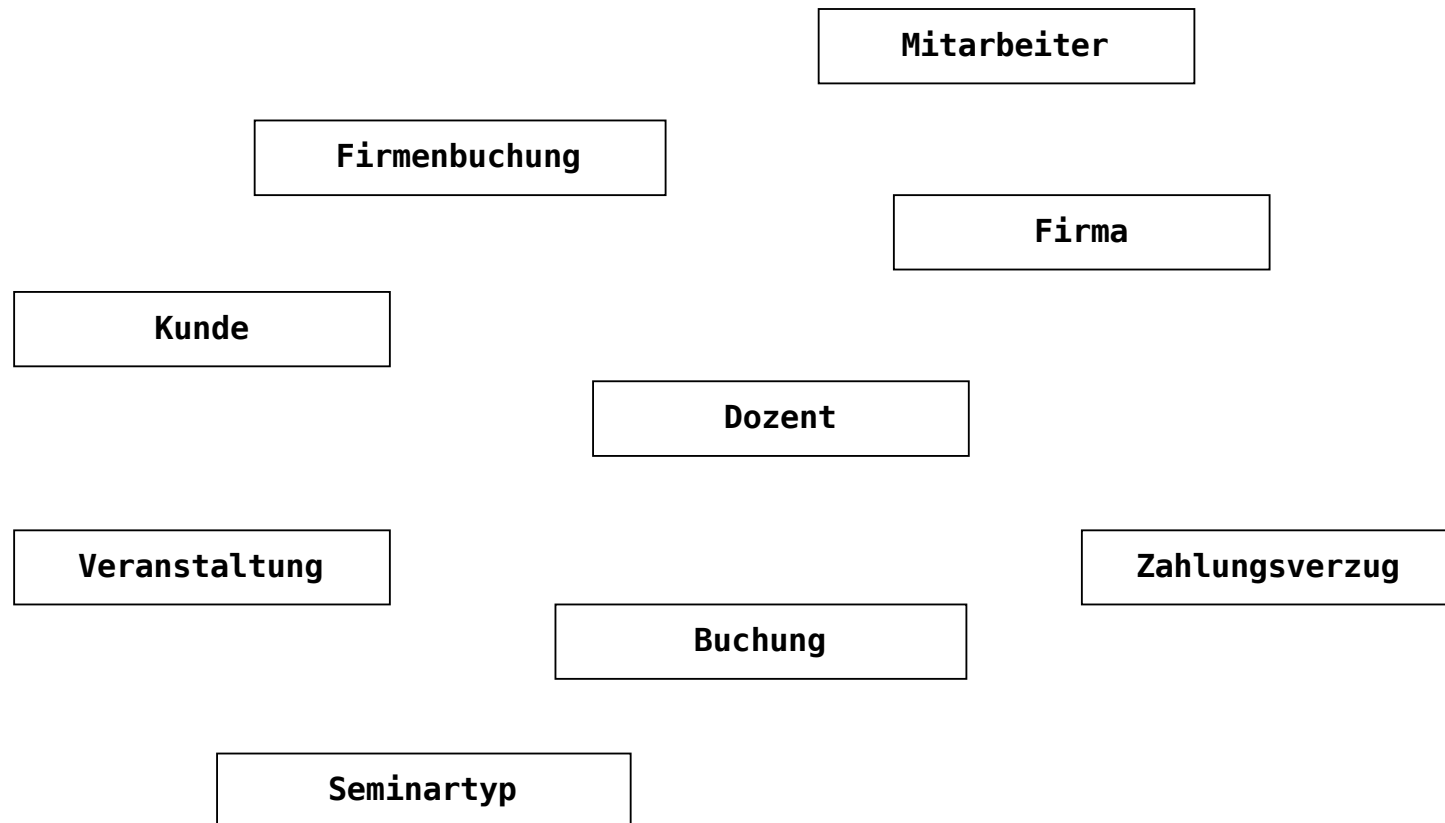
- Die Verantwortlichkeiten (*Responsibilities*) einer Klasse legen fest, welche Aufgaben die Instanzen der Klasse im Kontext des Gesamtsystems zu erfüllen haben.

➤ Beispiel: Verantwortlichkeiten der Klasse „Bankkonto“:

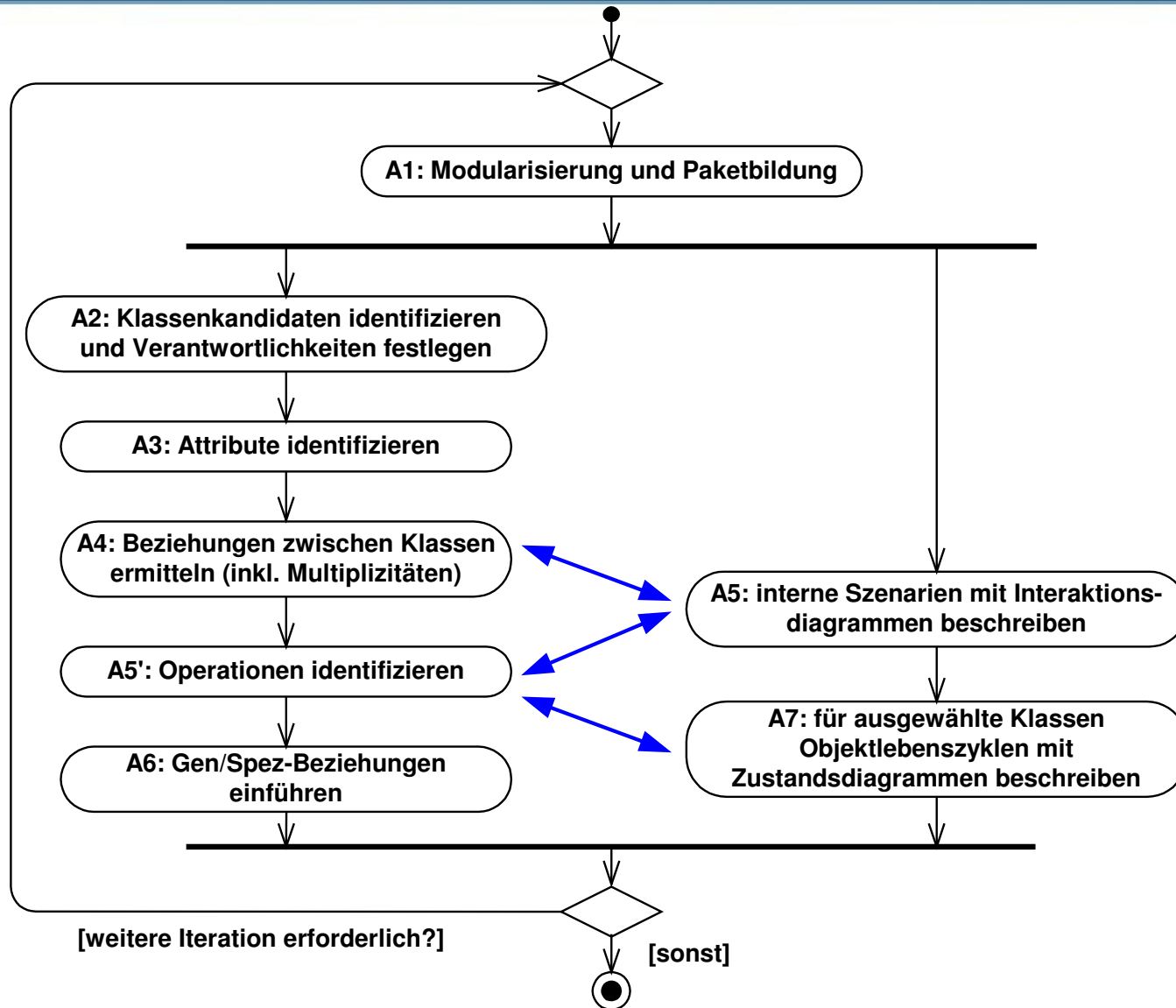
- ❖ Verwaltung des aktuellen Guthabensaldos,
- ❖ Verwaltung und Prüfung des Kontokorrentlimits,
- ❖ Verarbeitung von Soll- und Haben-Buchungen
- ❖ u.a.

- Die Verantwortlichkeiten einer Klasse werden typischerweise im frühesten Stadium der Analysephase *abstrakt* identifiziert.
 - Erst später werden sie auf *konkrete* Attribute und Operationen abgebildet, die zu ihrer Umsetzung benötigt werden.

OOA-Vorgehensmodell: Klassendiagramm für das Fallbeispiel Seminarorganisation (1. Iteration)



Ein exemplarisches OOA-Vorgehensmodell



OOA-Vorgehensmodell / Aktivität 3:

Identifikation von Attributen

- Zu den stabilen Klassenkandidaten werden nun erforderliche Attribute identifiziert (zunächst ohne Anspruch auf Vollständigkeit).
- Attribute können z.B. identifiziert werden ...
 - ... als „*Abfallprodukte*“ der Suche nach Klassenkandidaten (verworfenene Klassenkandidaten, die sinnvoller als Attribute einer anderen Klasse modelliert werden)
 - ... durch Befragung von *Experten* des Anwendungsbereichs
 - ... durch *Brainstorming* anhand der vorliegenden Anforderungsdokumente (Lastenheft, Pflichtenheft, Anwendungsfallbeschreibungen, Systemglossar)
 - ... durch Analyse der Anforderungsdokumente mit Hilfe der „*Adjektiv-Methode*“
 - ❖ Bei der „Adjektiv-Methode“ werden aus den vorliegenden Anforderungsdokumenten geeignete *Adjektiva* selektiert und den entsprechenden Klassenkandidaten als Attribute zugeordnet.

OOA-Vorgehensmodell: Checkliste für Attribute (1)

- **Indizien für die Modellierung als Attribut:**
 - keine Objektidentität, nur *Wertsemantik*
 - nur *einfache* strukturelle Eigenschaften
 - *kein* dynamisches Verhalten
 - Existenz *abhängig* von der Existenz eines (anderen) Objektes
- **Indizien für die Modellierung als separate Klasse:**
 - eigene *Objektidentität*
 - *komplexe* strukturelle Eigenschaften
 - eigenes *dynamisches Verhalten*
 - Existenz *unabhängig* von der Existenz anderer Objekte
- **Indizien für die Modellierung als Assoziation bzw. Verknüpfung:**
 - Attributwerte sind *Objekte* anderer Klassen
 - Attributwerte sind *Mengen* von Objekten anderer Klassen
- **Indiz für die Modellierung als Attribut einer Assoziationsklasse:**
 - Attributwert ist nur im Zusammenhang mit der *Beziehung* zwischen mehreren Objekten relevant.

OOA-Vorgehensmodell: Checkliste für Attribute (2)

▪ **Überprüfung der gefundenen Attribute:**

- Sind alle Attribute für die Zwecke des Zielsystems wirklich *notwendig*?
- Ist die *Anzahl* der Attribute einzelner Klassen sehr groß?
 - ❖ falls ja: Indiz für falsch gewählte Klassengranularität
- Lässt sich jedes Attribut auf (relativ) einfache Datentypen bzw. Datenstrukturen mit *Wert-Semantik* abbilden?
 - ❖ falls nicht: Indiz für separate Klasse und Assoziationsbeziehung
- Ist ein Attribut nur im Zusammenhang mit der Beziehung zu anderen Objekten relevant?
 - ❖ falls ja: Indiz für ein Assoziationsattribut
- Handelt es sich bei einem Attribut um
 - ❖ ein Klassenattribut?
 - ❖ ein Schlüsselattribut (= fachlogische Identifikation des Objektes) ?
 - ❖ ein abgeleitetes Attribut?

falls ja: Entsprechende Kennzeichnung vornehmen!

OOA-Vorgehensmodell: Klassendiagramm für das Fallbeispiel Seminarorganisation (2. Iteration)

Veranstaltung
VeranstaltungsNr. Dauer Beginn Ende Agenda Veranstaltungsort TeilnehmerMax storniert

Firma
Kurzname Name Adresse Kontakt Ansprechpartner Abteilung Funktion /Umsatz Kunde seit

Seminartyp
Kurztitel Titel Zielsetzung Inhaltsübersicht Dauer Unterlagen Zielgruppe Gebühr TeilnehmerMax TeilnehmerMin

Mitarbeiter
Name Berechtigung Funktion Passwort

Zahlungsverzug
Rechungsdatum Rechnungsbetrag

Kunde
Kundennummer Name Adresse Geburtsdatum Funktion /Umsatz Kunde seit

Dozent
Dozentennummer Name Adresse Geburtsdatum Biografie Honorar pro Tag Dozent seit

Buchung
angemeldet am bestätigt am Rechnungsversand am abgemeldet am Mitteilung am

Firmenbuchung

OOA-Vorgehensmodell / Aktivität 4 (Forts.): Identifikation von Beziehungen mittels Textanalyse

- Ausgangspunkt der Analyse sind wiederum die *Dokumente der Anforderungsdefinitionsphase* (Problem Statements, Lastenheft, Anwendungsfallbeschreibungen, ...)
- Sämtliche Vorkommen bereits identifizierter *Klassenkandidaten* werden im Text aufgesucht und darauf hin untersucht, ob sie *Hinweise auf Beziehungen* zu anderen bereits identifizierten Klassenkandidaten liefern.
- „Typische“ Hinweise auf semantische Beziehungen sind z.B.
 - Verb- bzw. Partizipialkonstruktionen, die Objektzusammenhänge charakterisieren („*A ist B zugeordnet*“, „*ein aus X bestehendes Y*“)
 - Aktivitäten eines Objektes, deren Durchführung eine Beziehung zu einem anderen Objekt zwingend voraussetzen (z.B. „*Kunde nimmt an Veranstaltung teil*“)
 - Substantive mit Ergänzungen im Genetiv, wenn die entsprechenden Modellinformationen klassenwertig sind oder mehrfach zugeordnet werden können (z.B. „*die Positionen der Rechnung*“, „*die Bankverbindung des Kunden*“, „*die Teilnehmer der Veranstaltung*“)

OOA-Vorgehensmodell / Aktivität 4 (Forts.): Spezifikation von Assoziationen im Klassenmodell

▪ **Vorgehensweise**

- Im ersten Schritt wird i.a. nur die Assoziation als solche identifiziert und im Klassendiagramm eingetragen.
- Ist die Menge der Assoziationen hinreichend stabil, können weitere Informationen (Multiplizitäten, Constraints, etc.) hinzugefügt werden.

▪ **Benennung der Assoziationen**

- Eine Benennung der Assoziation ist nicht unbedingt erforderlich, wenn ihre Semantik eindeutig aus dem Zusammenhang erkennbar ist.
- Eine Benennung mittels *Rollennamen* ist sinnvoll bzw. erforderlich, wenn
 - ❖ ... eine reflexive Assoziation vorliegt
 - ❖ ... zwischen zwei Klassen mehrere Assoziationen bestehen
 - ❖ ... eine Klasse in verschiedenen Assoziationen verschiedene Rollen spielt
- Eine Benennung mittels *Rollennamen* (Substantiven) ist gegenüber *Assoziationsnamen* (Verben) zu bevorzugen.

OOA-Vorgehensmodell: Checkliste für Assoziationen

▪ **Überprüfung der identifizierten Assoziationen**

- Existieren zwischen zwei Klassen *mehrere Assoziationen*?
 - ❖ Handelt es sich tatsächlich um semantisch wohlunterscheidbare Beziehungen?
 - ❖ Besteht zwischen den Projektionsmengen evtl. eine Inklusionsbeziehung? (UML: Constraint {subsets})
- Liegt zwischen zwei Klassen eine *1:1-Beziehung* vor?
 - ❖ Modellierung durch zwei separate Klassen ist sinnvoll, wenn
 - ... die Beziehung in min. einer Richtung optional ist,
 - ... min. eine der beiden Klassen sehr umfangreich ist,
 - ... die beiden Klassen eine wohlunterscheidbare Semantik haben.
 - ❖ Ansonsten ist eine Verschmelzung der beiden Klassen zu erwägen.
- Hat eine der Assoziationen Aggregations- oder Kompositionssemantik?
- Liegt bei einer der Assoziationen eine *abgeleitete Beziehung* vor?
 - ❖ Falls ja: Welche Gründe sprechen für bzw. gegen die redundante Modellierung?

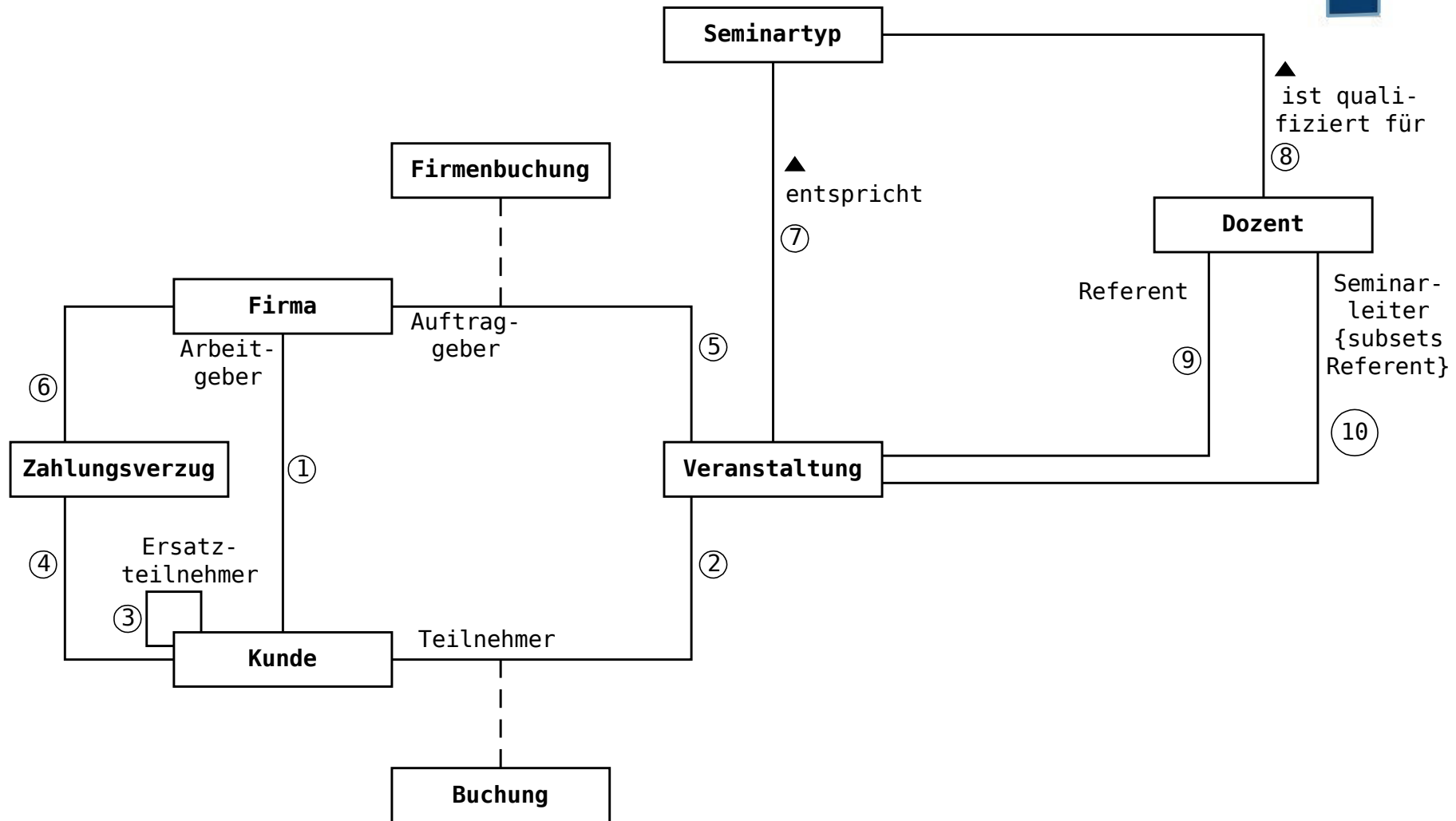
OOA-Vorgehensmodell: Assoziationen im Fallbeispiel Seminarorganisation (1)

- Aus der Anforderungsspezifikation zum Fallbeispiel "Seminarorganisation" lassen sich folgende Beziehungen zwischen den Klassen ableiten:
 1. Ein *Kunde* kann Mitarbeiter einer *Firma* sein.
 - Assoziation Arbeitgeber zwischen den Klassen Kunde und Firma
 1. Ein *Kunde* kann an einer *Veranstaltung* teilnehmen.
 - Assoziation Teilnehmer zwischen den Klassen Kunde und Veranstaltung
 1. Ein *Kunde* kann bei Abmeldung einen Ersatzteilnehmer melden.
 - reflexive Assoziation Ersatzteilnehmer über der Klasse Kunde
 1. Ein *Kunde* kann sich im *Zahlungsverzug* befinden.
 - Assoziation zwischen den Klassen Kunde und Zahlungsverzug
 1. Eine *Firma* kann eine *Veranstaltung* buchen.
 - Assoziation Auftraggeber zwischen den Klassen Firma und Veranstaltung

OOA-Vorgehensmodell: Assoziationen im Fallbeispiel Seminarorganisation (2)

1. Eine *Firma* kann sich im *Zahlungsverzug* befinden.
 - Assoziation zwischen den Klassen *Firma* und *Zahlungsverzug*
1. Eine *Veranstaltung* gehört zu einem bestimmten *Seminartyp*.
 - Assoziation entspricht zwischen den Klassen *Veranstaltung* und *Seminartyp*
1. Ein *Dozent* kann zur Durchführung von Veranstaltungen eines bestimmten *Seminartyps* qualifiziert sein.
 - Assoziation ist qualifiziert für zwischen den Klassen *Dozent* und *Seminartyp*
1. Ein Dozent kann an einer Veranstaltung mitwirken.
 - Assoziation Referent zwischen den Klassen *Dozent* und *Veranstaltung*
1. Ein Dozent kann eine Veranstaltung leiten.
 - Assoziation Seminarleiter zwischen den Klassen *Dozent* und *Veranstaltung*

OOA-Vorgehensmodell: Klassendiagramm für das Fallbeispiel Seminarorganisation (3. Iteration)



OOA-Vorgehensmodell / Aktivität 4 (Forts.): Spezifikation von Multiplizitäten

▪ **Vorgehensweise**

- Zu *jeder* Assoziation müssen die Multiplizitäten (für *beide* Richtungen!) aus der Anforderungsspezifikation erschlossen werden.
- Unklare Multiplizitäten stellen einen deutlichen Hinweis auf unpräzise Anforderungen dar und sollten unbedingt mit den Fachexperten diskutiert und geklärt werden!

▪ **Typische Fragestellungen**

- Gibt es zu einem festen, aber beliebigen Objekt der Klasse X *höchstens* ein Objekt der Klasse Y oder kann es *mehrere* zugeordnete Objekte geben?
- Gibt es zu einem festen, aber beliebigen Objekt der Klasse X *mindestens* ein Objekt der Klasse Y (*Muss-Beziehung*) oder ist es zulässig, dass *kein* zugeordnetes Objekt existiert (*Kann-Beziehung*)?
- Gibt es für die Anzahl der zugeordneten Objekte eine feste *Obergrenze* oder eine feste *Untergrenze*?

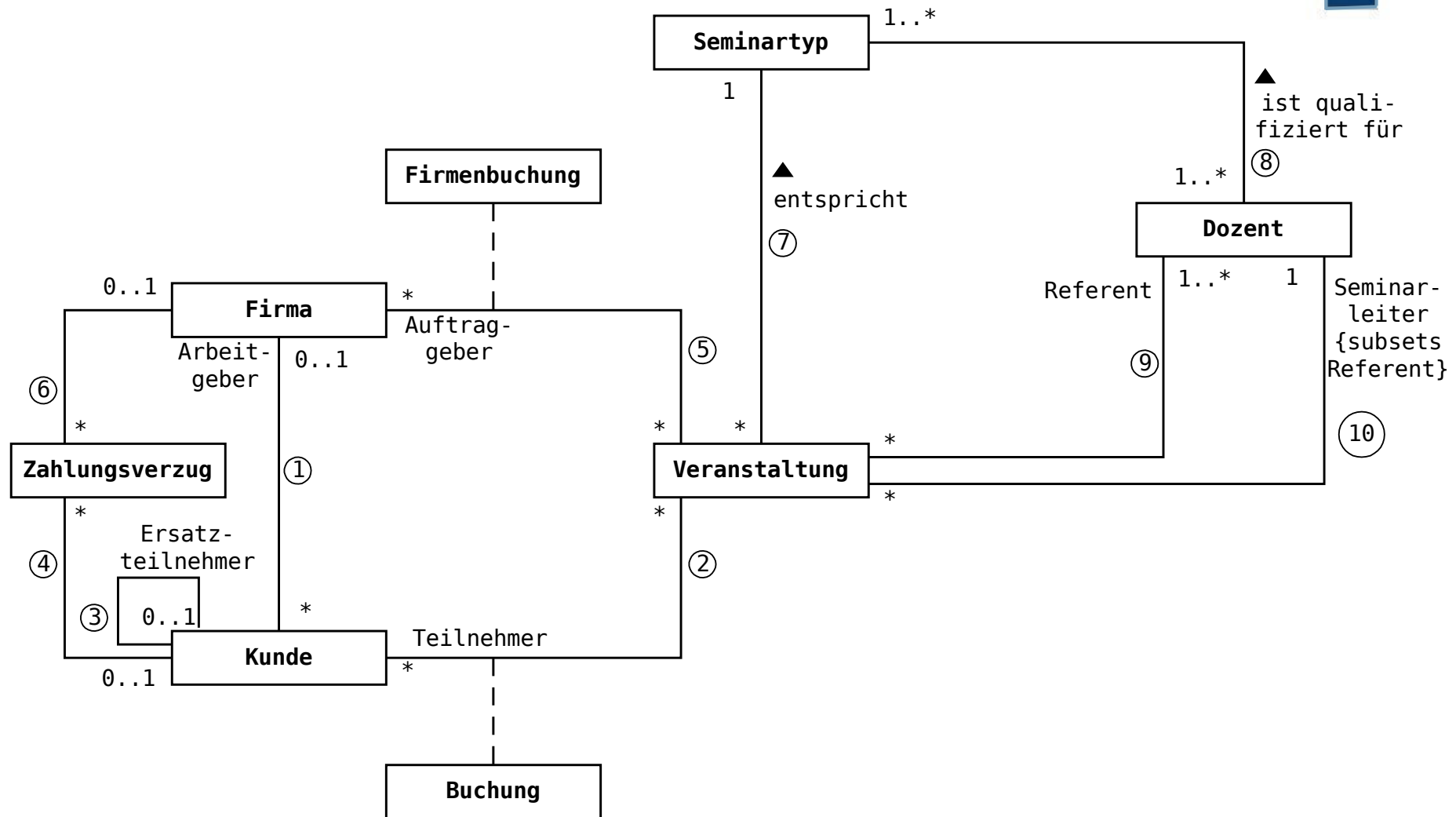
OOA-Vorgehensmodell: Multiplizitäten im Fallbeispiel Seminarorganisation (1)

- Aus der Anforderungsspezifikation zum Fallbeispiel "Seminarorganisation" lassen sich folgende Multiplizitäten für die gefundenen Assoziationen ableiten:
 1. Ein *Kunde* kann Mitarbeiter einer *Firma* sein (optionale Beziehung, 0..1). Eine *Firma* kann Arbeitgeber von beliebig vielen ($n \geq 0$) *Kunden* sein.
 2. Ein *Kunde* kann an beliebig vielen ($n \geq 0$) *Veranstaltungen* teilnehmen. An einer *Veranstaltung* können beliebig viele ($n \geq 0$) *Kunden* teilnehmen.
 3. Ein *Kunde* kann bei Abmeldung genau einen *Ersatzteilnehmer* melden (optionale Beziehung, 0..1). Ein *Ersatzteilnehmer* vertritt genau einen *Kunden*.
 4. Ein *Zahlungsverzug* kann genau einem *Kunden* zugeordnet sein (optionale Beziehung, 0..1). Einem *Kunden* können beliebig viele ($n \geq 0$) *Zahlungsverzüge* zugeordnet sein.
 5. Eine *Firma* kann beliebig viele ($n \geq 0$) *Veranstaltungen* buchen. Eine *Veranstaltung* kann von beliebig vielen ($n \geq 0$) *Firmen* gebucht werden.

OOA-Vorgehensmodell: Multiplizitäten im Fallbeispiel Seminarorganisation (2)

1. Ein *Zahlungsverzug* kann genau einer *Firma* zugeordnet sein (optionale Beziehung, 0..1). Einer *Firma* können beliebig viele ($n \geq 0$) *Zahlungsverzüge* zugeordnet sein.
2. Eine *Veranstaltung* gehört zu genau einem *Seminartyp*. Zu einem *Seminartyp* kann es beliebig viele ($n \geq 0$) *Veranstaltungen* geben.
3. Ein *Dozent* muss zur Durchführung von Veranstaltungen mindestens eines *Seminartyps* qualifiziert sein ($n \geq 1$). Zu einem *Seminartyp* muss es mindestens einen qualifizierten *Dozenten* geben ($n \geq 1$) .
4. Ein *Dozent* kann an beliebig vielen ($n \geq 0$) *Veranstaltungen* als Referent mitwirken. An einer *Veranstaltung* muss mindestens ein ($n \geq 1$) *Dozent* als Referent mitwirken.
5. Ein *Dozent* kann beliebig viele ($n \geq 0$) *Veranstaltungen* leiten. Zu einer *Veranstaltung* muss es genau einen *Dozenten* geben, der sie leitet.

OOA-Vorgehensmodell: Klassendiagramm für das Fallbeispiel Seminarorganisation (4. Iteration)



OOA-Vorgehensmodell / Aktivität 5:

Beschreibung interner Szenarien

- **Zielsetzung**

- Die in der Anforderungsdefinitionsphase ermittelten extern sichtbaren Funktionalitäten müssen auf Interaktionen zwischen internen Objekten abgebildet werden.

- **Vorgehensweise**

- Zu jedem relevanten Anwendungsfall sind alle relevanten externen Szenarien aus der Anforderungsspezifikation zu analysieren.
 - ❖ Zu jedem relevanten externen Szenario ist ein korrespondierendes *internes Szenario* zu erstellen.
 - ❖ Geeignete *Notation* für die internen Szenarien: *UML-Interaktionsdiagramme*, insbes. Sequenz- oder Kommunikationsdiagramme
- Ausgangspunkt für die Erstellung eines internen Szenarios:
 - ❖ Ablaufbeschreibung des entsprechenden externen Szenarios (mit Vor- und Nachbedingungen)
 - ❖ aktueller Stand des Klassenmodells

OOA-Vorgehensmodell / Aktivität 5 (Forts.): Beschreibung interner Szenarien

▪ **Vorgehensweise (Forts.)**

- Schritte zur Erstellung eines internen Szenarios:
 - ❖ Identifiziere diejenigen Klassen, die zur internen Realisierung des externen Szenarios benötigt werden (Anhaltspunkt: Verantwortlichkeiten der Klassen)
 - ❖ Definiere Objekte dieser Klassen als Interaktionspartner des internen Szenarios
 - ❖ Bilde den Ablauf des externen Szenarios auf eine Sequenz von Nachrichten (= Operationsaufrufen) zwischen diesen Interaktionspartnern ab
 - ❖ Beschreibe die entsprechende Nachrichtensequenz in Form eines Interaktionsdiagramms
- Generische Szenarien
 - ❖ Will man *mehrere* externe Szenarien zu *einem* (generischen) internen Szenario zusammenfassen, so müssen entsprechende *Fallunterscheidungen* in den Ablauf eingebaut werden.

OOA-Vorgehensmodell: Checkliste für interne Szenarien

▪ **Überprüfung und Konsolidierung des Modells**

- Sind alle *Klassen*, die für die internen Szenarien benötigt werden, im Klassenmodell bereits vorhanden?
 - ❖ Klassenmodell ggf. um zusätzliche Klassen ergänzen
- Ist für alle Nachrichten, die in den internen Szenarien auftreten, gewährleistet, dass das Senderobjekt über eine *Referenz* auf das Empfängerobjekt verfügt?
 - ❖ Die Referenz kann auf einer *statischen* Beziehung (*Assoziation*) beruhen oder *dynamisch* ermittelt werden (*Parameter, lokale Daten*)
- Für jede Nachricht, die ein Objekt während der Abwicklung eines internen Szenarios empfängt, muss die Klasse des Empfängerobjektes im Klassenmodell eine entsprechende *Operation* zur Verfügung stellen.
 - ❖ Klassenmodell um entsprechende Operationen ergänzen
 - ❖ Die Signatur der Operation kann zumindest teilweise aus den Nachrichtenparametern, die im internen Szenario auftreten, erschlossen werden.

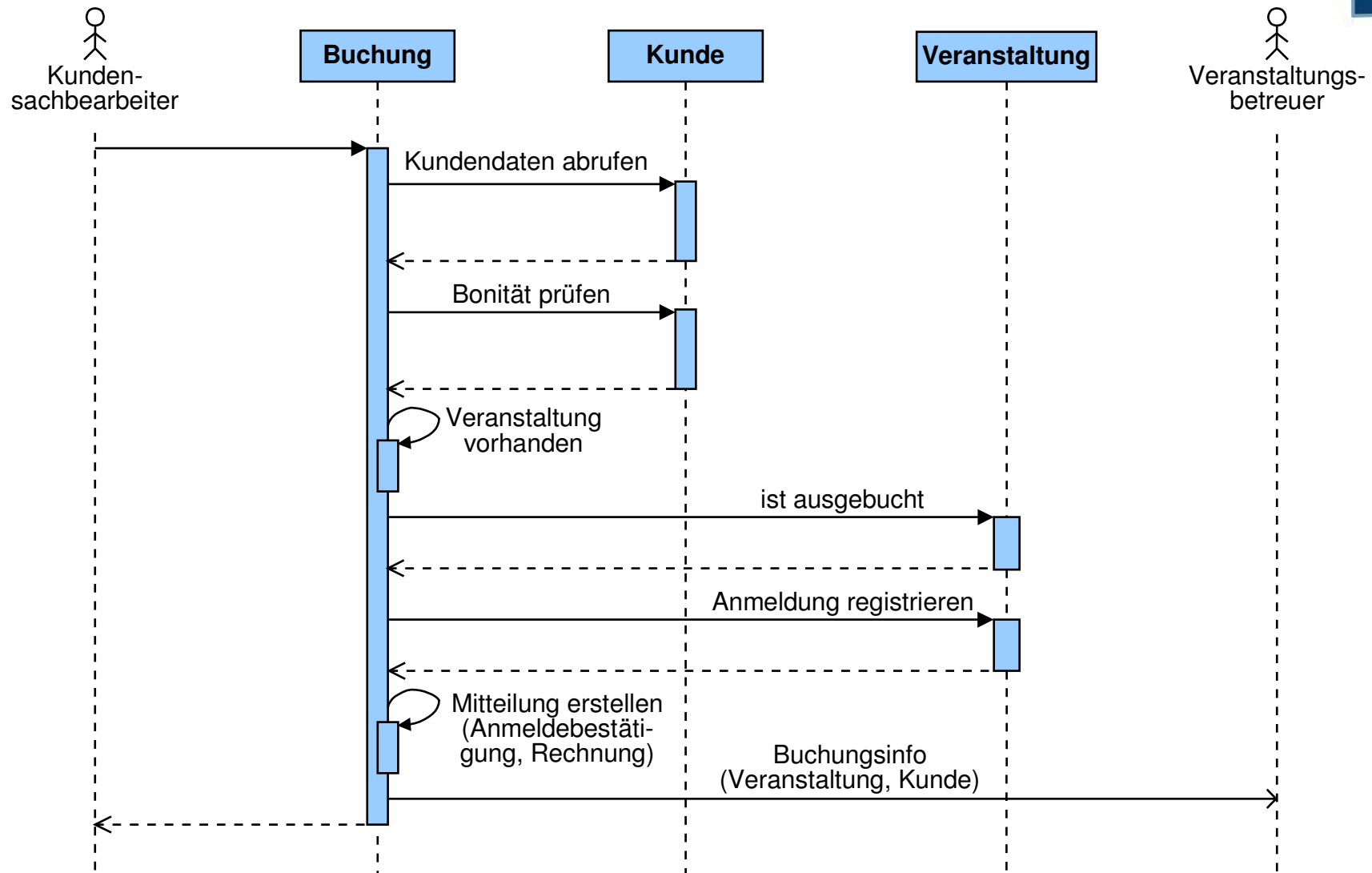
OOA-Vorgehensmodell: Szenario "Buchen"

aus dem Fallbeispiel Seminarorganisation (1)

Im Rahmen der Anforderungsdefinition wurde eines der externen Szenarien zum Anwendungsfall "Veranstaltung buchen" wie folgt beschrieben:

- **Voraussetzungen des Szenarios:**
 1. *Kunde ist Stammkunde (d.h. Kundendaten sind vorhanden).*
 2. *Die Bonitätsprüfung des Kunden verläuft positiv.*
 3. *Die zu buchende Veranstaltung ist vorhanden.*
 4. *Die zu buchende Veranstaltung ist nicht ausgebucht.*
- **Ablaufbeschreibung des Szenarios:**
 1. *Kundendaten abrufen*
 2. *Kundenbonität prüfen*
 3. *prüfen, ob Veranstaltung vorhanden*
 4. *prüfen, ob Veranstaltung ausgebucht*
 5. *Anmeldebestätigung und Rechnung erstellen*
 6. *Veranstaltungsbetreuer informieren*

OOA-Vorgehensmodell: Szenario "Buchen" aus dem Fallbeispiel Seminarorganisation (2)



OOA-Vorgehensmodell: Klassendiagramm für das Fallbeispiel Seminarorganisation (5. Iteration)

Veranstaltung
VeranstaltungsNr. Dauer Beginn Ende Agenda Veranstaltungsort TeilnehmerMax storniert
stornieren Anmeldg registrieren Abmeldg registrieren TlnListe erstellen ist ausgebucht alternative Veranst

Kunde
Kundennummer Name Adresse Geburtsdatum Funktion /Umsatz Kunde seit
KdDaten abrufen Bonität prüfen AdrAufkleber drucken

Firma
Kurzname Name Adresse Kontakt Ansprechpartner Abteilung Funktion /Umsatz Kunde seit

Dozent
Dozentennummer Name Adresse Geburtsdatum Biografie Honorar pro Tag Dozent seit

Seminartyp
Kurztitel Titel Zielsetzung Inhaltsübersicht Dauer Unterlagen Zielgruppe Gebühr TeilnehmerMax TeilnehmerMin

Buchung
angemeldet am bestätigt am Rechnungsversand am abgemeldet am Mitteilung am
anmelden abmelden Mitteilung erstellen Veranstaltg vorhanden

Mitarbeiter
Name Berechtigung Funktion Passwort

Zahlungsverzug
Rechnungsdatum Rechnungsbetrag

Firmenbuchung

OOA-Vorgehensmodell / Aktivität 6: Einführung von Gen/Spez-Beziehungen

- **Zielsetzung**

- Durch die Einführung von Gen/Spez-Beziehungen ins Strukturmodell sollen die *Zusammenhänge und Unterschiede konzeptionell verwandter Klassen* deutlich gemacht werden.

- **Vorgehensweise**

- Zum Auffinden potenzieller Gen/Spez-Beziehungen gibt es prinzipiell zwei verschiedene Wege:

- ❖ **Bottom-Up-Klassifikation (Generalisierung)**

Fragestellung: Gibt es im aktuellen Strukturmodell *gleichartige bzw. ähnliche Klassen*, aus denen sich durch Ausklammern gemeinsamer Eigenschaften bzw. gemeinsamen Verhaltens eine *gemeinsame Oberklasse* bilden lässt?

- ❖ **Top-Down-Klassifikation (Spezialisierung)**

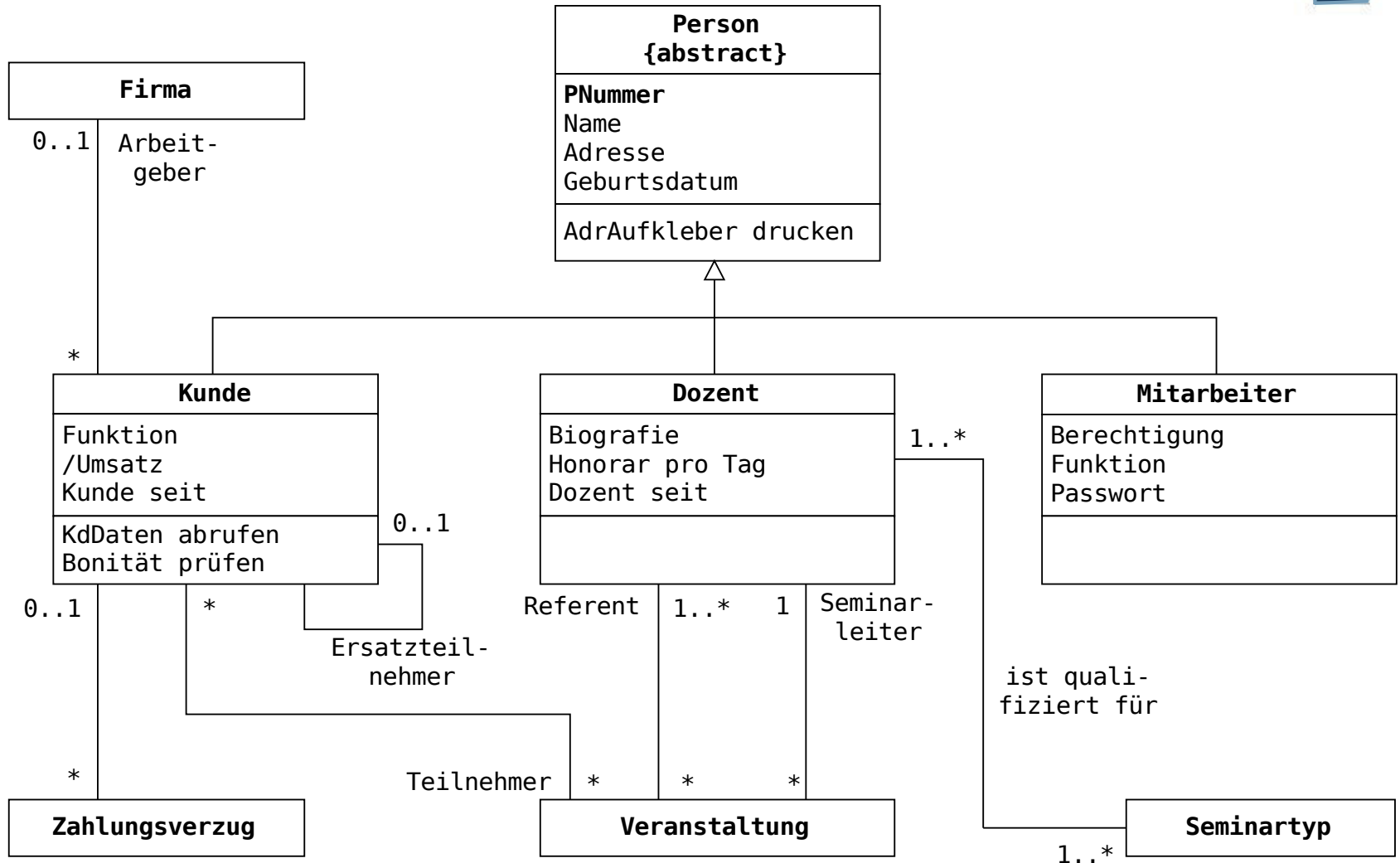
Fragestellung: Gibt es im aktuellen Strukturmodell Klassen, deren Objekte sich in *einigen* ihrer Eigenschaften bzw. in *einigen* Teilen ihres Verhaltens unterscheiden?

OOA-Vorgehensmodell / Aktivität 6 (Forts.): Bottom-Up-Klassifikation

▪ **Vorgehensweise**

- Gesucht werden Klassen mit signifikanten *Ähnlichkeiten* in ihren Eigenschaften bzw. in ihrem Verhalten.
 - ❖ Gibt es gemeinsame *Attribute*?
 - ❖ Gibt es gemeinsame *Assoziationen* / *Verknüpfungen* ?
 - ❖ Gibt es gemeinsame bzw. ähnliche *Operationen*?
- Mögliche Lösungsansätze:
 - ❖ Eine der bereits vorhandenen Klassen wird als Oberklasse gewählt, die übrigen (ähnlichen) Klassen werden zu deren Unterklassen.
 - ❖ Es wird eine neue (im Regelfall *abstrakte*) Klasse als gemeinsame Oberklasse gebildet.

OOA-Vorgehensmodell: Bottom-Up-Klassifikation im Fallbeispiel Seminarorganisation

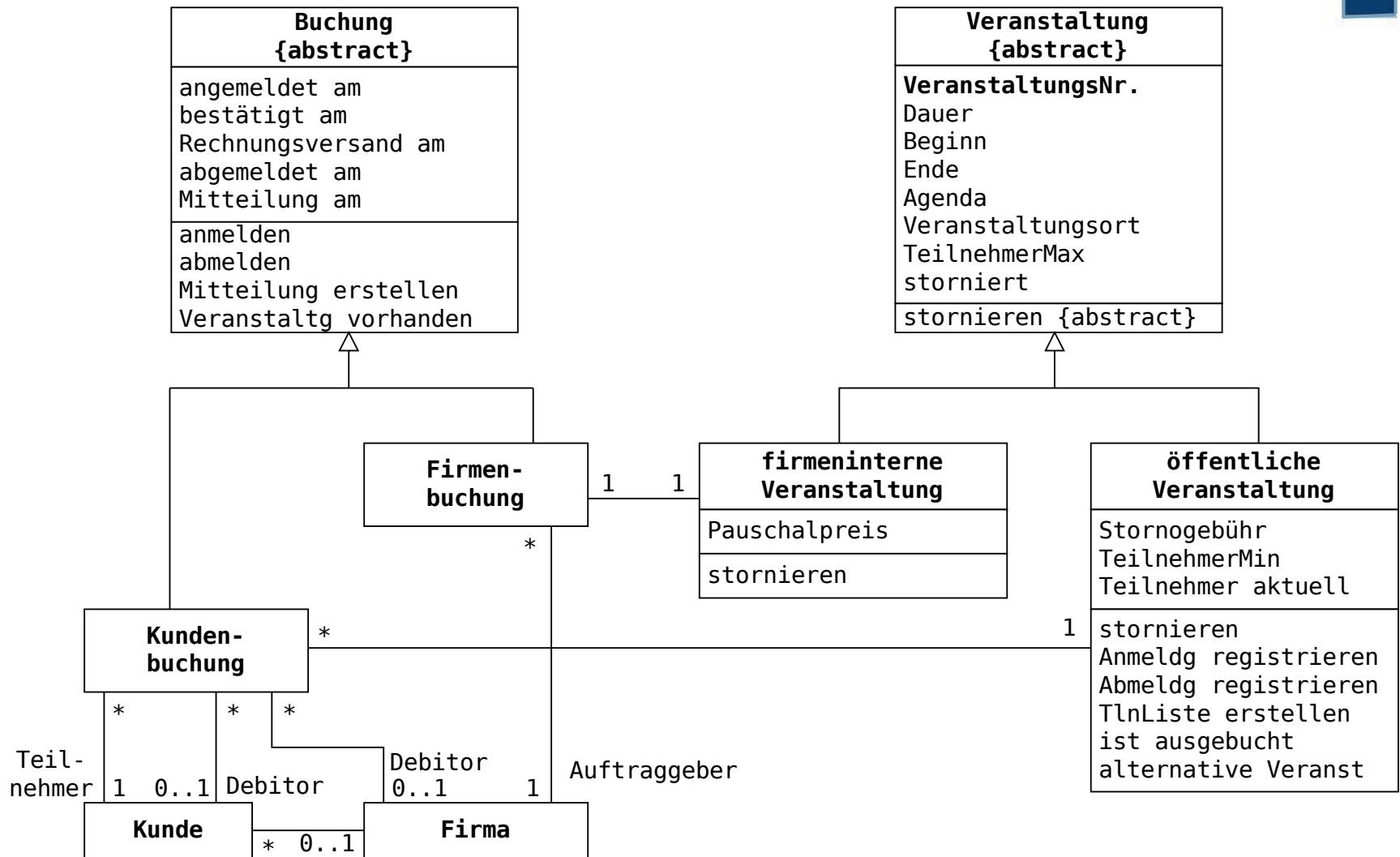


OOA-Vorgehensmodell / Aktivität 6 (Forts.): Top-Down-Klassifikation

▪ **Vorgehensweise**

- Gesucht werden Klassen, deren Objekte sich in *einigen* ihrer Eigenschaften bzw. in *einigen* Teilen ihres Verhaltens unterscheiden.
 - ❖ Gibt es Attribute oder Assoziationen, die für einige Objekte der Klasse *keine sinnvolle Semantik* haben?
 - ❖ Gibt es Operationen, die für einige Objekte der Klasse *nicht anwendbar* sind?
 - ❖ Gibt es Operationen, die für einige Objekte der Klasse *unterschiedliche Funktionalität* aufweisen?
- Diejenigen Teile der Eigenschaften bzw. des Verhaltens, die für alle Objekte *gleich* sind, verbleiben in der bereits bestehenden Klasse, die damit zur (abstrakten oder konkreten) Oberklasse wird.
- Attribute, Operationen und Assoziationen, die für einige Objekte der Klasse nicht sinnvoll definiert sind bzw. in denen sie sich unterscheiden, werden in geeignete, neu zu bildende Unterklassen „ausgelagert“.

OOA-Vorgehensmodell: Top-Down-Klassifikation im Fallbeispiel Seminarorganisation



OOA-Vorgehensmodell:

Checkliste für Gen/Spez-Beziehungen (1)

▪ **Fragestellungen zur Abgrenzung der beteiligten Klassen**

- Welche *Attribute* bzw. *Operationen* gehören zur gemeinsamen Oberklasse, welche verbleiben in den Unterklassen?
- Welche Operationen müssen in einer Unterklasse ggf. *redefiniert* werden?
- Ist die gemeinsame Oberklasse *abstrakt* oder *konkret*?
- Sind einzelne Operationen *abstrakt* oder *konkret*?
- Welche *Assoziationen* zu anderen Klassen sind sämtlichen Unterklassen gemeinsam (gehören also zur Oberklasse), welche Assoziationen bestehen nur für eine (oder mehrere) Unterklassen?
- Gibt es ein geeignetes *Diskriminatorattribut*?
 - ❖ Diskriminator = Attribut der Oberklasse, anhand dessen die Zugehörigkeit eines Objektes zu einer der Unterklassen eindeutig entschieden werden kann
 - ❖ Bsp.: Diskriminatorattribut Veranstaltungstyp als Attribut der Klasse Veranstaltung (Werte: intern, öffentlich)

OOA-Vorgehensmodell:

Checkliste für Gen/Spez-Beziehungen (2)

▪ **Überprüfung und Konsolidierung des Modells**

- Verbessert die Einführung der jeweiligen Gen/Spez-Beziehung das *Verständnis* des Modells?
- Entspricht die Gen/Spez-Beziehung dem „*natürlichen*“ Verständnis des Problembereichs?
- Handelt es sich wirklich um eine *konzeptionelle* „*is_a*“-Beziehung, oder handelt es sich lediglich um eine Form von Implementierungsvererbung?
- Charakterisieren die Unterklassen wirklich *unterschiedliche* Objekte, oder handelt es sich lediglich um verschiedene *Rollen* des gleichen Objektes?
 - ❖ Modellierung von Rollen durch verschiedene → *Schnittstellen* der gleichen Klasse!
- Benötigen alle Unterklassen wirklich *sämtliche* geerbten Eigenschaften?
- Umfasst die Gen/Spez-Hierarchie *maximal 3 bis 4 Hierarchiestufen*?