Fachhochschule
Dortmund
University of Applied Sciences and Arts

Softwaretechnik 1(A)

Modellkonsistenz

Autorin: Prof. Dr. Sabine Sachweh

Themen dieser Vorlesung

- **Überblick über die Diagramme**
- Abhängigkeiten und Inkonsistenzen
- Tipps für Klassendiagramme
- Beispiel: Buchhandlung
- Übungsaufgaben

Fachhochschule
Dortmund
University of Applied Sciences and Arts

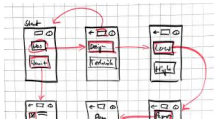
Modellkonsistenz
01: Überblick über die Diagramme

Fachhochschule Dortmund
University of Applied Sciences and Arts

we focus on students

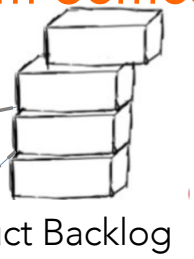
Projekt in diesem Semester

Product Vision



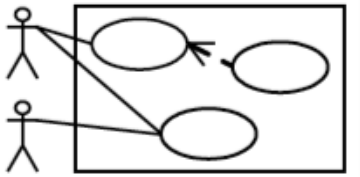
Product Protoype

Item/ Story Card



Product Backlog

Funktionale Anforderungen



Use Case Diagramm

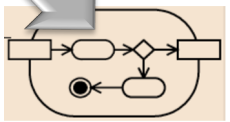
Struktur

Bounded Context

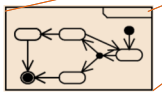
Verhalten

Text Text Text
Text Text Text
textuelle Beschreibung

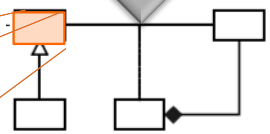
Anwendungsfall-spezifikations-schablone



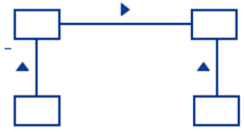
Aktivitäts-diagramm



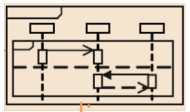
Zustands-diagramm



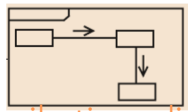
Klassendiagramm (Typebene)



Objektdiagramm (Instanzebene)



Sequenzdiagramm (Instanzebene)



Kommunikationsdiagramm (Instanzebene)

Fachhochschule Dortmund
University of Applied Sciences and Arts

we focus on students

Projekt in diesem Semester

Product Vision

Product Prototype

Item/ Story Card

Product Backlog

Funktionale Anforderungen

Use Case Diagramm

Struktur

Verhalten

Bounded Context

Text Text Text
Text Text Text
textuelle Beschreibung

Anwendungsfall-spezifikations-schablone

Aktivitäts-diagramm

Zustands-diagramm

Klassendiagramm (Typebene)

Objektdiagramm (Instanzebene)

Sequenzdiagramm (Instanzebene)

Kommunikationsdiagramm (Instanzebene)

Überblick über die UML-Diagramme

▪ Use Case Diagramm	Dynamisches Modell
▪ Klassendiagramm	Statisches Modell
▪ Objektdiagramm	Statisches Modell
▪ Aktivitätsdiagramm	Dynamisches Modell
▪ Sequenzdiagramm / Kommunikationsdiagramm	Dynamisches Modell
▪ Zustandsdiagramm	Dynamisches Modell

UML-Diagramme

Diagrammtyp	Kürzel	Diese zentrale Frage beantwortet das Diagramm	Stärken
Use-Case-Diagramm (dynamisch)	uc	Was leistet mein System für seine Umwelt (Nachbarsysteme, Stakeholder)?	Außensicht auf das System. Geeignet zur Kontextabgrenzung. Hohes Abstraktionsniveau, einfache Notationsmittel.
Klassendiagramm	---	Aus welchen Klassen besteht mein System und wie stehen diese untereinander in Beziehung?	Beschreibt die statische Struktur des Systems. Enthält alle relevanten Strukturzusammen-hänge/Datentypen. Brücke zu dynamischen Diagrammen. Normalerweise unverzichtbar.
Objektdiagramm	---	Welche innere Struktur besitzt mein System zu einem bestimmten Zeitpunkt zur Laufzeit (Klassendiagramm-schnappschuss)?	Zeigt Objekte und Attributbelegungen zu einem bestimmten Zeitpunkt. Verwendung beispielhaft zur Veranschau-lichung Detailniveau wie im Klassen-diagramm. Sehr gute Darstellung von Mengen-verhältnissen.

UML-Diagramme

Diagrammtyp	Kürzel	Diese zentrale Frage beantwortet das Diagramm	Stärken
Aktivitätsdiagramm	act	Wie läuft ein bestimmter flussorientierter Prozess oder ein Algorithmus ab?	Sehr detaillierte Visualisierung von Abläufen mit Bedingungen, Schleifen, Verzweigungen. Parallelisierung und Synchronisation. Darstellung von Datenflüssen.
Zustandsautomat	stm	Welche Zustände kann ein Objekt, eine Schnittstelle, ein Use Case, … bei welchen Ereignissen annehmen?	Präzise Abbildung eines Zustandsmodells mit Zuständen, Ereignissen, Nebenläufigkeiten, Bedingungen, Ein- und Austrittsaktionen. Schachtelung möglich.
Sequenzdiagramm	sd	Wer tauscht mit wem welche Informationen in welcher Reihenfolge aus?	Darstellung des Informationsaustauschs zwischen Kommunikationspartnern. Sehr präzise Darstellung der zeitlichen Abfolge.
Kommunikations-diagramm	sd	Wer kommuniziert mit wem? Wer „arbeitet“ im System zusammen?	Stellt den Informationsaustausch Zwischen Kommunikationspartnern dar.

Fachhochschule
Dortmund
University of Applied Sciences and Arts

Modellkonsistenz
02: Abhängigkeiten und Inkonsistenzen

Inkonsistenzen zwischen den Modellen

- Inkonsistenzen nicht vermeidbar
- teilweise sogar gewünscht
  - Vermeidung zu früher Design-Entscheidungen
  - Behebung in einigen Situation zu aufwendig

- Umgang mit Inkonsistenzen
  - Konsistenzregeln kontrollieren
  - Einschätzen
    - Auswirkung und Risiko
  - Verfolgen und Beheben
  - Messen

Klassifikation von Inkonsistenzen Klassen-, Sequenz- und Zustandsdiagrammen

Konfliktebene	Verhalten (Dynamisch)	Struktur (Statisch)
Modell-Modell		▪ unerfüllbare Assoziationen ▪ unspezifizierte Referenzen (Typen)
Modell-Instanz	▪ inkompatible Definitionen	▪ fehlende Instanzdefinition
Instanz-Instanz	▪ Aufrufinkonsistenz ▪ Verhaltensinkonsistenz ▪ Inkompatibles Verhalten	▪ unzusammen-hängende Modelle

[J. P. S. Wagemann, Consistency Maintenance of UML Models with Description Logics, Vrije Universiteit Brussel, 2003]

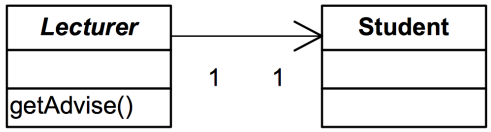
Klassifikation von Inkonsistenzen Klassen-, Sequenz- und Zustandsdiagrammen

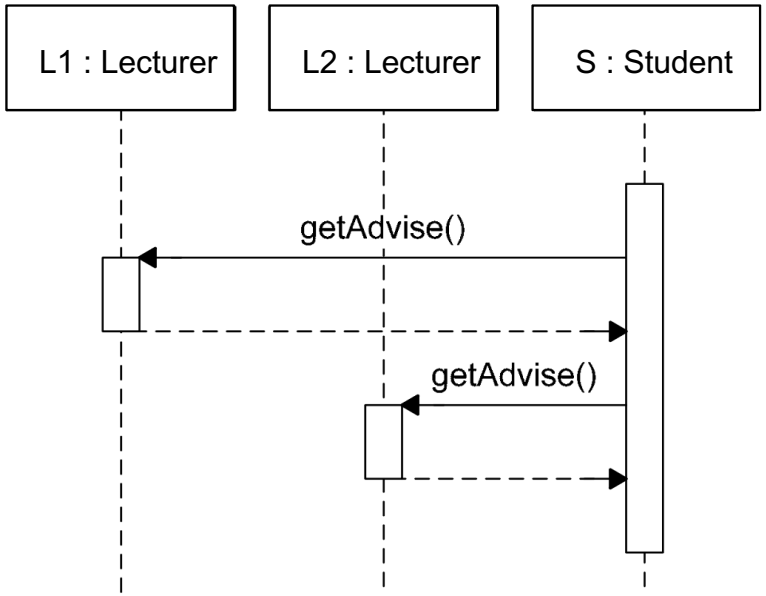
Konfliktebene	Verhalten (Dynamisch)	Struktur (Statisch)
Modell-Modell		▪ unerfüllbare Assoziationen ▪ unspezifizierte Referenzen (Typen)
Modell-Instanz	▪ inkompatible Definitionen	▪ fehlende Instanzdefinition
Instanz-Instanz	▪ Aufrufinkonsistenz ▪ Verhaltensinkonsistenz ▪ Inkompatibles Verhalten	▪ unszusammen-hängende Modelle

[J. P. S. Wagemann, Consistency Maintenance of UML Models with Description Logics, Vrije Universiteit Brussel, 2003]

Inkompatible Definitionen

Beispiel: Klassen- und Sequenzdiagramm

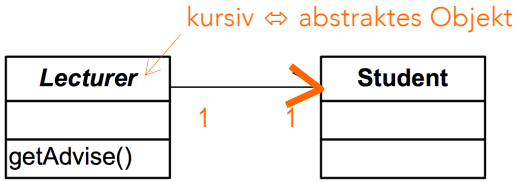


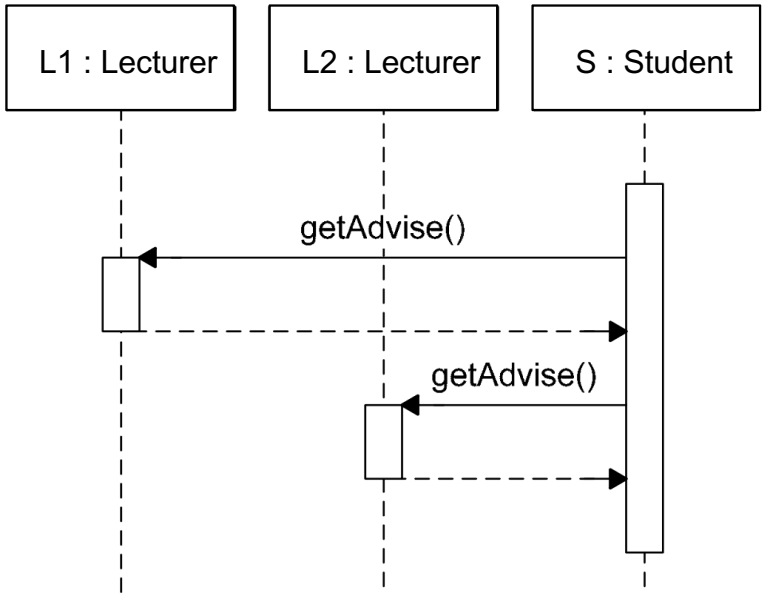


Gibt es hier Inkonsistenzen?

Inkompatible Definitionen

Beispiel: Klassen- und Sequenzdiagramm





Probleme:

- Multiplizität
- Navigierbarkeit
- Abstraktes Objekt

Verhaltenskonflikte auf Instanzebene

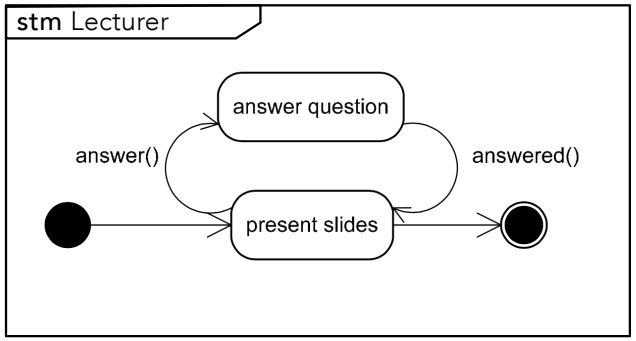
- **Aufrufinkonsistenz**
  - tritt beispielsweise immer dann auf, wenn ein Objekt einer Subklasse nicht verwendet werden kann, da es sich bei einer bestimmten Abfolge von Aufrufen nicht so verhält, wie es die vererbende Klasse spezifiziert.

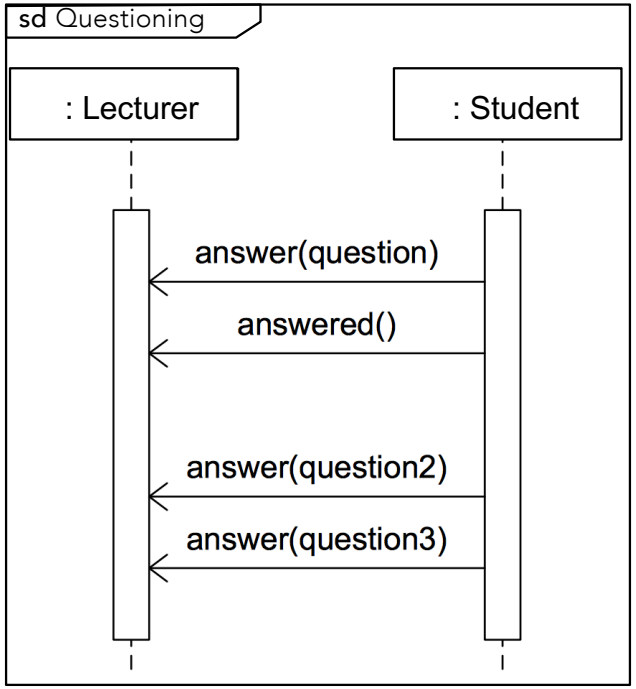
- **Verhaltensinkonsistenz**
  - wird beispielsweise die entgegengesetzte Richtung betrachtet und untersucht ob Subklassen dasselbe beobachtbare Verhalten zeigen wie ihre Elternklassen.

- **inkompatibles Verhalten**
  → Beispiel auf nächster Folie

Beispiel: Verhaltensinkonsistenz

Beispiel: Zustands- und Sequenzdiagramm





Probleme

- widersprüchliche Verhaltensdefinitionen

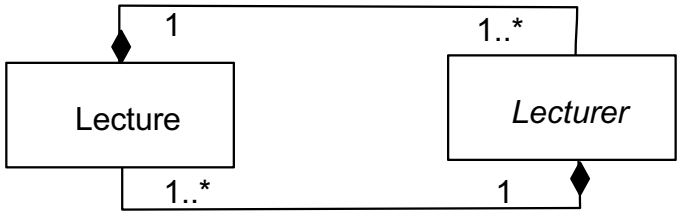
Klassifikation von Inkonsistenzen Klassen-, Sequenz- und Zustandsdiagrammen

Konfliktebene	Verhalten (Dynamisch)	Struktur (Statisch)
Modell-Modell		▪ unerfüllbare Assoziationen ▪ unspezifizierte Referenzen (Typen)
Modell-Instanz	▪ inkompatible Definitionen ✓	▪ fehlende Instanzdefinition
Instanz-Instanz	▪ Aufrufinkonsistenz ✓ ▪ Verhaltensinkonsistenz ✓ ▪ Inkompatibles Verhalten ✓	▪ unzusammen-hängende Modelle

[J. P. S. Wagemann, Consistency Maintenance of UML Models with Description Logics, Vrije Universiteit Brussel, 2003]

Unerfüllbare Assoziationen

Beispiel: Klassendiagramm

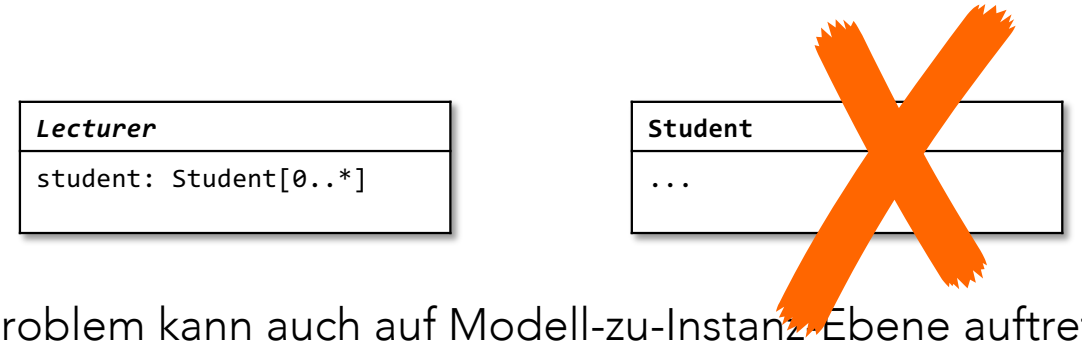


Probleme

- mit Realität nicht vereinbare Relationen
- Henne/Ei-Problem
- bzw. leere oder unendliche Menge

Unspezifizierte Referenzen (Typen)

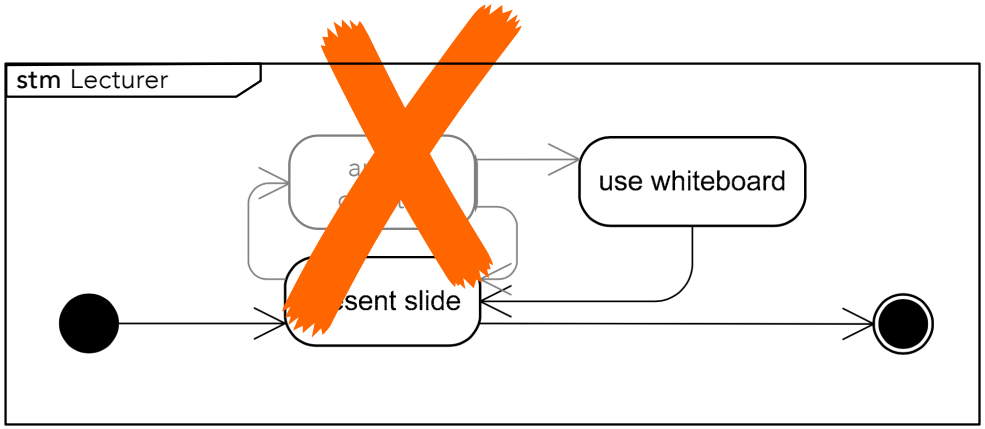
- **unspezifizierten Referenzen** *(Dangling References)*
  - tritt immer dann auf, wenn im Laufe der Entwicklung in einem Diagramm ein bestimmter Typ z.B. für ein Attribut verwendet, seine Klassendefinition aber später wieder entfernt wurde und die Referenz nun nicht mehr spezifiziert ist bzw. „hängt“.



- das gleiche Problem kann auch auf Modell-zu-Instanz-Ebene auftreten, wenn die Klasse eines Objekts noch nicht definiert oder bereits wieder entfernt wurde.
  → **fehlende Instanzdefinition**

Unzusammenhängende Modelle

Beispiel: Zustandsdiagramm



Problem

- Zustände oder Objekte nicht mehr erreichbar

Klassifikation von Inkonsistenzen Klassen-, Sequenz- und Zustandsdiagrammen

Konfliktebene	Verhalten (Dynamisch)	Struktur (Statisch)
Modell-Modell		▪ unerfüllbare Assoziationen ✓ ▪ unspezifizierte Referenzen (Typen) ✓
Modell-Instanz	▪ inkompatible Definitionen ✓	▪ fehlende Instanzdefinition ✓
Instanz-Instanz	▪ Aufrufinkonsistenz ✓ ▪ Verhaltensinkonsistenz ✓ ▪ Inkompatibles Verhalten ✓	▪ unzusammen-hängende Modelle ✓

[J. P. S. Wagemann, Consistency Maintenance of UML Models with Description Logics, Vrije Universiteit Brussel, 2003]

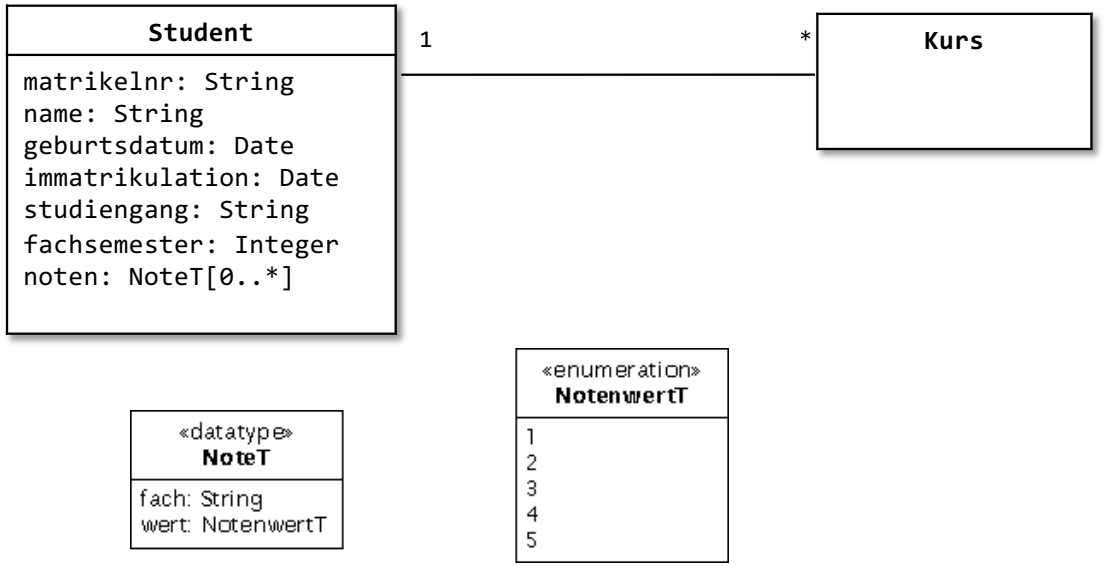
Fachhochschule
Dortmund
University of Applied Sciences and Arts

Modellkonsistenz
03: Tipps für Klassendiagramme

Allgemeine Tipps für Klassendiagramme

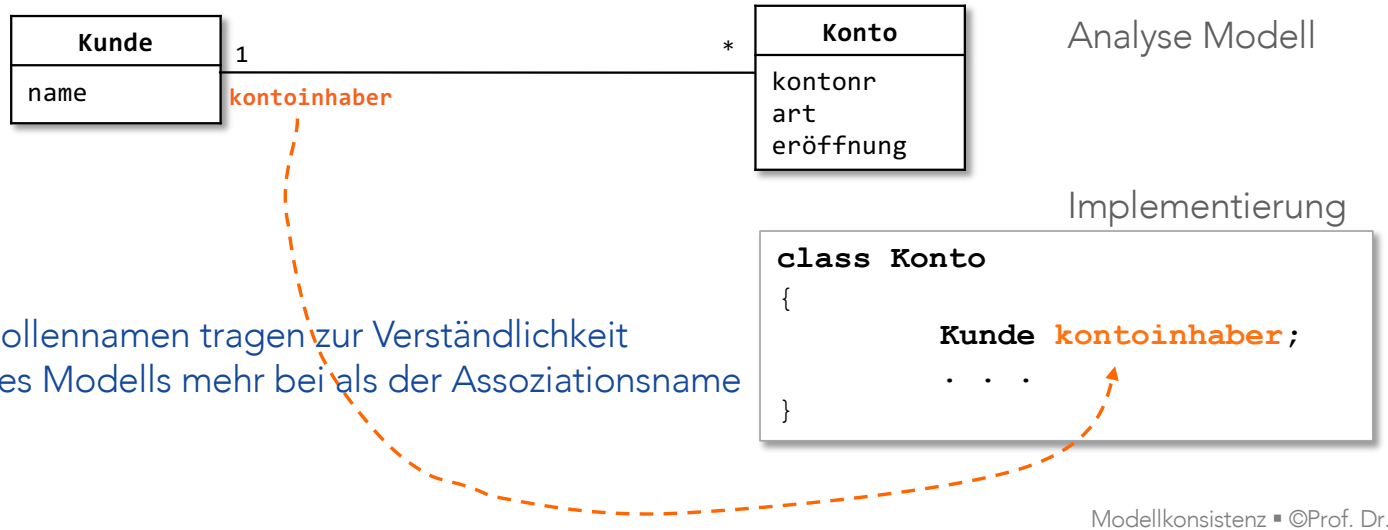
- Objektwertige Attribute nicht redundant durch Assoziation modellieren
- Selbstdefinierte Datentypen „isoliert“ darstellen
- Möglichst Rollennamen verwenden
- Multiplizitäten angeben

Klassendiagramm – Selbstdefinierte Datentypen



Klassendiagramm: Assoziation - Rollenname

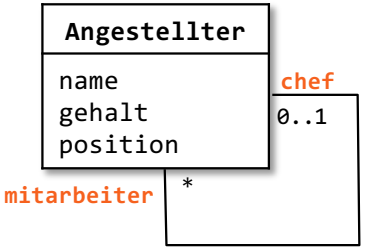
- Beschreibt Bedeutung eines Objekts in einer Assoziation
- Binäre Assoziationen besitzen maximal zwei Rollen
- Wird an das Ende der Assoziation bei der Klasse geschrieben, deren Bedeutung in der Assoziation die Rolle beschreibt



Klassendiagramm: Assoziation - Rollenname

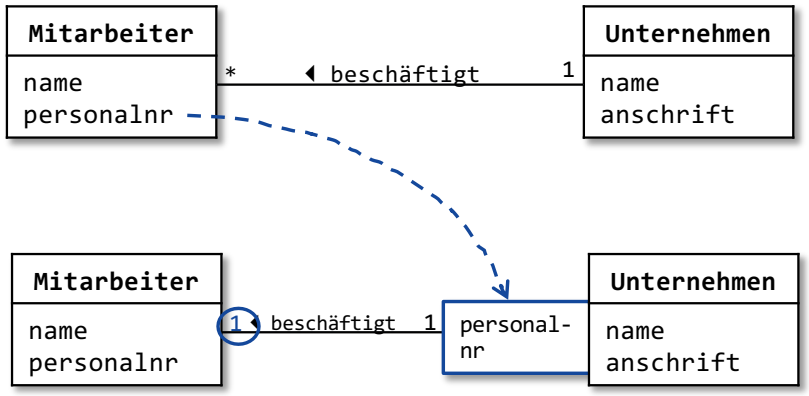
- Rollenname ist **nicht optional** ...
  - wenn zwischen zwei Klassen **mehr als eine Assoziation** besteht

  - bei **reflexiven Assoziationen**



Assoziation: qualifizierte Assoziation

- **Einteilung der Menge** der assoziierten Objekte **durch spezielles Attribut**, dessen Wert ein oder mehrere Objekte auf der anderen Seite selektiert
- Erhöhen den Informationsgehalt des Klassenmodells
- Nur in binären Assoziationen zulässig
- Das qualifizierende Attribut wird in einem Rechteck an der Seite der Klasse notiert



⇨Ein Unternehmen beschäftigt eine Menge von Mitarbeitern

⇨ Jeder Mitarbeiter gehört genau zu einem Unternehmen

⇨ Mitarbeiter werden über ihre Personalnummer identifiziert

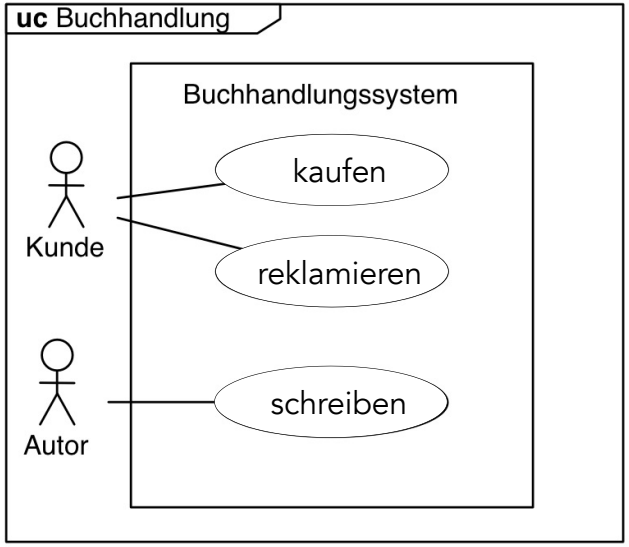
Qualifikationsangaben können die Kardinalität verändern!

Fachhochschule
Dortmund
University of Applied Sciences and Arts

Modellkonsistenz
04: Beispiel: Buchhandlung

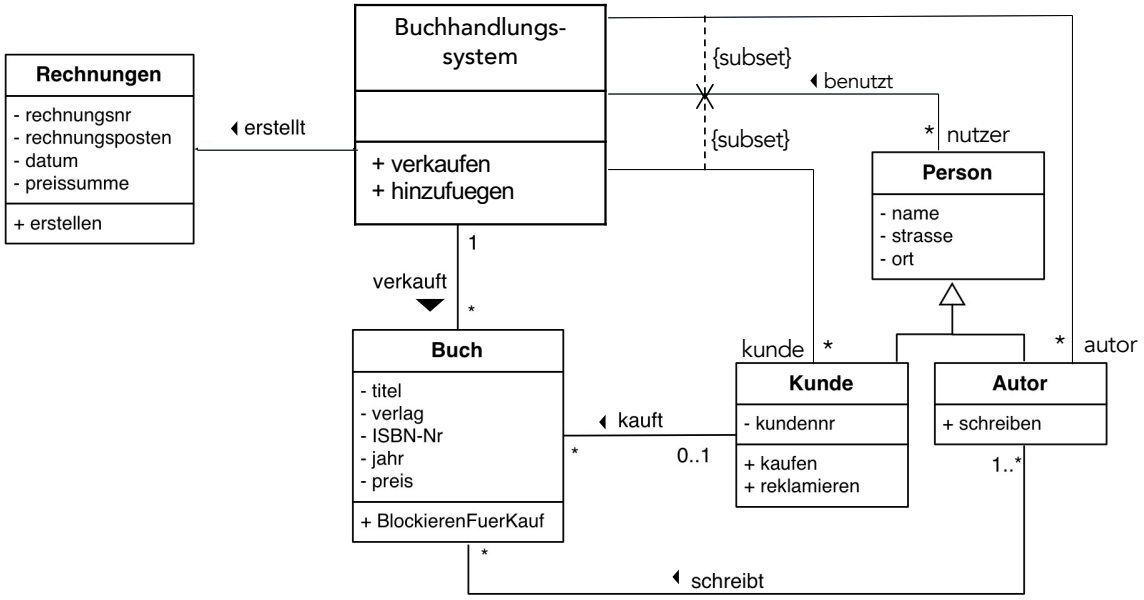
Beispiel: Buchhandlung

Use Case Diagramm



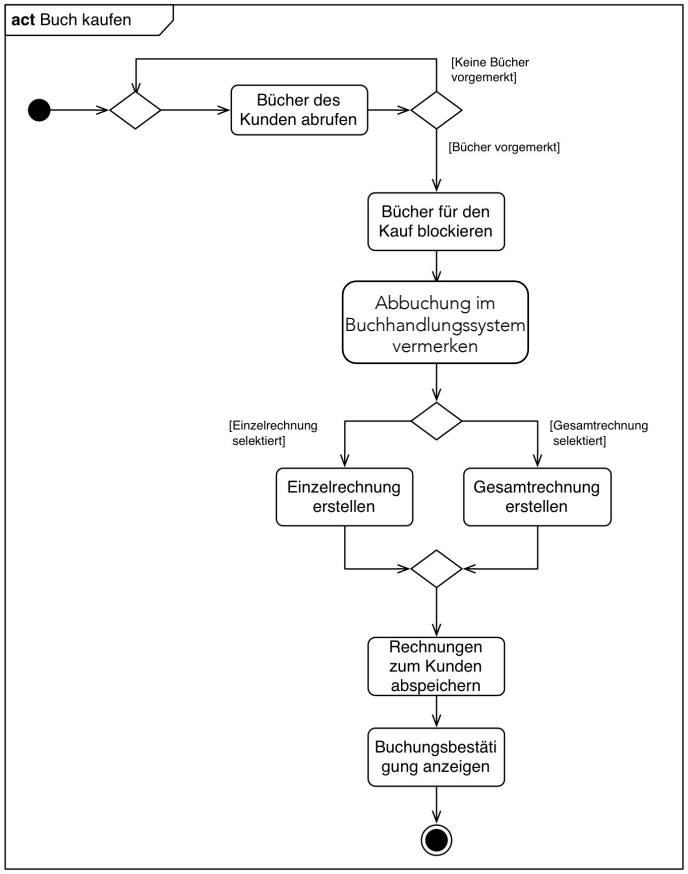
Beispiel Buchhandlung

Klassendiagramm



Beispiel: Buchhandlung

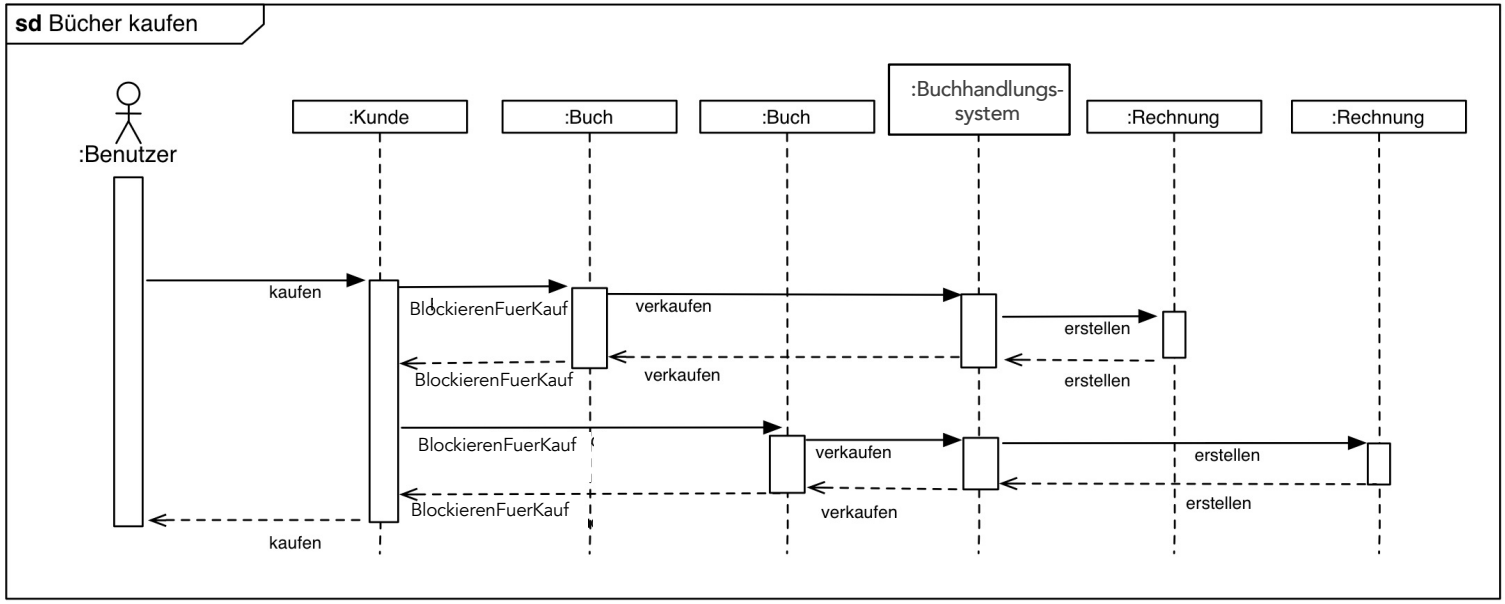
Aktivitätsdiagramm



- Müssen vorgemerkte Bücher verwaltet werden?
- Müssen Einzel- und Gesamtrechnungen unterschieden werden?

Beispiel: Buchhandlung

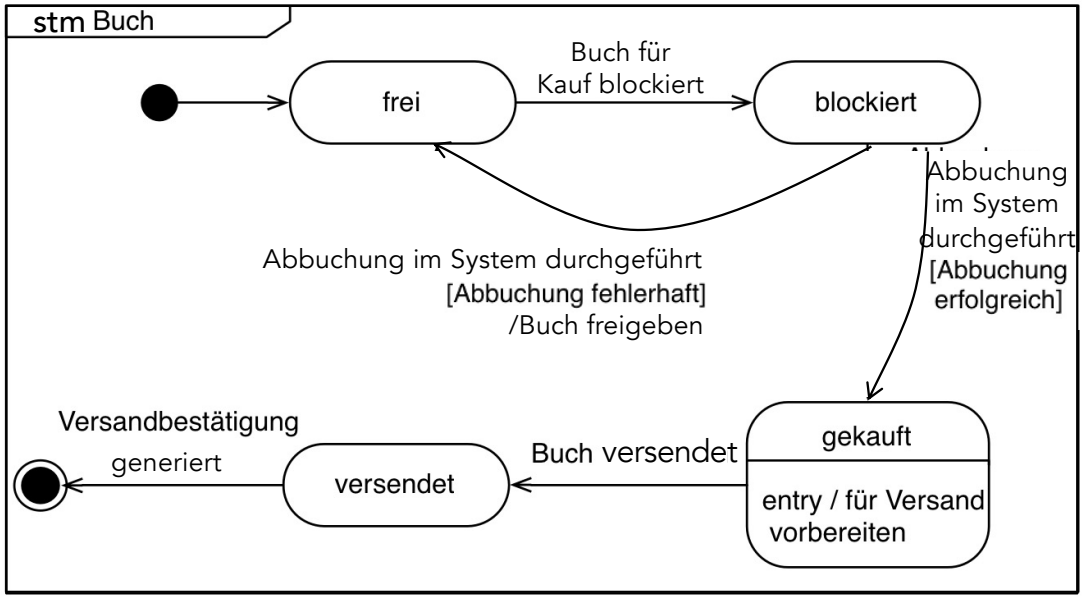
Sequenzdiagramm



Beispiel: Buchhandlung

Zustandsautomat

- Operation: „Buch freigeben“ fehlt im Klassendiagramm
- Operation: „für Versand vorbereiten“ fehlt im Klassendiagramm

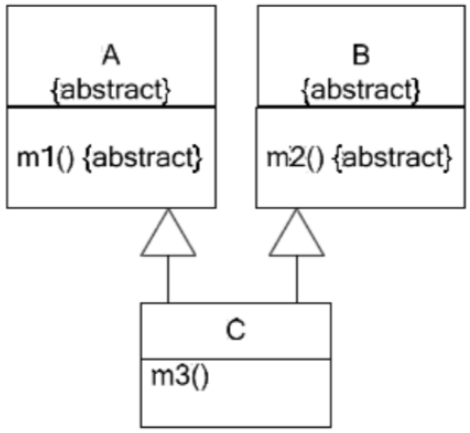


Fachhochschule
Dortmund
University of Applied Sciences and Arts

Modellkonsistenz
05: Übungsaufgaben

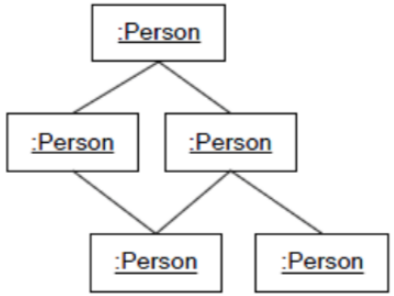
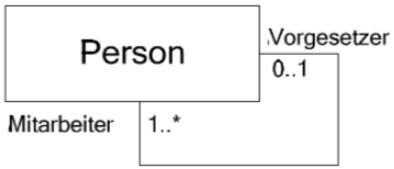
Aufgabe: Fehlerkorrekturen

Bewerten Sie folgende Modelle und geben im Fehlerfall eine Korrektur an:

Vererbung	**richtig ☐ falsch ☐**	**Korrektur**
	**Wenn falsch, warum?**	

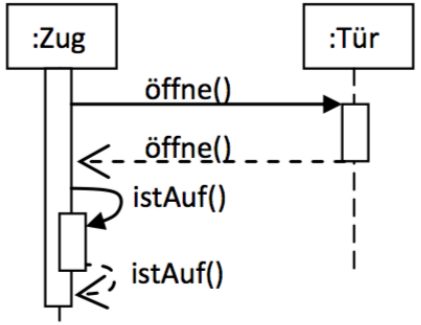
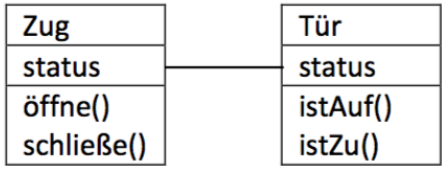
Aufgabe: Fehlerkorrekturen

Bewerten Sie folgende Modelle und geben im Fehlerfall eine Korrektur an:

Objekt- und Klassen-diagramm	**richtig ☐ falsch ☐**	**Korrektur**
 Hinweis: Das obige Objektdia-gramm ist korrekt. 	**Wenn falsch, warum?**	

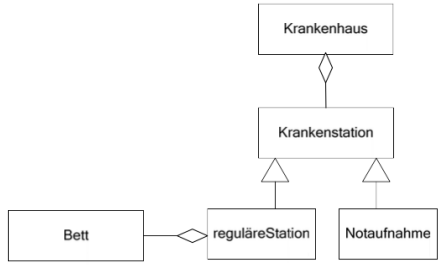
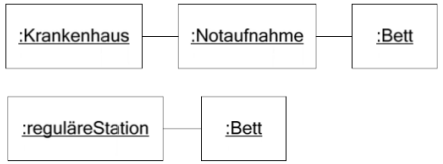
Aufgabe: Fehlerkorrekturen

Bewerten Sie folgende Modelle und geben im Fehlerfall eine Korrektur an:

Sequenzdiagramm	richtig ☐ falsch ☐	Korrektur
 Hinweis: Das obige Sequenzdiagramm ist korrekt. 	Wenn falsch, warum?	

Aufgabe: Fehlerkorrekturen

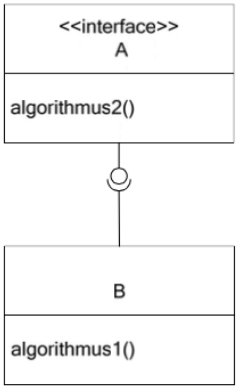
Bewerten Sie folgende Modelle und geben im Fehlerfall eine Korrektur an:

Klassendiagramm	richtig ☐ falsch ☐	Korrektur
 Hinweis: Das obige Klassen-diagramm ist korrekt. 	**Wenn falsch, warum?**	

Aufgabe: Fehlerkorrekturen

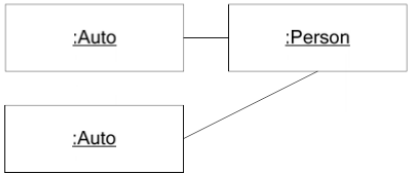
Bewerten Sie folgende Modelle und geben im Fehlerfall eine Korrektur an:

Klassendiagramm	**richtig ☐ falsch ☐**	**Korrektur**
	Wenn falsch, warum?	



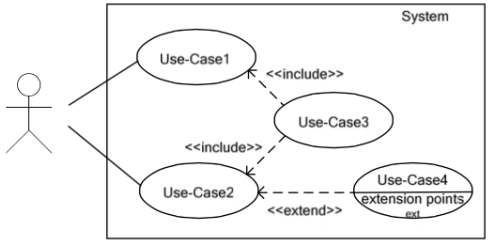
Aufgabe: Fehlerkorrekturen

Bewerten Sie folgende Modelle und geben im Fehlerfall eine Korrektur an:

Klassendiagramm	richtig ☐ falsch ☐	Korrektur
 Hinweis: Das obige Klassendiagramm ist korrekt. 	**Wenn falsch, warum?**	

Aufgabe: Fehlerkorrekturen

Bewerten Sie folgende Modelle und geben im Fehlerfall eine Korrektur an:

UseCase Diagramm	richtig ☐ falsch ☐	Korrektur
	Wenn falsch, warum?	

Aufgabe: Richtig oder falsch

(1) Welche der folgenden Aussagen zu Basiskonzepten der objektorientierten Analyse sind richtig?	
In einem Objekt werden Attribute definiert.	☐
Alle Objekte einer Klasse verwenden dieselben Operationen.	☐
Objekte sind identisch, wenn sie dieselben Attributwerte besitzen.	☐
Der Wert eines Abgeleiteten Attributs wird automatisch aus anderen Attributwerten berechnet.	☐
Eine Klassenoperation besitzt keine Implementierung und muss in den Unterklassen implementiert werden.	☐

Aufgabe: Richtig oder falsch

(2) Welche der folgenden Aussagen zu abstrakten Klassen sind richtig sind richtig?	
Von einer abstrakten Klasse können keine Objekte erzeugt werden.	☐
Eine abstrakte Klasse muss mindestens eine abstrakte Operation besitzen.	☐
Abstrakte Operationen müssen in der Unterklasse implementiert werden	☐
Eine abstrakte Klasse definiert nur Operationen für Unterklassen.	☐

Aufgabe: Richtig oder falsch

(3) Welche der folgenden Aussagen zu Szenarios sind richtig?	
Ein Szenario wird durch eine Menge von Anwendungsfällen dokumentiert.	☐
Szenarios beschreiben nur die erfolgreiche Bearbeitung eines Anwendungsfalls.	☐
Bei einem Szenario handelt es sich um ein dynamisches Konzept.	☐
Szenarios werden durch Zustandsdiagramme dargestellt.	☐
Ein Szenario ist eine spezifische Folge von Aktionen, die zur Verdeutlichung des Verhaltens eines Systems dient.	☐

Aufgabe: Richtig oder falsch

(4) Welche der folgenden Aussagen zu Zustandsautomaten sind richtig?	
Ein Zustandsautomat beschreibt den Lebenszyklus von Klassen.	☐
Eine Operation einer Klasse kann immer nur in einem Zustand aktiviert werden.	☐
Ein Zustandsübergang wird durch ein Ereignis ausgelöst.	☐
Ein Historienzustand ermöglicht beim Wiedereintritt in einen zusammengesetzten Zustand den automatischen Übergang in den ersten Unterzustand.	☐
Ein Zustand ist eine Zeitspanne, in der ein Objekt auf ein Ereignis wartet.	☐

www.fh-dortmund.de

facebook.com/fhdortmund instagram.com/fhdortmund twitter.com/fh_dortmund youtube.com/FachhochschuleDO