Praktische Informatik für Wirtschaftsmathematiker, Ingenieure und Naturwissenschaftler I (PIWIN I, 3 V + 1 Ü) WS 2002/03

14. Vorlesungswoche

Softwaretechnik: Methoden der objektorientierten Analyse

Unterlagen:

Gumm/Sommer, Kapitel 11

Helmut Balzert: Lehrbuch der Software-Technik 1 / 2,

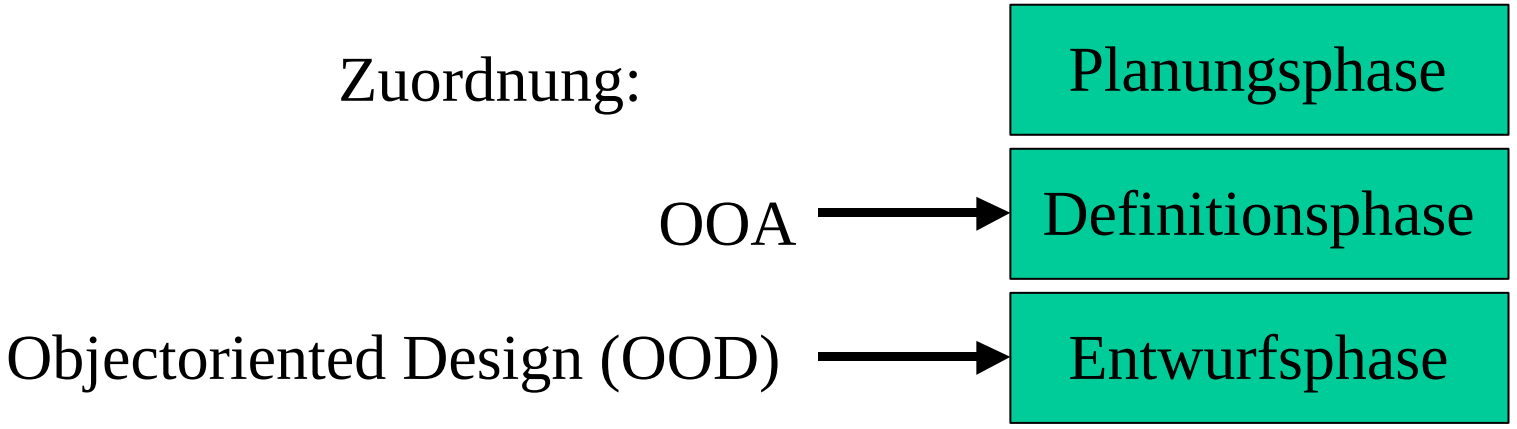
Spektrum Akademischer Verlag, Heidelberg u.a. 2001

UB: Lehrbuchsammlung L Sr 389-1

Folien nach Heinecke (FH Gelsenkirchen), Heyer (Uni Leipzig)

Methoden der Objektorientierten Analyse

- Grundbegriffe
- Anwendungsfalldiagramme
- Klassendiagramme
- Zustandsdiagramme
- Aktivitätsdiagramme
- Vorgehensweise bei Objektorientierte Analyse
- Anwendung der OOA



Grundbegriffe (Wiederholung aus OO Programmierung)

- Objekt
- Klasse
- Attribut
- Operation
- Botschaft
- Vererbung
- Polymorphismus

Begriffe sind im wesentlichen aus OO Programmierung bekannt, wesentlich:

Fokus auf vereinheitlichte graphische Beschreibung (UML)

später treten weitere Begriffe hinzu, die in OO Programmierung nicht auftreten: z.B. Assoziation

Objekt

- Individuelles Exemplar
  - Dinge, Personen, Begriffe der realen und der Vorstellungswelt
  - unterscheidbar von allen anderen Objekten: Identität
  - auch Exemplar oder Instanz (instance, class instance) genannt
- hat festgelegte Eigenschaften
  - Eigenschaften (Attribute) haben Werte: Daten
  - von außerhalb des Objektes nicht zugreifbar
- hat festgelegtes Verhalten
  - Verhalten folgt Algorithmen: Operationen
  - Operationen können von außen ausgelöst werden
  - Daten sollen nur über solche Operationen gelesen / geändert werden

$\Rightarrow$ Geheimnisprinzip

  - Daten sind verkapselt
  - nur über Operationen erreichbar

Darstellung eines Objekts

Objektname : Objektklasse
Attribut1: Wert1 Attribut2: Wert2 ...
Operation1 Operation2 ...

: Kunde	
anrede:	Frau
titel:	Dr.
vorname:	Erika
nachname:	Musterman
strasse:	Beispiel Str. 123
plz:	47111
ort:	Musterstadt
funktion:	testkunde
umsatz:	2.500,00 €
erfassen aktualisieren loeschen adresseDrucken umsatzErmitteln	

- Operationen können weggelassen werden, wenn sie im Kontext nicht erforderlich sind
- Attribute können weggelassen werden, wenn ihr Wert im Kontext unwesentlich ist
- Klassenangabe kann weggelassen werden, wenn sie klar ist
- Objektname kann weggelassen werden, wenn er irrelevant ist
- mindestens Objekt- oder Klassenname muss vorhanden sein

Klasse

- Gruppe von Objekten
  - mit gleichem Verhalten (Operationen)
  - mit gleichen Eigenschaften (Attributen)
  - mit unterschiedlichen Daten (Attributwerten)
- „Schablone“ für Objekte
  - spezifiziert Verhalten und Eigenschaften
  - besitzt Mechanismus, um Objekte zu erzeugen
  - Objekte wissen, zu welcher Klasse sie gehören
  - Klasse kann mit Hilfe einer Objektverwaltung wissen, welche Objekte zu ihr gehören
- Sonderfall: Abstrakte Klasse
  - spezifiziert Verhalten und Eigenschaften für Objekte untergeordneter Klassen
  - besitzt keinen Mechanismus, um Objekte zu erzeugen und daher keine Objekte

Darstellung einer Klasse

Klassenname
+ Attribut1: Typ1 # Attribut2: Typ2 - Attribut3: Typ3 ...
# Operation1 - Operation2 + Operation3 ...

Kunde
+ anrede: string + titel: string + vorname: string + nachname: string + strasse: string + plz: string + ort: string # funktion: string - umsatz: double
+ erfassen + aktualisieren # loeschen # adresseDrucken - umsatzErmitteln

- Operationen bzw. Attribute können weggelassen werden, wenn sie im Kontext nicht erforderlich sind
- ***Klassenname*** soll bei abstrakter Klasse kursiv sein
- Zusätzliche Angaben:
  - Typen der Attribute
  - Zugriffsmöglichkeiten auf Attribute und Operationen (public+,private-,protected#)
  - Schnittstellen der Operationen

Attribut

- Attribut
  beschreibt Eigenschaft der Objekte einer Klasse
  - alle Objekte haben dieselben Attribute
  - Geheimnisprinzip : Attribute nicht sichtbar für andere Klassen und deren Objekte (außer bei Vererbung)
- Attributwert ist Datum eines Objektes
  - Attributwerte desselben Attributs unterschiedlich in den verschiedenen Objekten
  - können nur über Operationen verändert werden
  - Muß-Attribute müssen immer einen Attributwert haben, Kann-Attribute nicht
- Schlüsselattribute
  - ermöglichen Identifizierung von Objekten
  - einzelnes Attribut oder (minimale) Attributkombination

Attribut (2)

- Angabe der Sichtbarkeit / des Zugriffs
  - uneingeschränkter Zugriff (public): +
  - Zugriff nur aus Operationen der eigenen Klasse (private): -
  - Zugriff aus untergeordneten Klassen (protected): #
- Angabe des Typs
  - verschiedene vordefinierte Typen, z.B.
    - string
    - int (verschiedene Ausprägungen)
    - float (verschiedene Ausprägungen)
    - fixed (verschiedene Ausprägungen)
    - boolean
    - date
    - time
  - selbstdefinierte Typen

Attribut (3)

- weitere sinnvolle Angaben (textuell)
  - Ergonomischer Name (auf der Benutzungsoberfläche)
  - Beschreibung als Text
  - Einheit (z.B. DM, kg)
  - Voreinstellung
  - intern (nicht auf der Benutzungsoberfläche)?
  - transient (nicht in der Datenbank zu speichern)?
  - Schlüssel?
  - Kann-Attribut?
  - Restriktionen (bzgl. Operationen oder anderer Attribute)
  - nur lesender Zugriff?
  - Struktur (Notation wie in Data Dictionary)

Attribut (4)

- Klassenattribut
  beschreibt Eigenschaft der Klasse
  - nur ein Attributwert für alle Objekte der Klasse
  - unabhängig davon, ob Objekte zu der Klasse existieren
  - meist gekennzeichnet durch (K) hinter dem Namen

Operation

- Operationen (auch Methoden genannt) beschreiben Verhalten der Objekte einer Klasse
  - Operation realisiert einen Algorithmus
  - Operation wird von außen über Botschaft aufgerufen
  - Operation kommuniziert mit Aufrufer (Sender) über Ein- und Ausgabeparameter
- Signatur einer Operation
  - Name der Operation (i.d.R. Verb, da Tätigkeit)
  - Ein-/Ausgabe-Schnittstelle (Parameter, i.d.R. mit Typangabe): Operationsname (**in** Par11, Par21, ...; **out** Par21, Par22,...)
  - Ausnahmebehandlung im Fehlerfall

Operation (2)

- Spezifikation
  - Beschreibung des Algorithmus aus fachlicher Sicht
  - teils Standardbefehle:
    - Klassenname.create (Attribut1, Attribut2,...)
    - Klassenname.delete ()
    - Klassenname.setAttributname ()
    - Klassenname.getAttributname ()
  - teils freier Text
    - Klassenname.operationsname (in ..., out ...)

Operation (3)

- Klassenoperation
  - Mögliche Fälle:
    - Manipulation eines Klassenattributs
    - Bearbeitung aller Objekte der Klasse
    - Selektion von Objekten der Klasse
  - Operation wird meist mit (K) gekennzeichnet wie bei Attributen

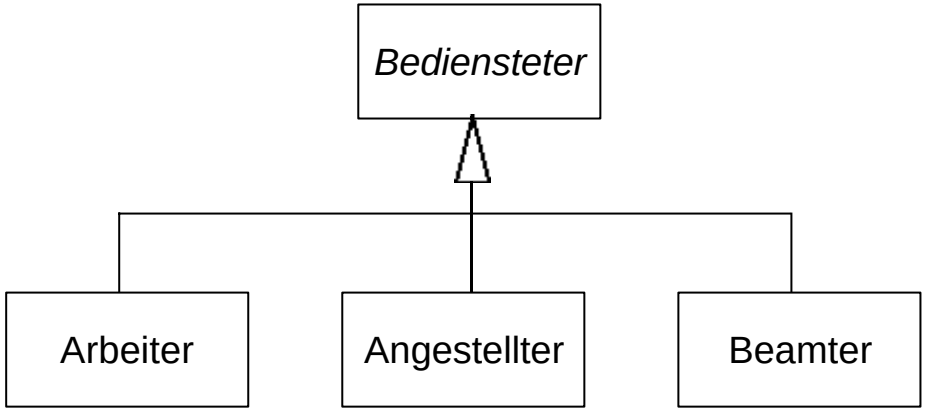
Botschaft

- Botschaft ist
  Aufforderung zur Erbringung einer Dienstleistung
  - Sender (*client*) sendet Botschaft an Empfänger (*server*)
  - Empfänger interpretiert Botschaft und führt Operation aus
  - Botschaft besteht aus Operationsnamen und Argumenten
  - Empfänger kann Ergebnisse zurückgeben
  - Sender weiß nicht, wie Empfänger die Operation ausführt
- Protokoll
  - Menge der Botschaften,
    die an Objekte einer Klasse gesendet werden können
  - Erhält Objekt eine Botschaft, zu der es keine Operation besitzt,
    sucht es die Operation in der direkt übergeordneten Klasse
  - Sender und Empfänger sind meist Objekte,
    können aber auch Klassen sein
- Modellierung der Kommunikation mit Botschaften
  - Sequenzdiagramm, Kollaborationsdiagramm (➔Verhaltensdiagramme)

Vererbung

- Vererbung ist eine Beziehung zwischen Klassen
  - Klasse2 verfügt über Eigenschaften und Verhalten von Klasse1: Klasse2 erbt von Klasse1
  - Klasse2 kann von Klasse1 ererbtes Verhalten redefinieren
  - Vererbung kann über mehrere Stufen erfolgen
  - Vererbung definiert Klassenhierarchie (Vererbungsstruktur)
  - Oberklassen sind Klassen, von denen eine Klasse K erbt (direkt oder über andere Oberklassen)
  - Unterklassen sind Klassen, die von einer Klasse K erben (direkt oder über andere Unterkalssen)
- Einfachvererbung
  - Jede Klasse hat höchstens eine direkte Oberklasse

Darstellung von Vererbung

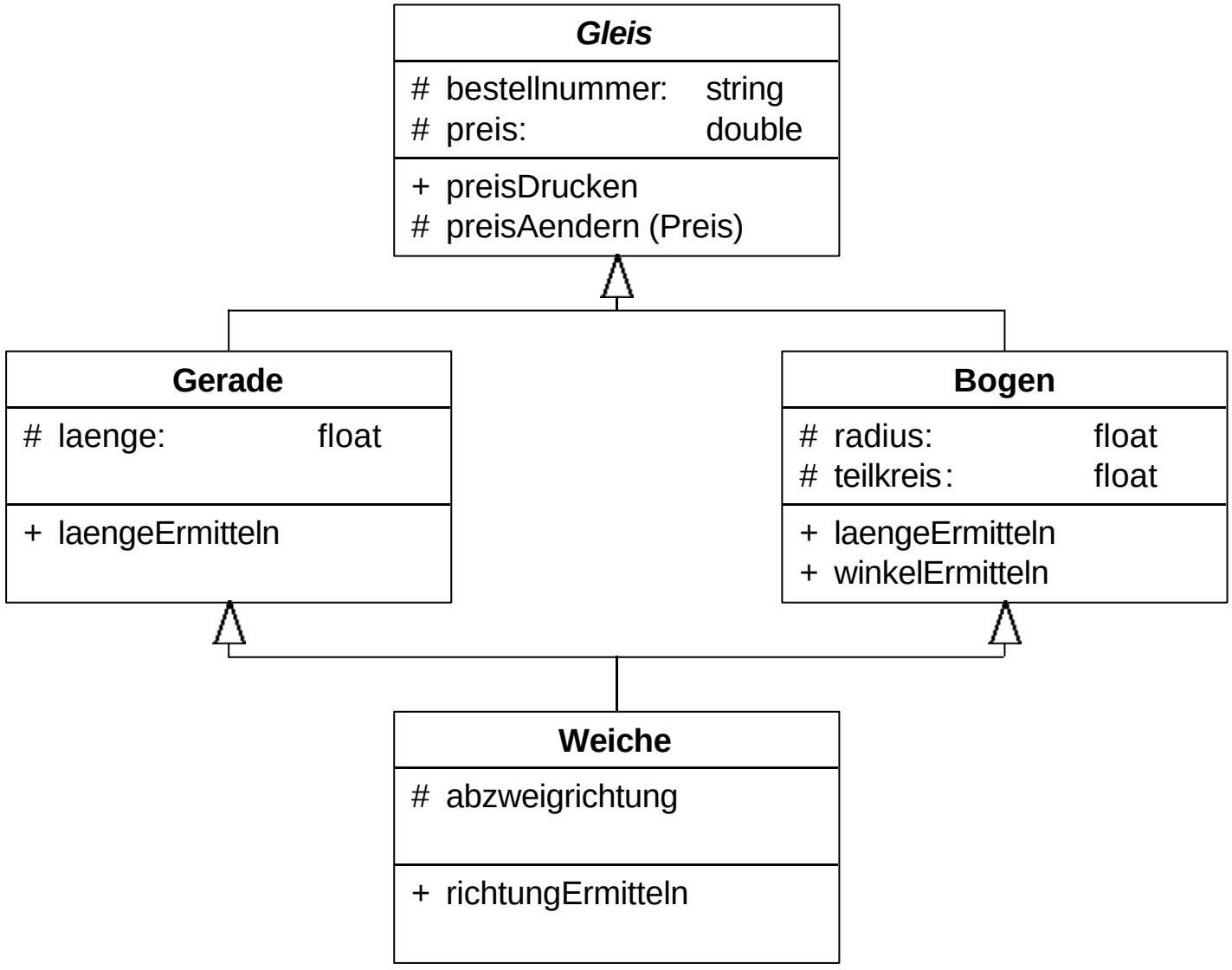


- Verbindung durch Linien
- Oberklassen möglichst über Unterklassen
- Pfeil in Richtung auf die Oberklasse

Vererbung (2)

- Redefinition von Operationen
  - Operation mit gleichem Namen wie in Oberklasse
  - Schnittstelle muß kompatibel sein
    - gleiche Anzahl Parameter
    - gleiche Typen bei Parametern oder Untertypen
  - Bedingungen der Operation müssen kompatibel sein
    - Beibehaltung oder Lockerung von Vorbedingungen
    - Beibehaltung oder Verschärfung von Nachbedingungen
  - in der Unterklasse
  - Operation in der Unterklasse ist Spezialisierung
- Mehrfachvererbung
  - Klasse kann mehrere direkte Oberklassen haben
  - Namenskonflikte bei geerbten Attributen oder Operationen möglich

Beispiel für Vererbung

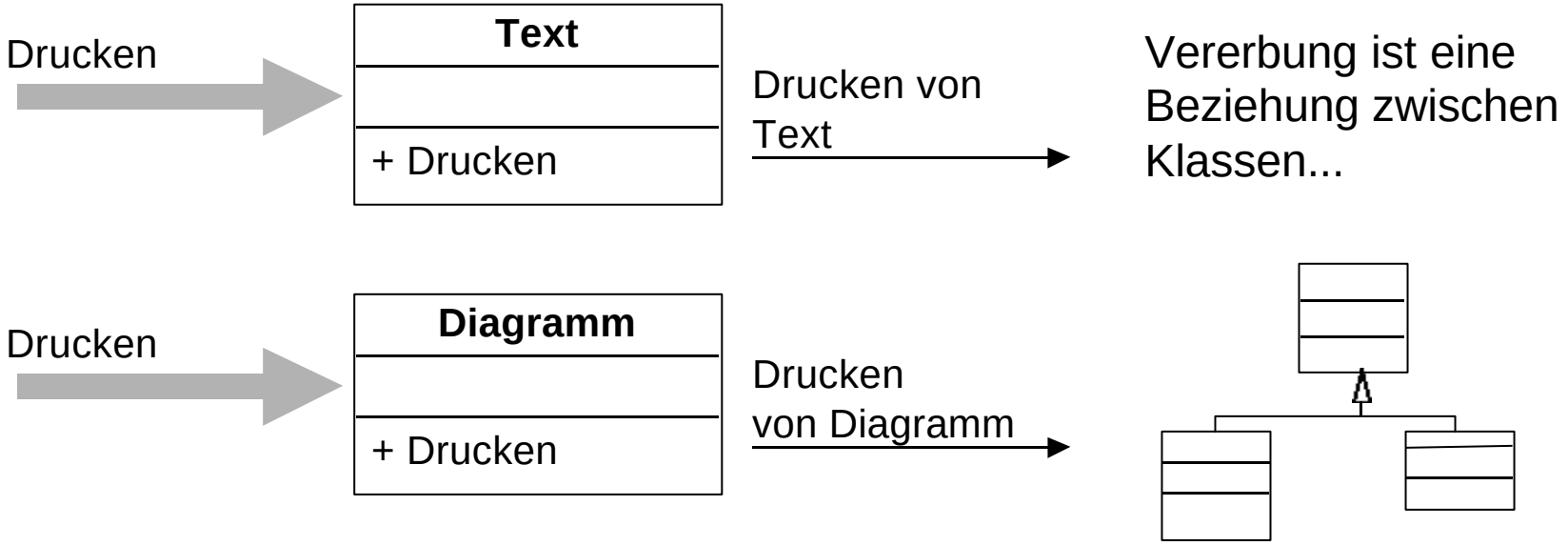


Vererbung (3)

- Vererbung
  - ☺ unterstützt Änderbarkeit
  - ☺ erlaubt Generalisierung / Spezialisierung
  - ☹ verletzt Geheimnisprinzip

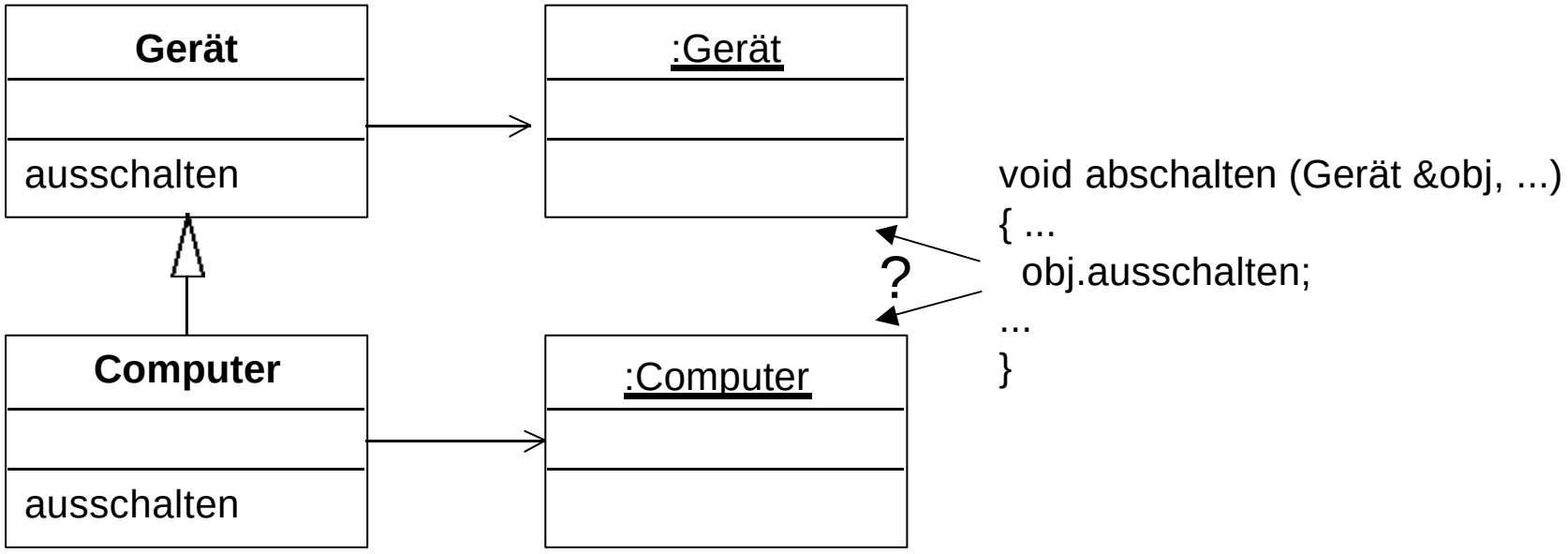
Polymorphismus

- Dieselbe Botschaft an Objekte verschiedener Klassen
  - unterschiedliche Operationen möglich
  - Sender braucht Klasse des Empfangsobjekts nicht zu kennen
  - Sender braucht Ablauf der Operation nicht zu kennen



Polymorphismus (2)

- Spätes Binden
  - Operation sendet Botschaft „opname“ an Objekt einer Klasse K1
  - Klasse hat Unterklasse K2, die Operation „opname“ redefiniert
  - erst zur Laufzeit kann entschieden werden, ob „opname“ aus K1 oder „opname“ aus K2 auszuführen ist



Anwendungsfalldiagramme

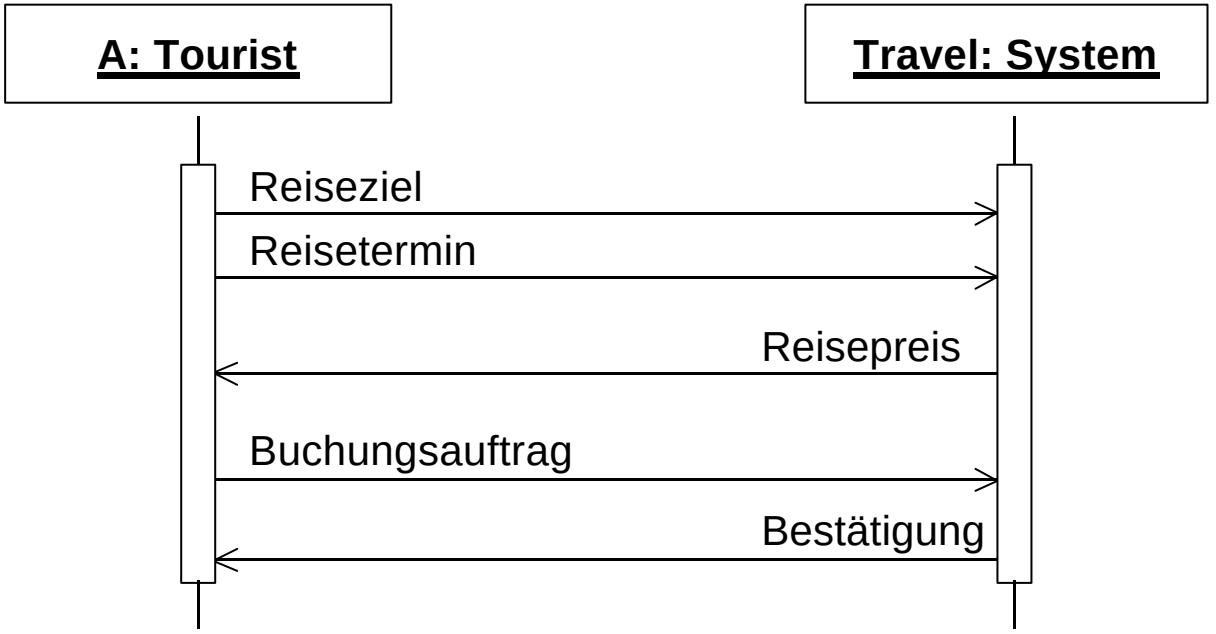
- Begriffe
- Darstellung von Szenarien
- Erweiterte Sequenzdiagramme
- Darstellung von Anwendungsfällen
- Stereotypen
- Syntaktische Regeln

Begriffe

- Szenario
  ist spezifische Folge von Aktionen, die zur Verdeutlichung des Systemverhaltens dient
  - Ausgangspunkt der Spezifikation
  - beschreibt Einzelfall
- Anwendungsfall
  ist Verallgemeinerung der Menge der Szenarien
  - beschreibt Gesamtmenge der Fälle
  - berücksichtigt Bedingungen und Unterschiede
  - entspricht dem Kontextdiagramm der Strukturierten Analyse
- Akteur
  ist außerhalb des Systems liegende Rolle
  - Benutzer oder System
  - löst Anwendungsfall aus

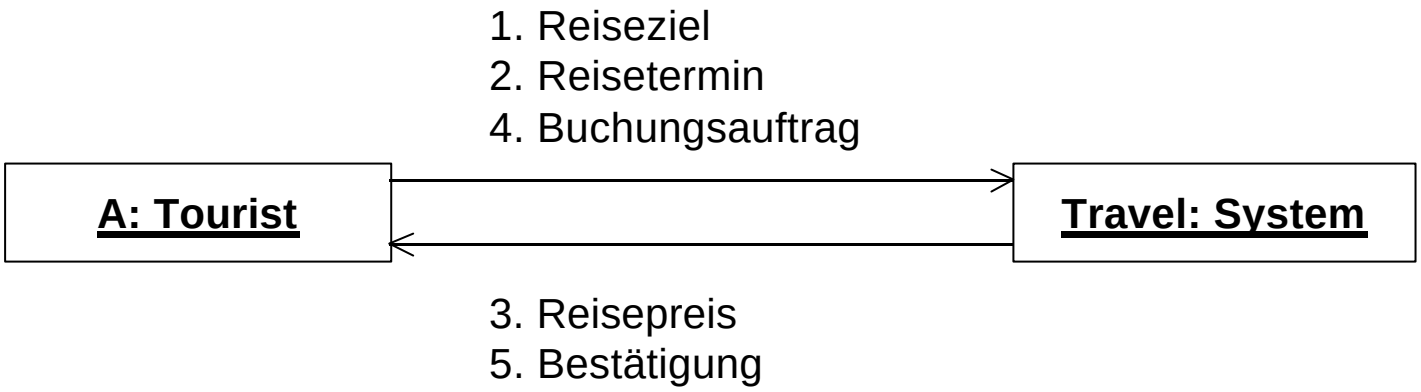
Darstellung von Szenarien

- Szenario
  - textuelle Beschreibung
    - Eine Touristin A übermittelt erst das Reiseziel und dann den Reisetermin an das Informationssystem Travel. Das System nennt ihr darauf den Preis der Reise. Sie ist einverstanden und gibt den Buchungsauftrag. Nach dessen Eingang erhält sie eine Bestätigung.
  - Sequenzdiagramm (Abläufe)



Darstellung von Szenarien (2)

- Kollaborationsdiagramm (Informationsaustausch)



- Bezeichnungen der Objekte

O	Objekt O
O:C	Objekt O der Klasse C
O/R	Objekt O in der Rolle R
O/R:C	Objekt O der Klasse C in der Rolle R
:C	unbenanntes Objekt der Klasse C
/R	unbenanntes Objekt in der Rolle R
/R:C	unbenanntes Objekt der Klasse C in der Rolle R

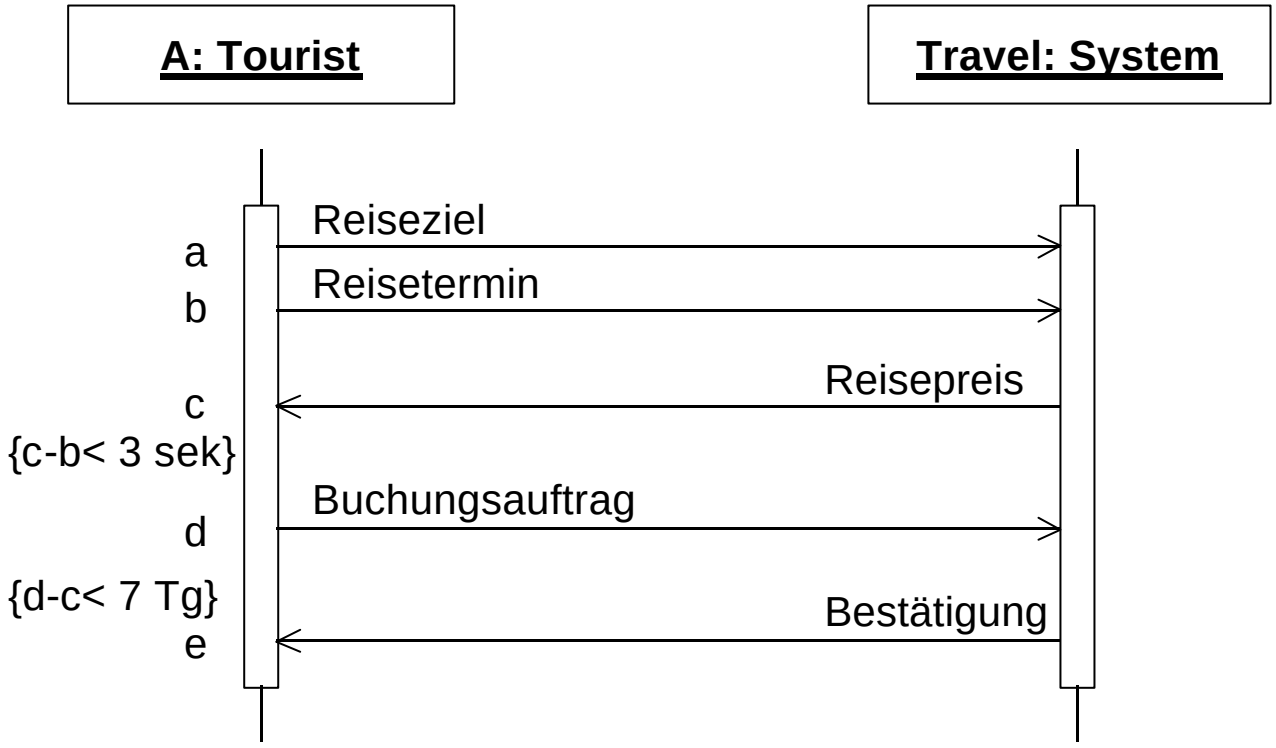
z.B. A/Tourist:Person (hier als Akteur)

Erweiterte Sequenzdiagramme

- Erweiterung zum Anwendungsfall
  - textuelle Beschreibung
    - Eine Touristin übermittelt erst das Reiseziel und dann den Reisetermin an das Informationssystem. Das System nennt ihr darauf den Preis der Reise. Falls sie einverstanden ist, gibt sie den Buchungsauftrag. Nach dessen Eingang erhält sie eine Bestätigung.
  - textuelle Beschreibung mit Zeitbeschränkungen
    - Eine Touristin übermittelt erst das Reiseziel und dann den Reisetermin an das Informationssystem. Das System nennt ihr darauf *innerhalb von 3 Sekunden* den Preis der Reise. Falls sie einverstanden ist, gibt sie *innerhalb von 7 Tagen* den Buchungsauftrag. Nach dessen Eingang erhält sie eine Bestätigung.

Erweiterte Sequenzdiagramme (2)

– Sequenzdiagramm mit zeitlichen Einschränkungen
  • verbindet Szenario und Anwendungsfall

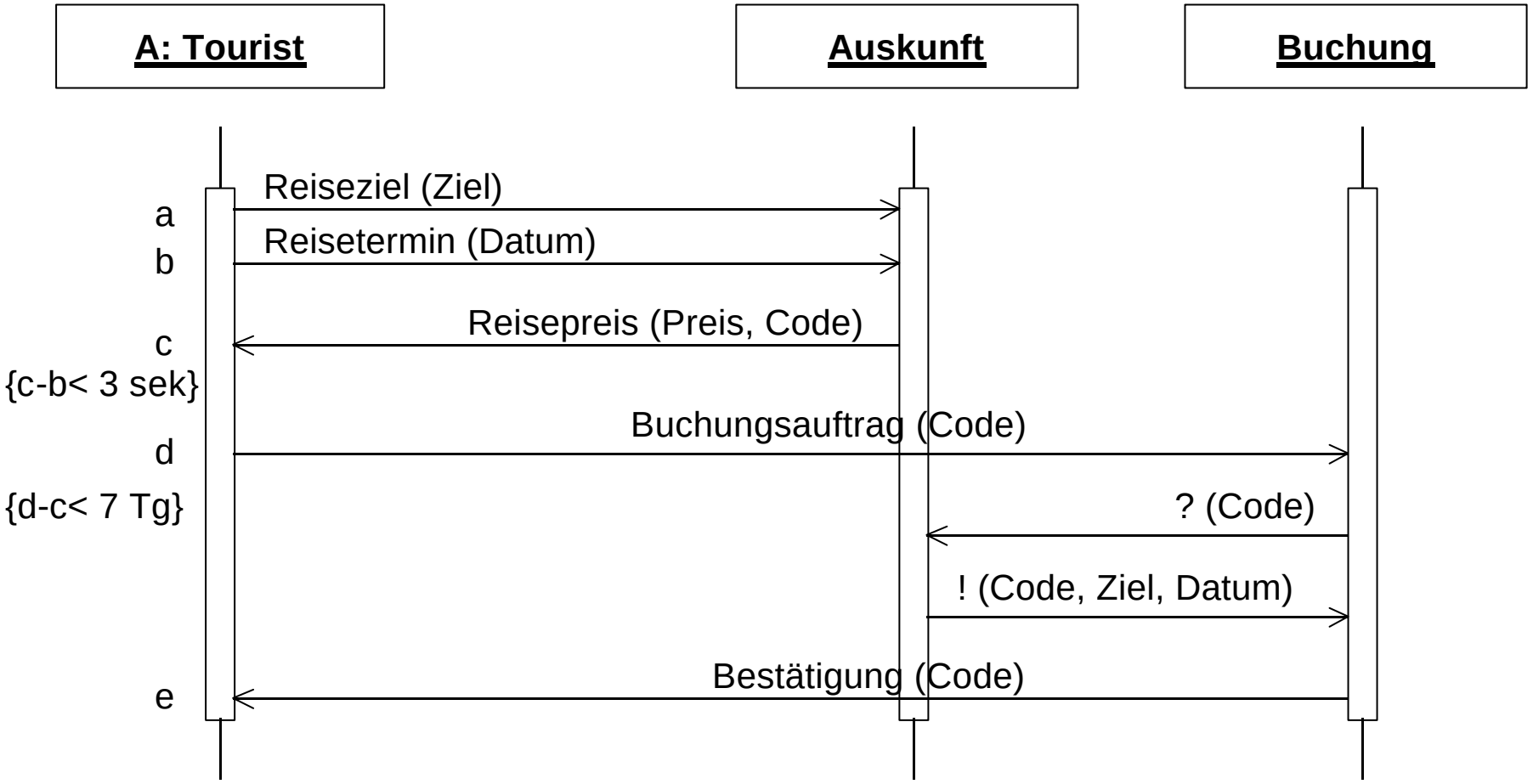


Erweiterte Sequenzdiagramme (3)

- Verfeinerung von Szenarien
  - vertikale Verfeinerung
    - Botschaften aufteilen, z.B. Reiseziel in Ort und Hotel
    - in der Regel kein neues Diagramm erforderlich
  - horizontale Verfeinerung
    - Objekte / Klassen aufteilen, z.B. System in Auskunftssystem und Buchungssystem
    - Diagramm muß dann erweitert werden
- Beispiel für verfeinertes Szenario
  - textuelle Beschreibung
    - Eine Touristin A übermittelt erst das Reiseziel und dann den Reisetermin an die Auskunft eines Informationssystems. Sie erhält darauf innerhalb von 3 Sekunden den Preis der Reise und einen Buchungscode. Da sie mit dem Preis einverstanden ist, gibt sie innerhalb von 7 Tagen unter Angabe des Codes den Buchungsauftrag an die Buchungsstelle. Nach dessen Eingang holt sich die Buchungsstelle alle nötigen Informationen von der Auskunft und erteilt ihr danach eine Bestätigung.

Erweiterte Sequenzdiagramme (4)

- Horizontal verfeinertes Sequenzdiagramm
  - neue Instanz / Klasse
  - zusätzliche vertikale Verfeinerung durch Parameter der Botschaften

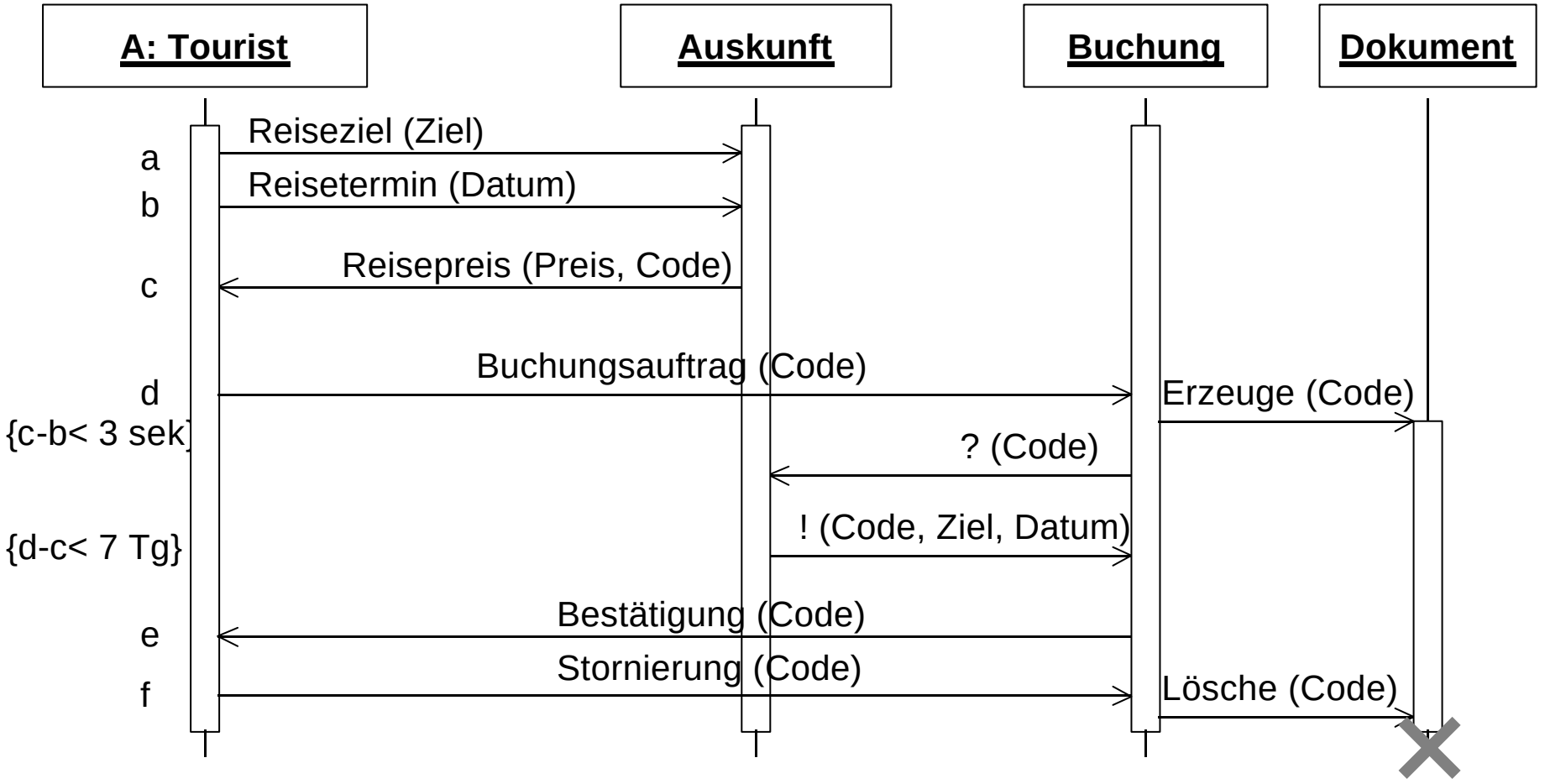


Erweiterte Sequenzdiagramme (5)

- Szenario mit Erzeugung und Löschung von Objekten
  - textuelle Beschreibung
    - Eine Touristin A übermittelt erst das Reiseziel und dann den Reisetermin an die Auskunft eines Informationssystems. Sie erhält darauf innerhalb von 3 Sekunden den Preis der Reise und einen Buchungscode. Da sie mit dem Preis einverstanden ist, gibt sie innerhalb von 7 Tagen unter Angabe des Codes den Buchungsauftrag an die Buchungsstelle. Nach dessen Eingang *legt die Buchungsstelle unter diesem Buchungscode ein Dokument an* und holt sich über den Code alle nötigen Informationen von der Auskunft. Danach erhält die Touristin A von ihr eine Bestätigung. *Nach dem Erhalt der Bestätigung storniert A die Reise unter Angabe des Codes bei der Buchungsstelle, worauf diese das zugehörige Dokument löscht.*

Erweiterte Sequenzdiagramme (6)

- Sequenzdiagramm mit Erzeugen und Löschen von Objekten
  - neue Klasse
  - Instanzierung und Löschung des Objekts in der Ablaufsequenz

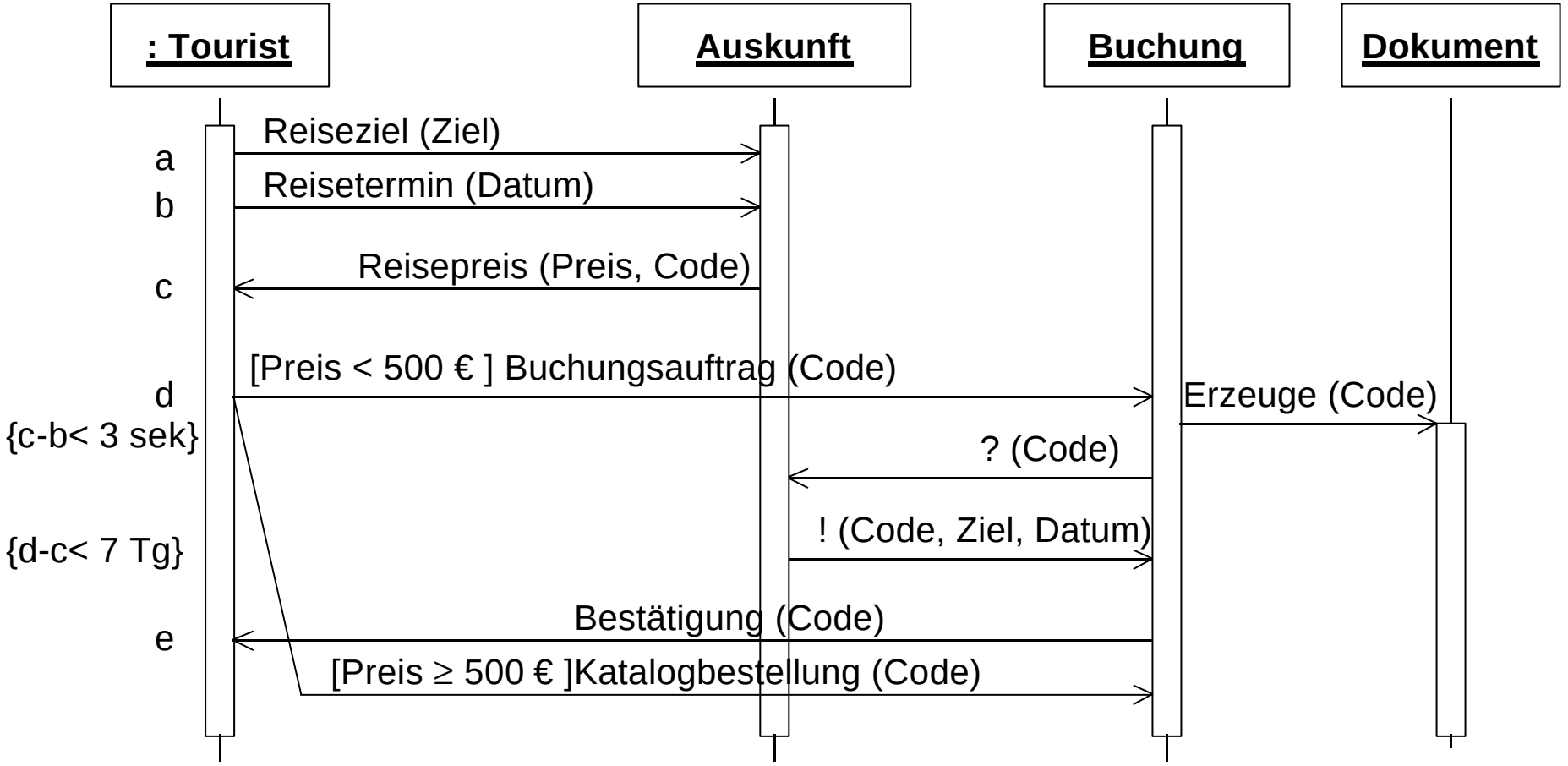


Erweiterte Sequenzdiagramme (7)

- Anwendungsfall mit Alternativen
  - textuelle Beschreibung
    - Eine Touristin übermittelt erst das Reiseziel und dann den Reisetermin an die Auskunft eines Informationssystems. Sie erhält darauf innerhalb von 3 Sekunden den Preis der Reise und einen Buchungscode. *Wenn der Preis unter 500 € liegt,* gibt sie innerhalb von 7 Tagen unter Angabe des Codes den Buchungsauftrag an die Buchungsstelle. Nach dessen Eingang legt die Buchungsstelle unter diesem Buchungscode ein Dokument an und holt sich über den Code alle nötigen Informationen von der Auskunft. Danach erhält die Touristin von ihr eine Bestätigung. *Wenn der Preis über 500 € liegt, wird keine Reise gebucht, sondern ein Katalog mit weiteren Angeboten der Region angefordert.*

Erweiterte Sequenzdiagramme (8)

- Sequenzdiagramm für Anwendungsfall
  - Alternativen mit Angabe der Bedingung



Erweiterte Sequenzdiagramme (9)

☺ Übersichtliche Beschreibung von Szenarien

- Botschaften mit Parametern
- hoher Detaillierungsgrad

☺ Charakteristika des Anwendungsfalls darstellbar

- Zeitbeschränkungen
- Rollen / Klassen
- einzelne Entscheidungen

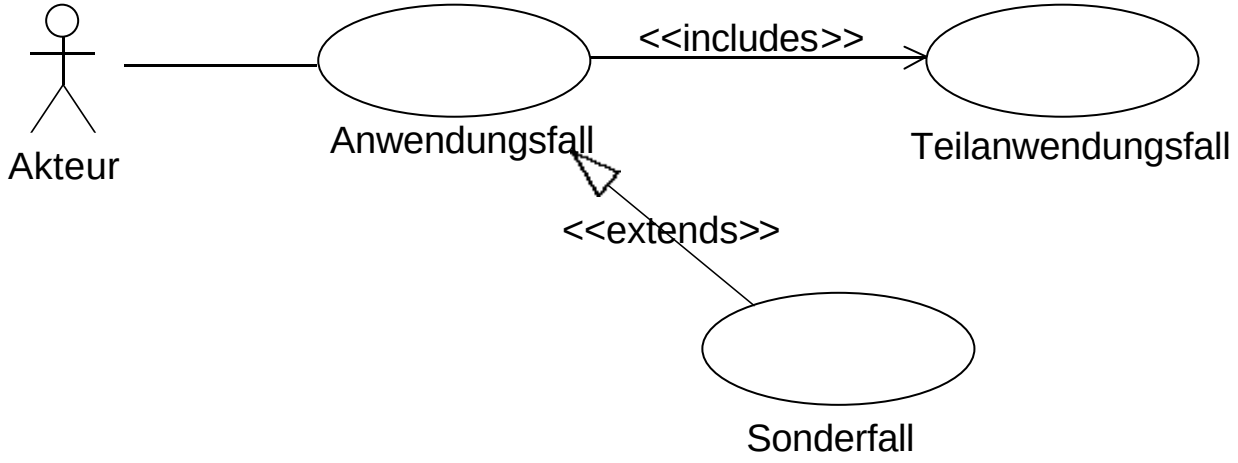
☹ Ungeeignet für komplexe Anwendungsfälle

- zu detailliert
- unübersichtlich bei vielen Entscheidungen
- unübersichtlich bei Schleifen

⇒ abstraktere Darstellung von Anwendungsfällen nötig

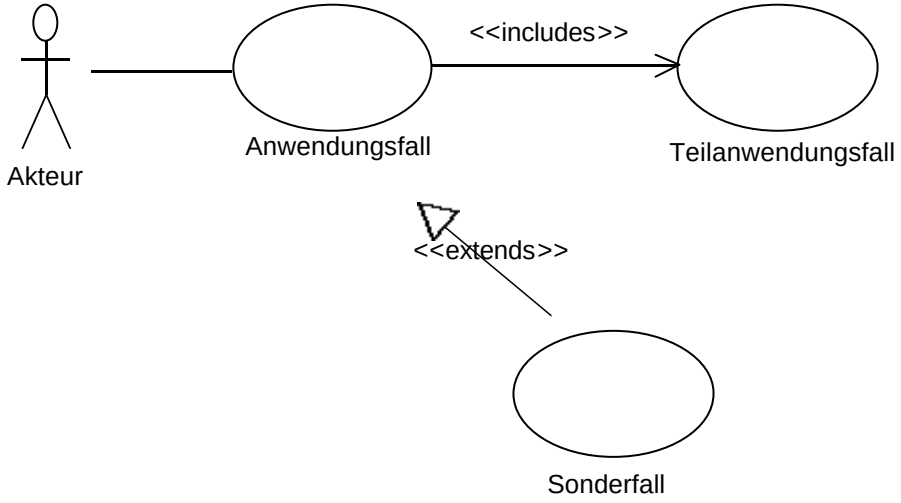
Darstellung von Anwendungsfällen

- Symbole
  - Akteur als Figur
  - Anwendungsfall als Ellipse
  - Linien zur Verbindung
  - Pfeile für Beziehungen
- Graphische Darstellung



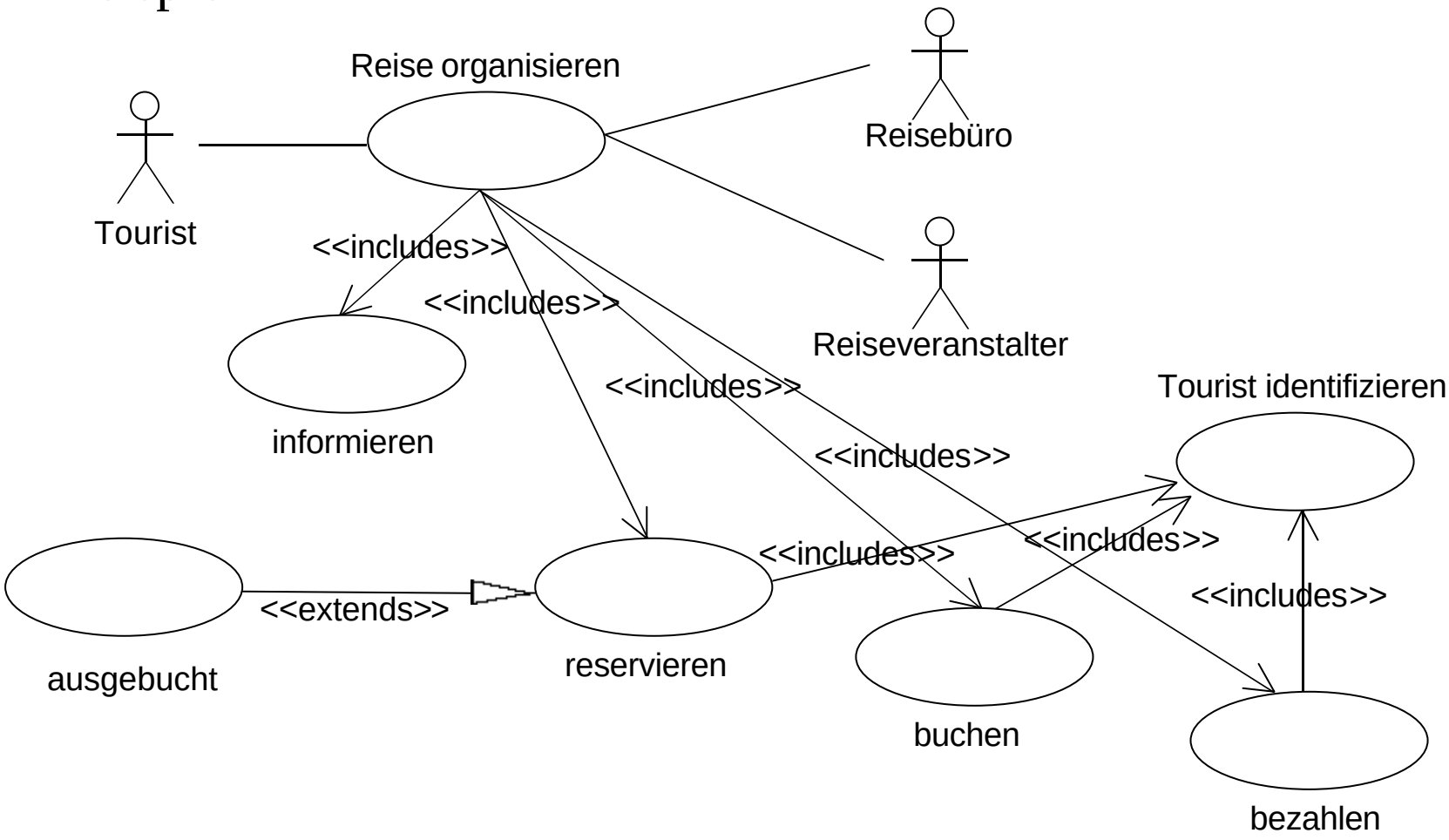
Darstellung von Anwendungsfällen (2)

- Beziehungen
  - Include-Beziehung
    - Anwendungsfall verwendet anderen
    - Teilprozesse können von mehreren Anwendungsfällen benutzt werden
  - Extend-Beziehung
    - Erweiterung eines Anwendungsfalls
    - Sonderfall bei bestimmten Bedingungen
    - Optionale Verzweigung
  - Specialize-Beziehung
    - Vererbung
    - Redefinition



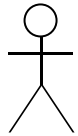
Darstellung von Anwendungsfällen (3)

- Beispiel

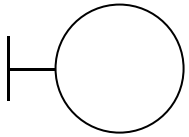


Stereotypen für Anwendungsfälle

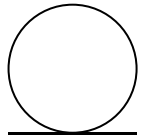
- Akteur — actor
  - wie bereits benutzt



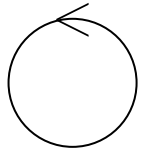
- Grenzobjekt / -klasse — boundary
  - Objekte, mit denen ein Akteur interagiert
  - je nach Detaillierungsgrad
    - einzelne Objekte der Benutzungsoberfläche
    - ganze Systeme
  - können auch Personen sein



- Entitätsobjekt / -klasse — entity
  - enthalten Informationen für den Prozeß
  - Arbeitsgegenstände
  - Werkzeuge

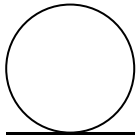


- Steuerobjekt / -klasse — control
  - enthalten Anwendungslogik
  - steuern die Übermittlung von Nachrichten
  - häufig als Methoden in Grenz- und Entitätsklassen

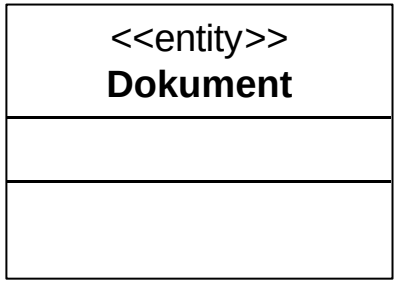


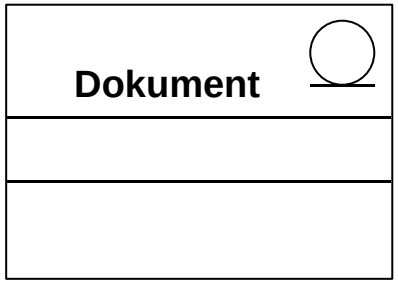
Stereotypen für Anwendungsfälle (2)

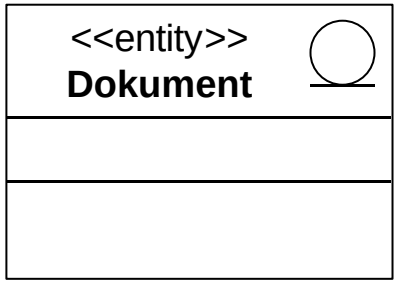
- Mehrere Darstellungsalternativen
  - z.B. für Entitätsklasse „Dokument“



Dokument

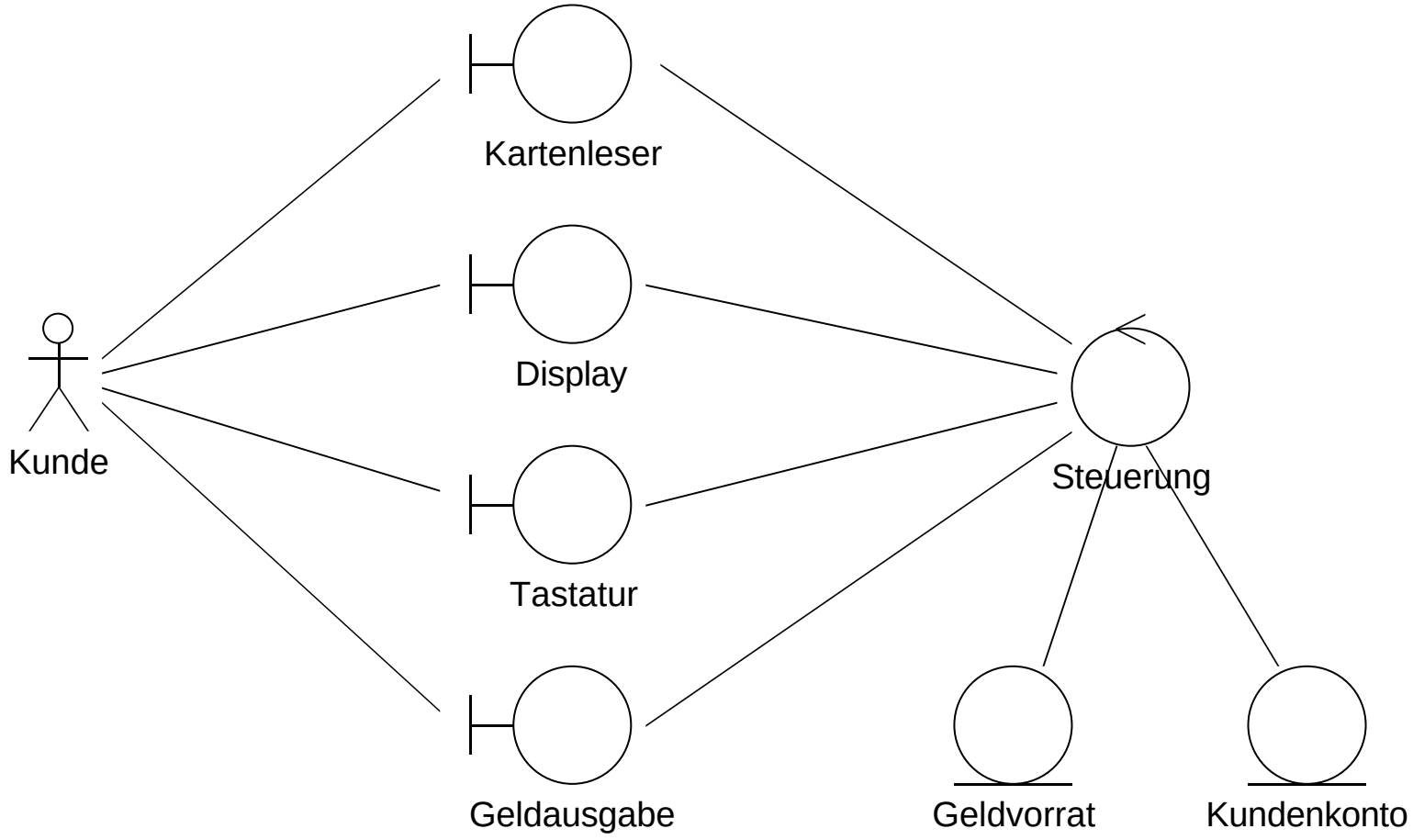






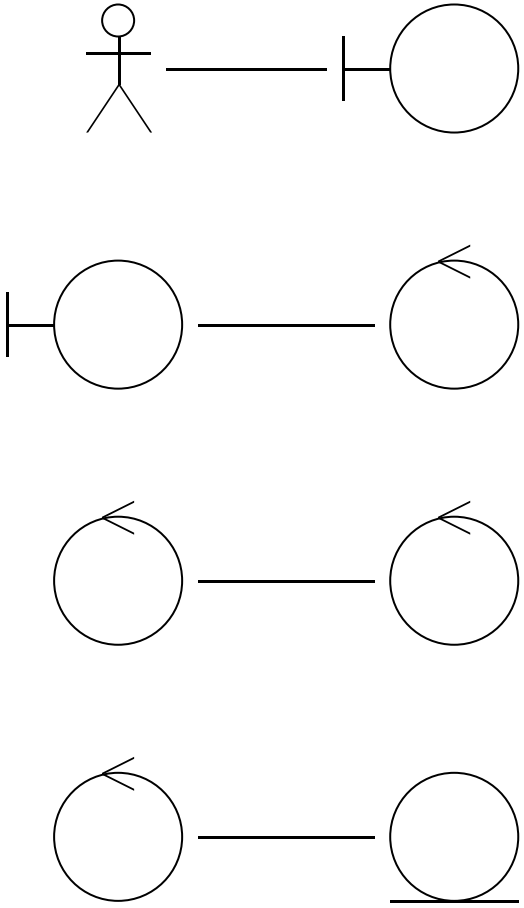
Stereotypen für Anwendungsfälle (3)

- Beispiel
  - Modellierung eines Geldautomaten

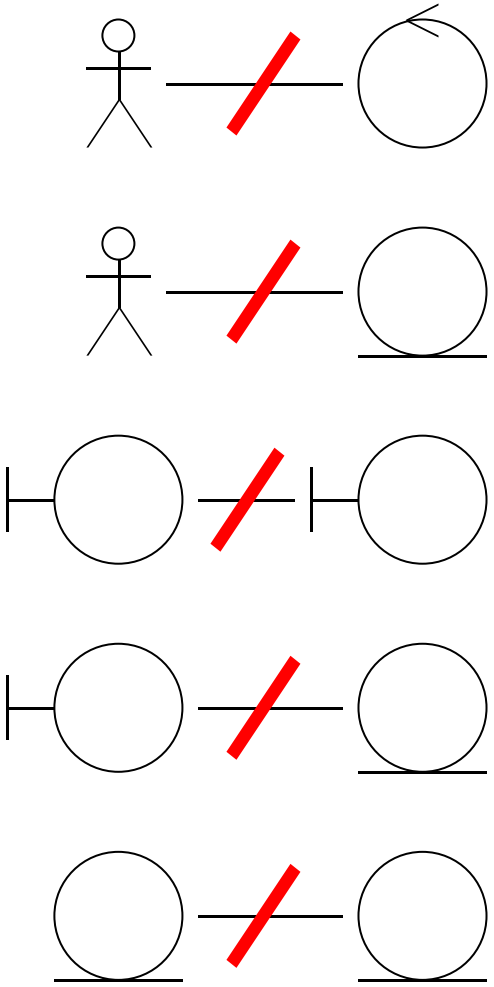


Syntaktische Regeln

- erlaubt



- verboten

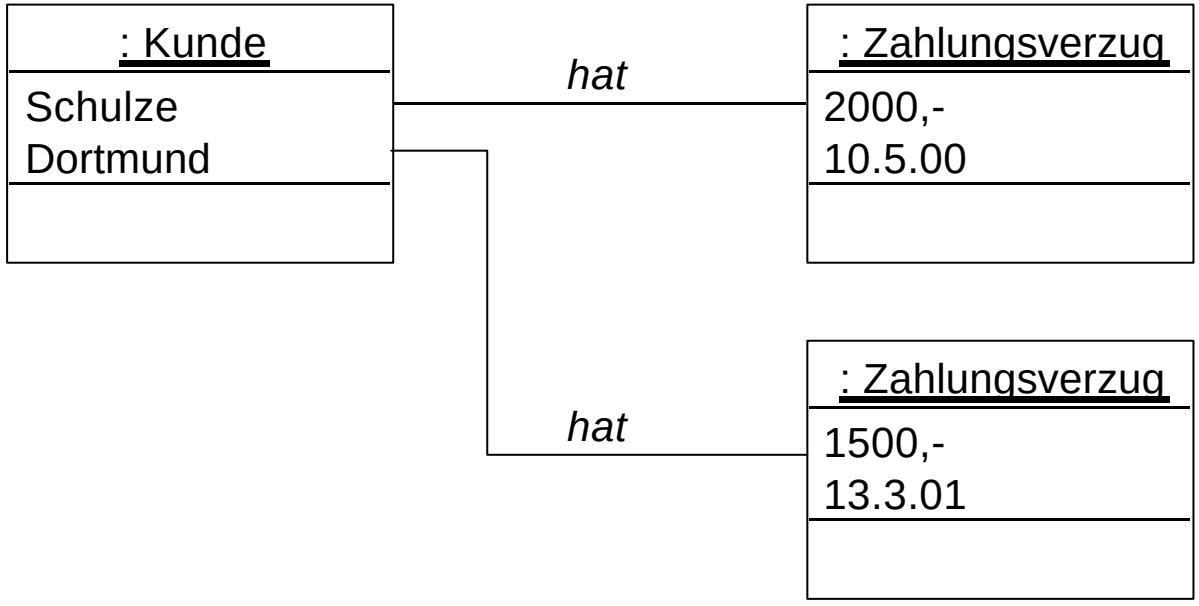


Klassendiagramme

- Assoziationen
- Zusicherungen
- Aggregation und Komposition
- Schnittstellen
- Pakete und Komponenten
- Abhängigkeiten
- Entwurfsmuster

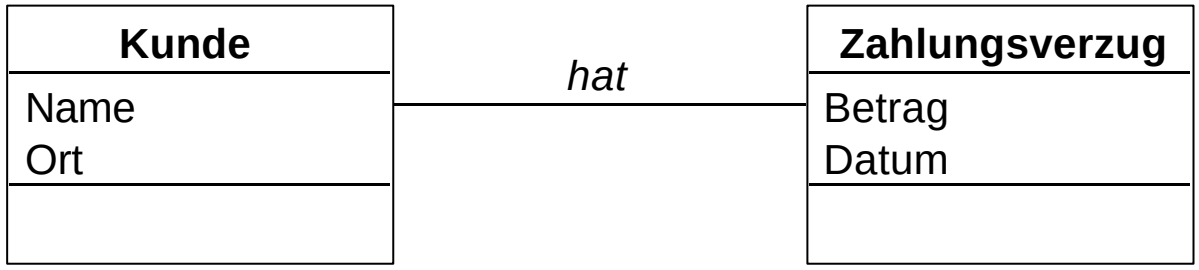
Assoziationen

- Beziehung
  - Relation zwischen Objekten
  - verknüpft Objekte gleichrangiger Klassen
  - kann auch Objekte einer einzigen Klasse verknüpfen



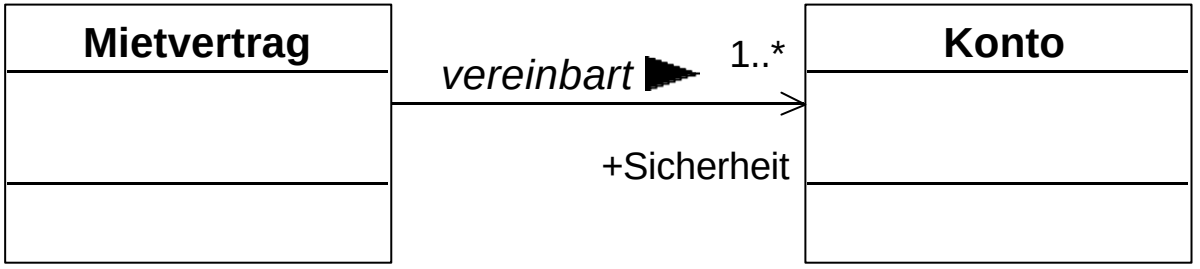
Assoziationen (2)

- Assoziation
  - ist Menge von Beziehungen zwischen Klassen
  - verknüpft Klassen
  - kann auch Klasse mit sich selbst verknüpfen



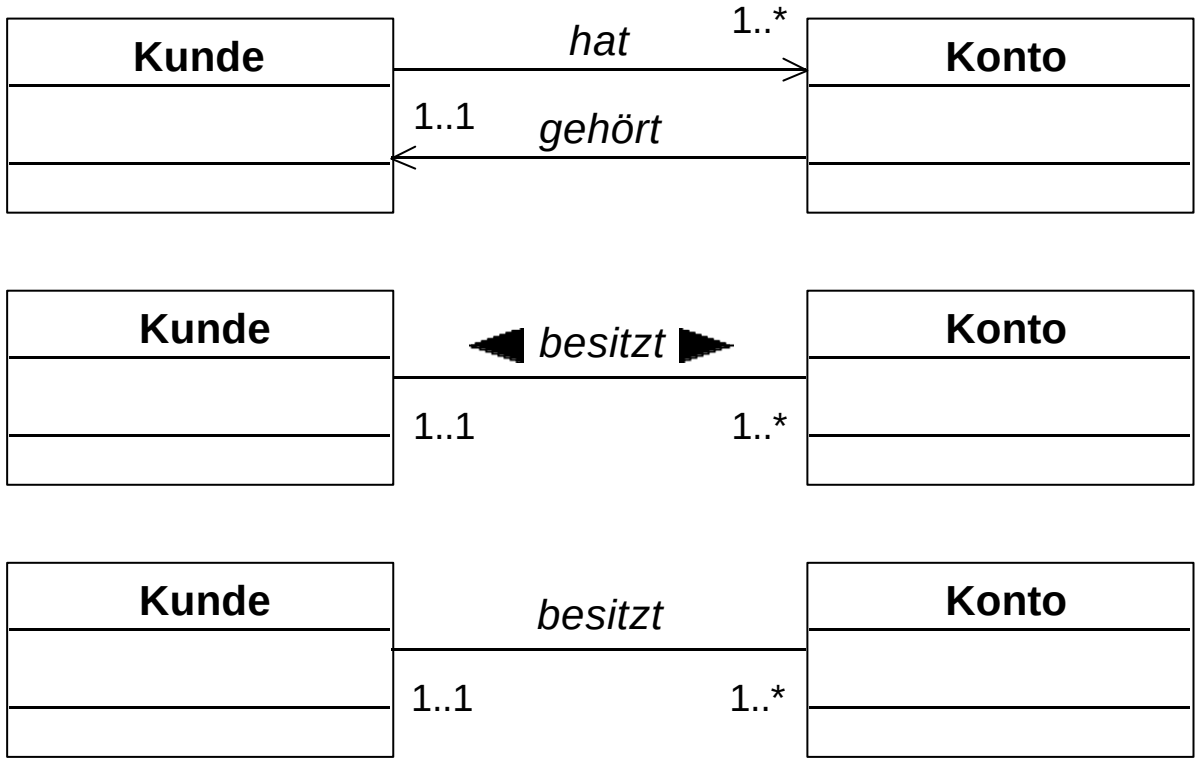
Assoziationen (3)

- Unidirektionale Assoziation
  - ist navigierbar
  - hat Kardinalität (Multiplizität)
  - kann Rolle definieren



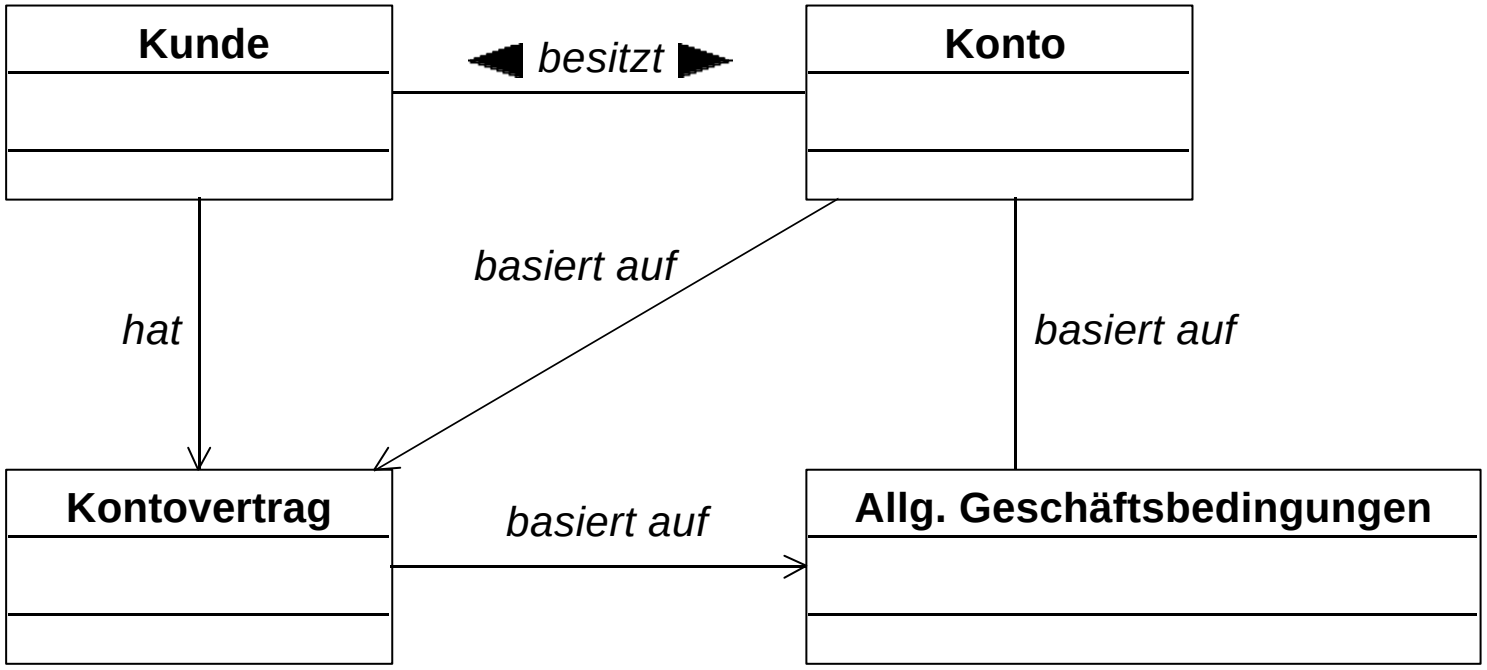
Assoziationen (4)

- Bidirektionale Assoziation
  - zwei unidirektionale Assoziationen
  - mit Angabe der Navigationsrichtungen
  - ohne Richtungsangabe



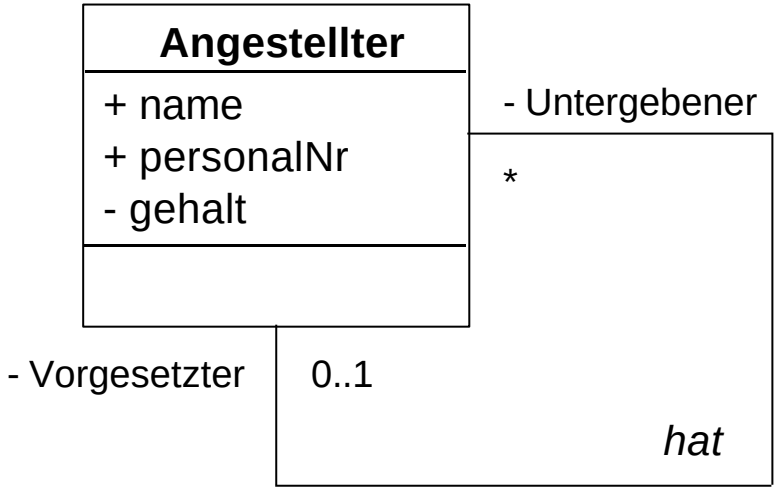
Assoziationen (5)

- Kombination von Assoziationen
  - Navigationspfade beachten



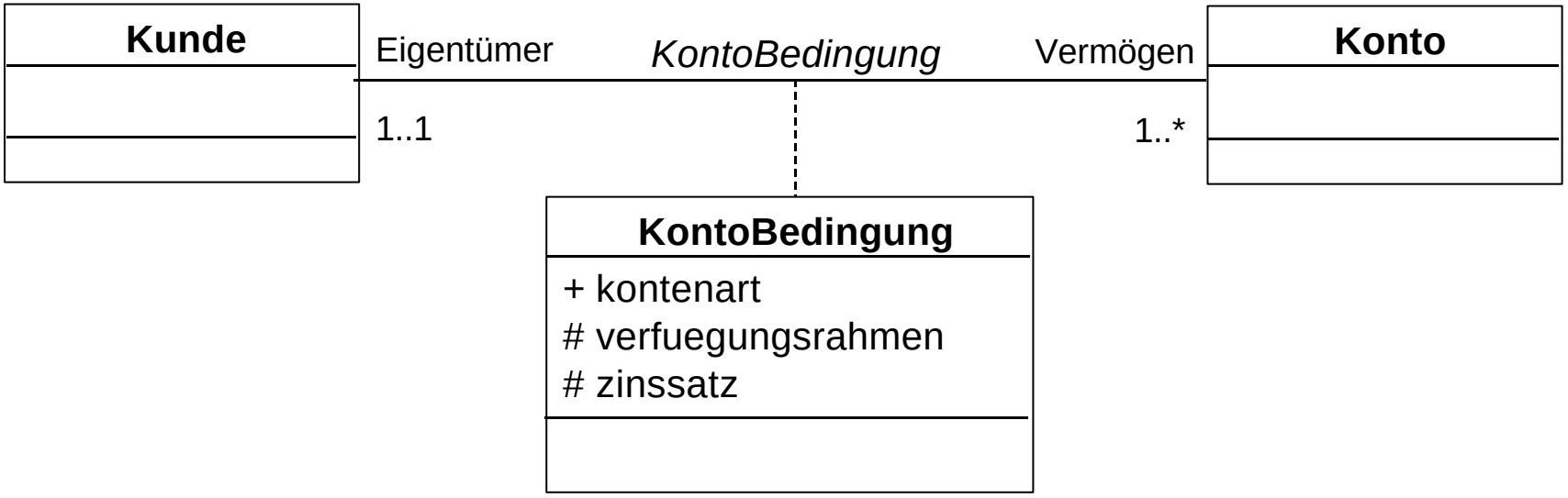
Assoziationen (6)

- Reflexive Assoziation
  - Klasse in Assoziation mit sich selbst
  - Rollen beachten



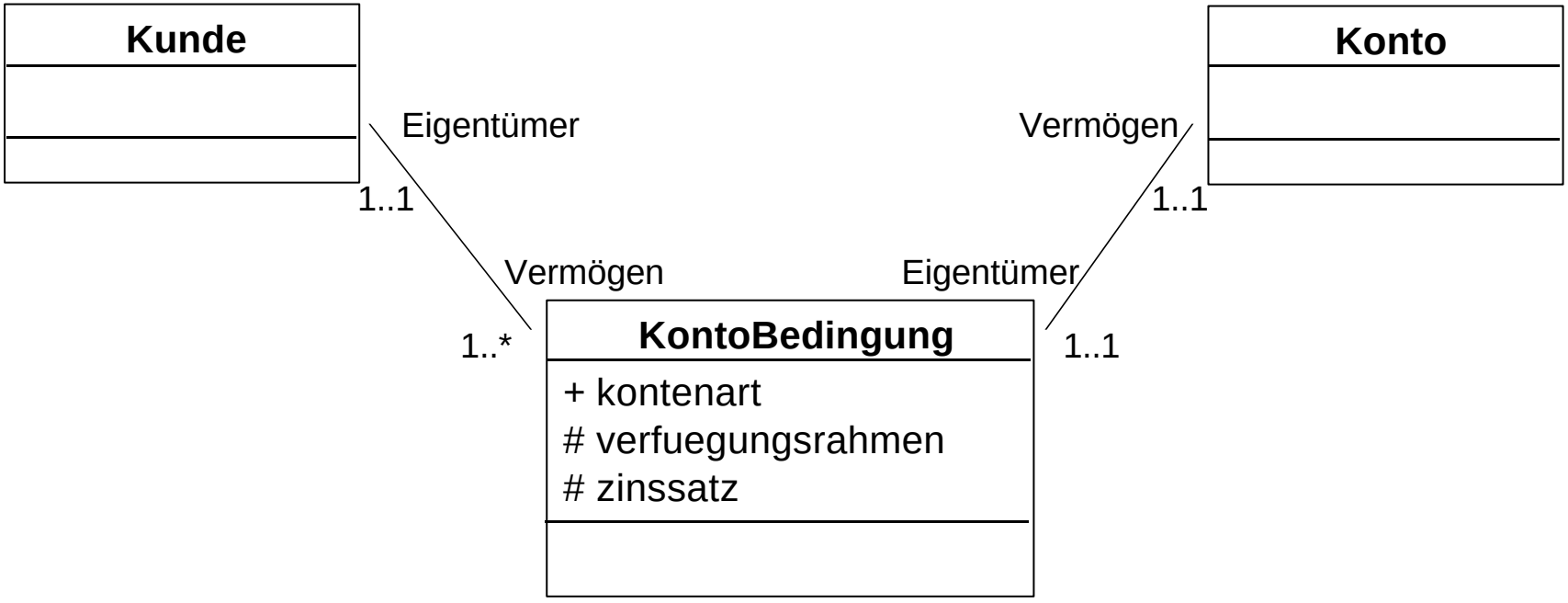
Assoziationen (7)

- Attributierte Assoziation
  - Beziehung hat Eigenschaften
  - Eigenschaftswerte sind unterschiedlich bei verschiedenen Objektbeziehungen



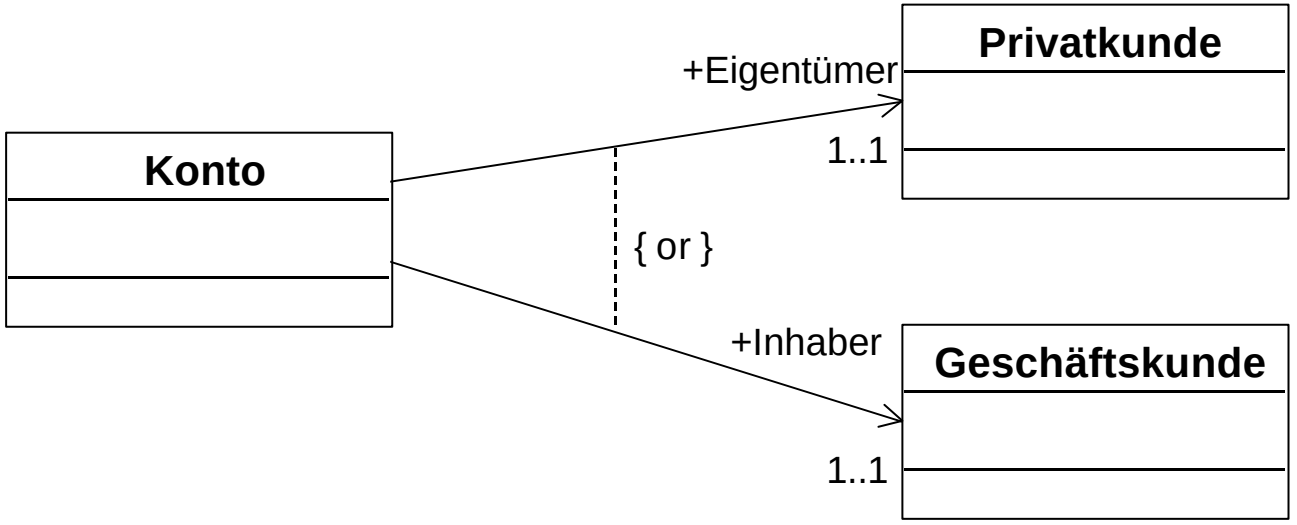
Assoziationen (8)

- Assoziationsklasse
  - kann Objekte nicht unabhängig erzeugen
  - Attribute bezeichnen die Beziehung von Objekten anderer Klassen



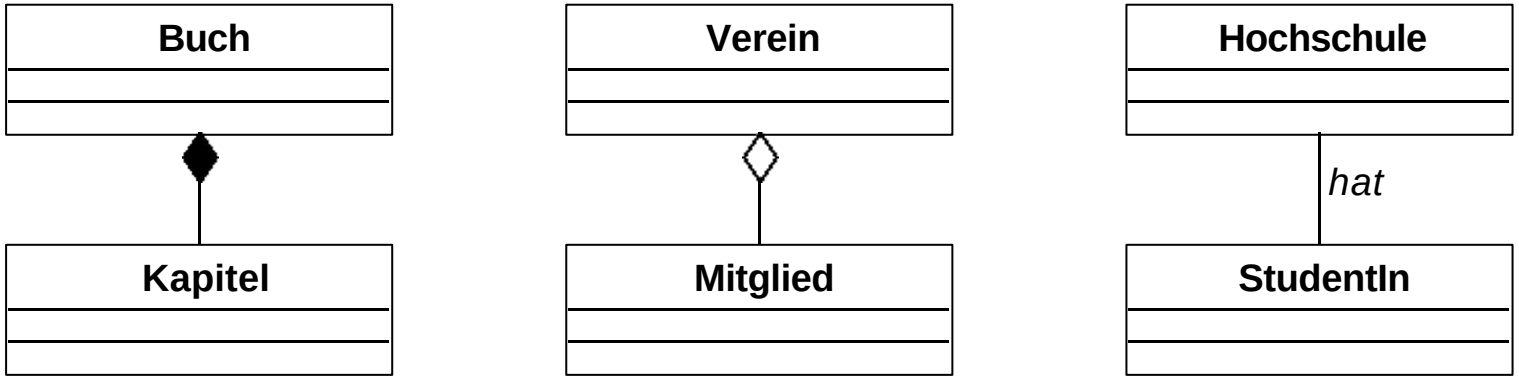
Zusicherungen

- Einschränkung für Modellelemente *constraint*
  - Inhalte, Zustände, Semantik
  - Aufzählung von Eigenschaftswerten
  - Abhängigkeitsbeziehung
    - freie Formulierung
    - logischer Ausdruck



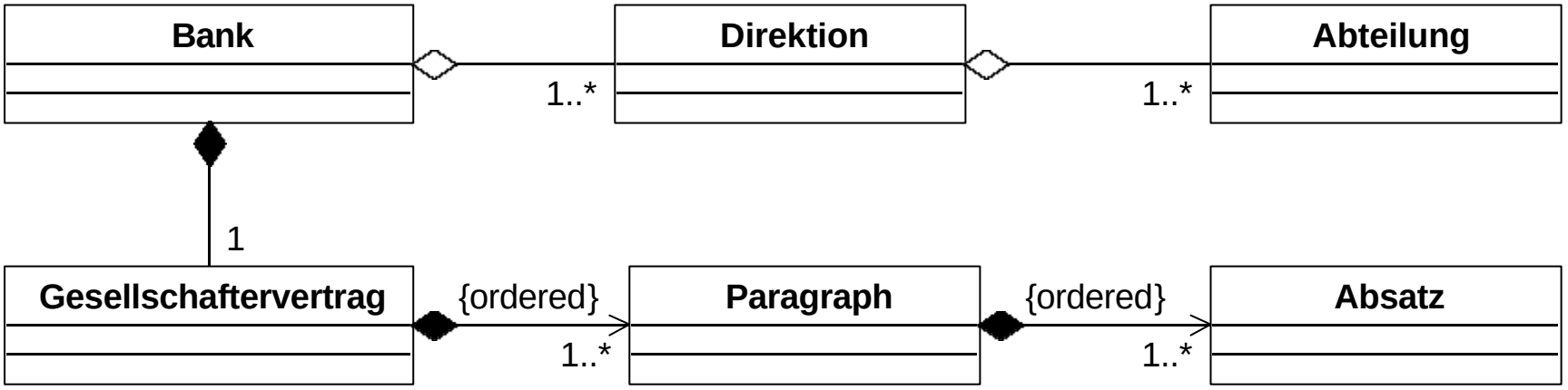
Aggregation und Komposition

- Aggregation
  ist Assoziation mit der Bedeutung „ist-Teil-von“
  - entspricht “part-of“-Assoziation bei ER-Modellen
  - zwischen realen und zwischen logischen Objekten / Klassen möglich
  - engere Beziehung als allgemeine Assoziation
- Komposition
  ist Aggregation mit sehr starker Bindung
  - Teile ohne das Ganze nicht vorstellbar
  - Das Ganze nur durch seine Teile definiert



Aggregation und Komposition (2)

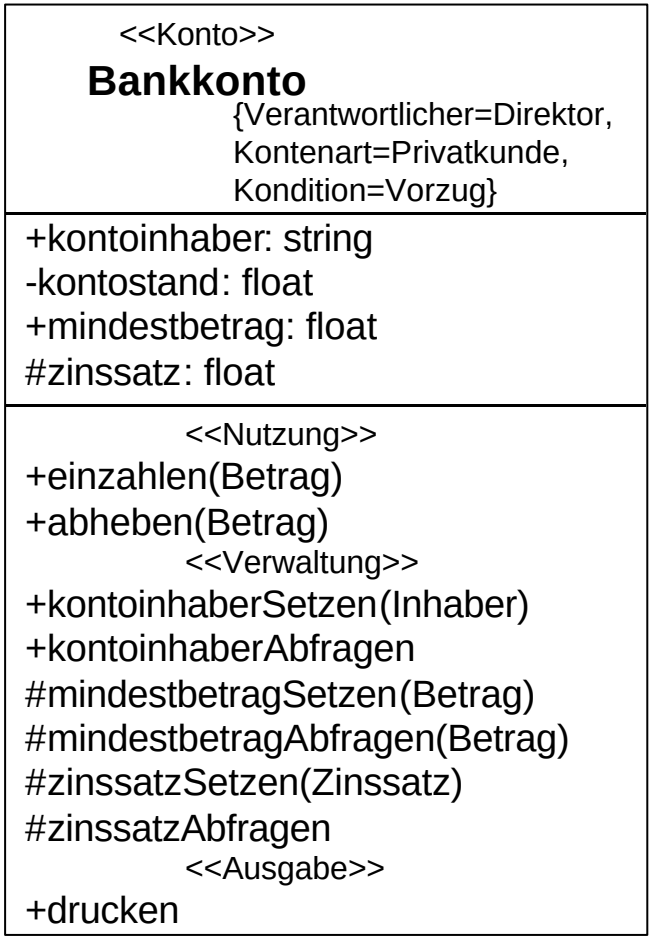
- Beispiel
  - Komposition oder Aggregation?

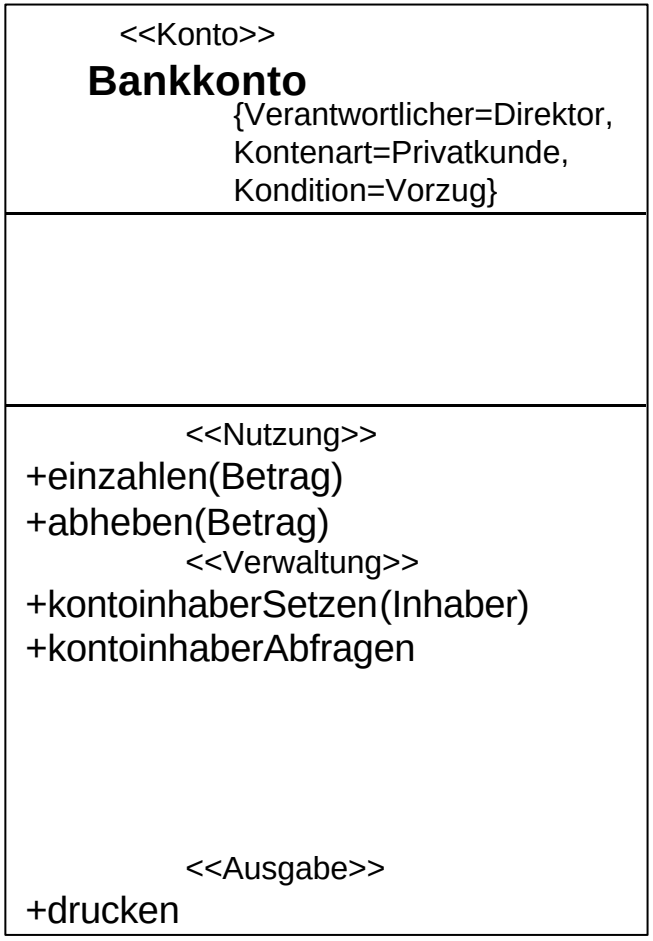


  - Organisatorische Strukturen und Mitgliedschaften meist keine Komposition

Schnittstellen

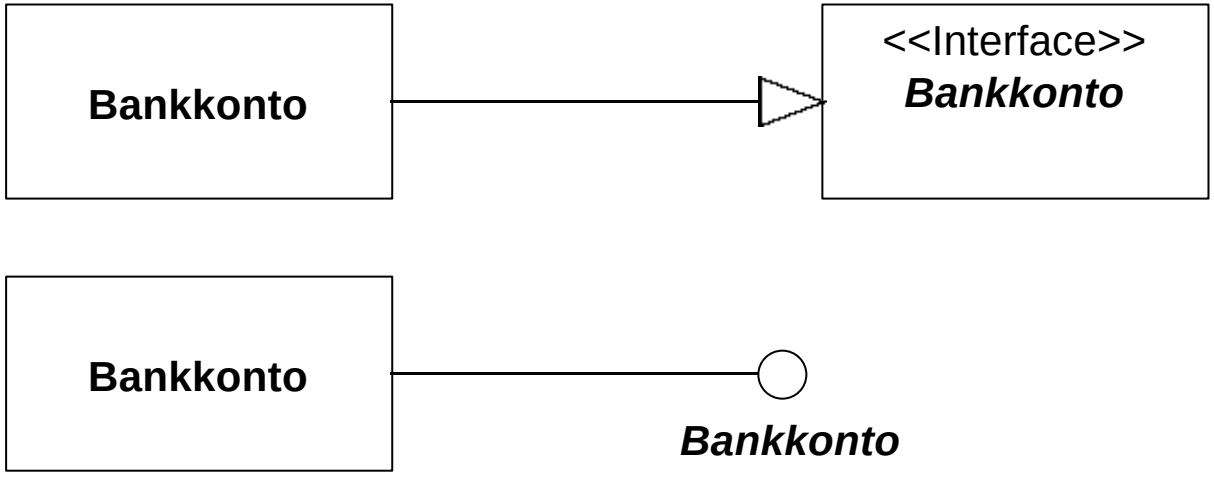
- Schnittstelle
  ist Spezifikation des nach außen sichtbaren Verhaltens
  - bei Klasse Menge der aktivierbaren Methoden



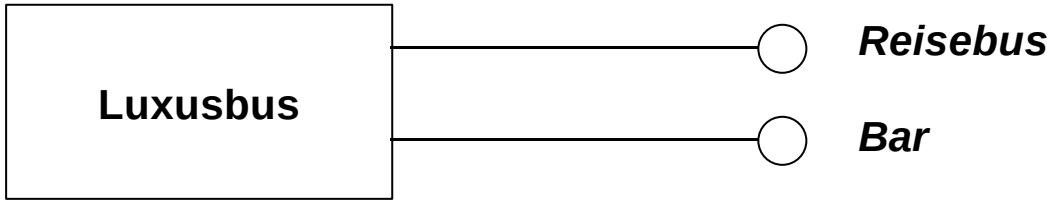


Darstellung der Klasse-Schnittstelle-Beziehung

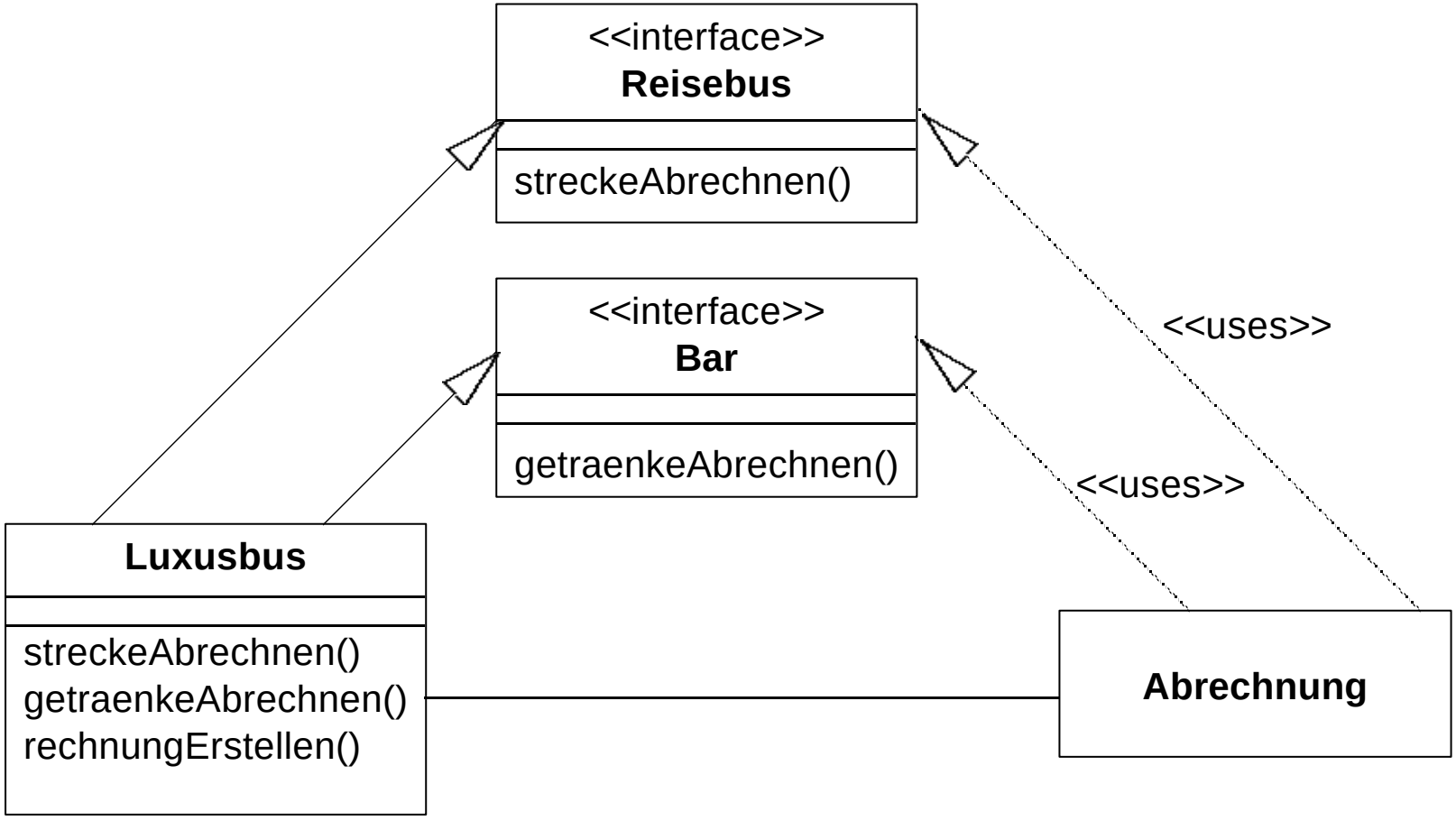
- Klasse mit einer Schnittstelle



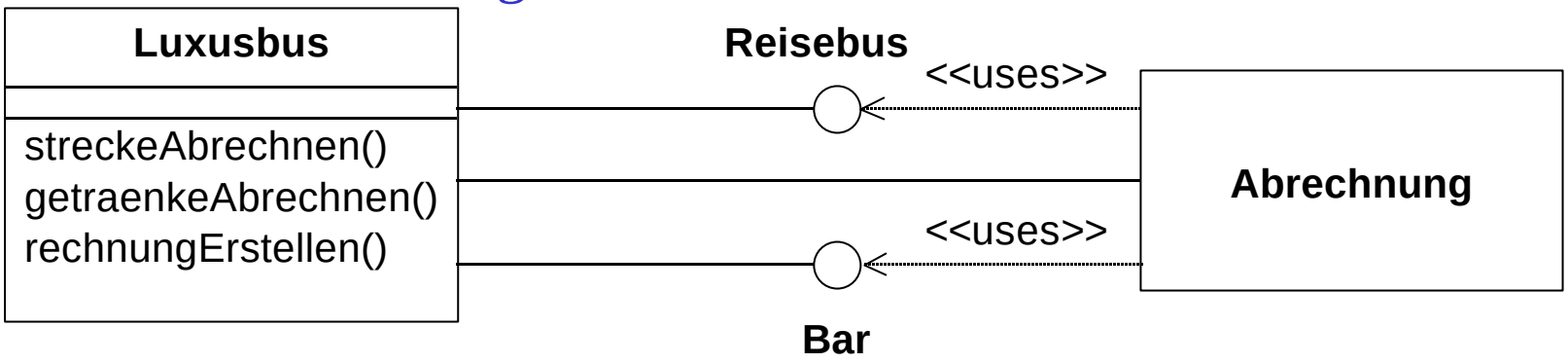
- Klasse mit zwei Schnittstellen



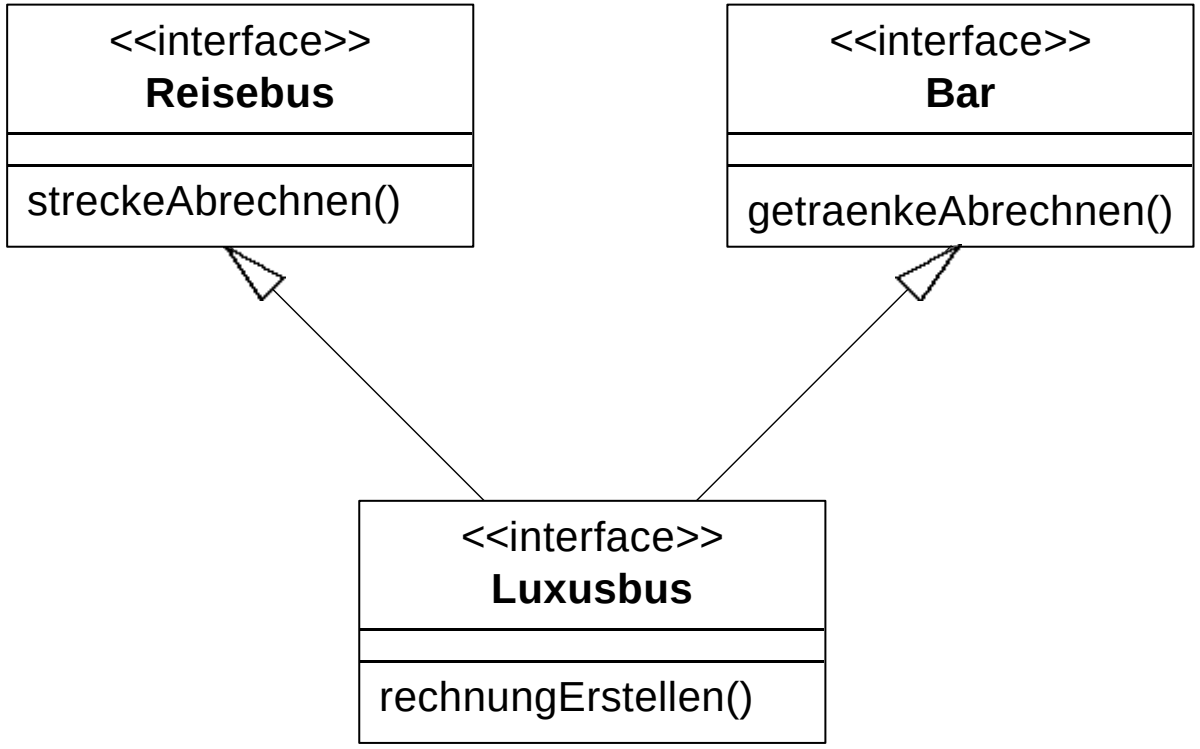
Darstellung von Schnittstellen mit Klassensymbolen



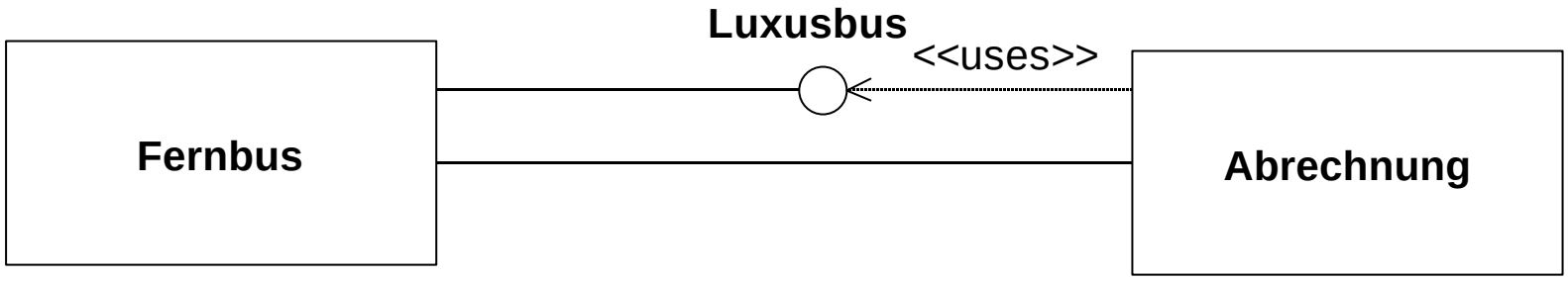
Darstellung von Schnittstellen in Kurzform



Vererbung bei Schnittstellen

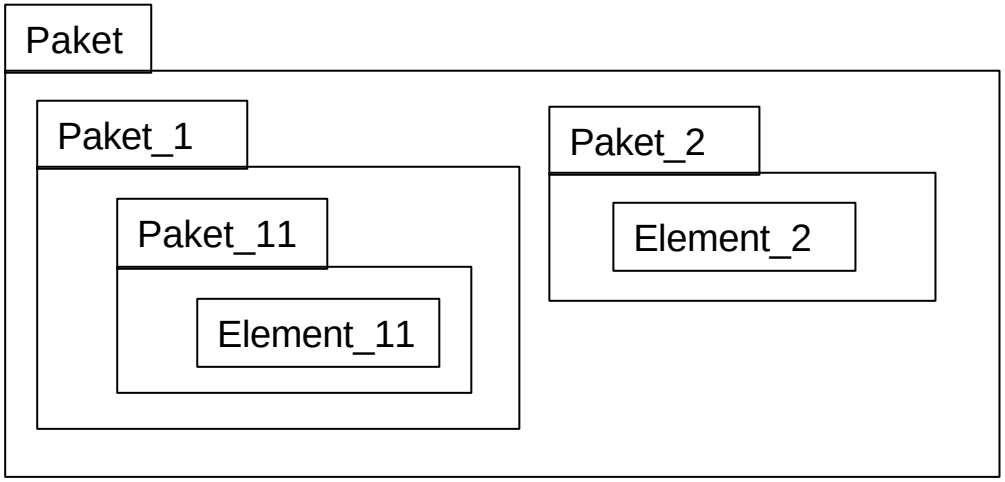


Klassenbeziehung mit Schnittstelle



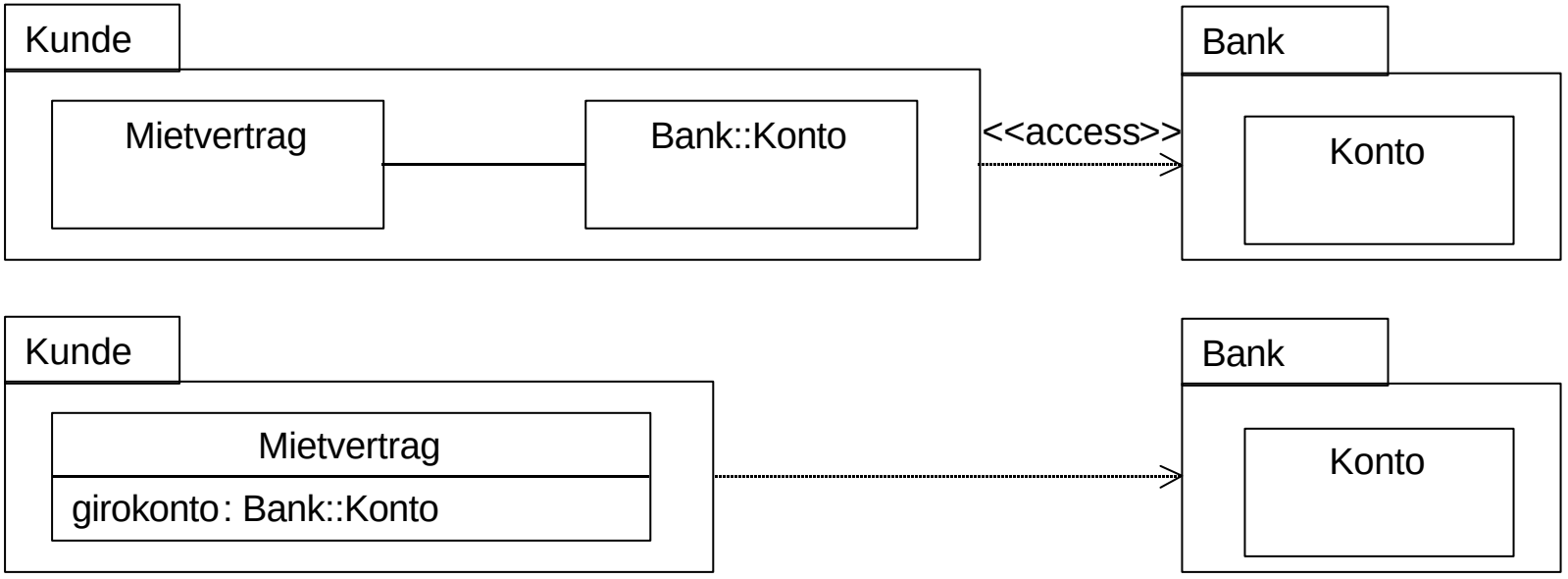
Pakete und Komponenten

- Paket
  ist eine Gruppe von Modellelemente als Einheit
  - Gliederung eines Systems in Teilsysteme
  - Gesamtsystem ist ein Paket
  - Pakete können Pakete enthalten
  - Namen müssen innerhalb eines Pakets eindeutig sein
  - Bezüge auf Elemente anderer Pakete möglich
  - insbesondere auf Klassen anwendbar bis in die Implementationsphase



Zugriff auf Elemente in anderen Paketen

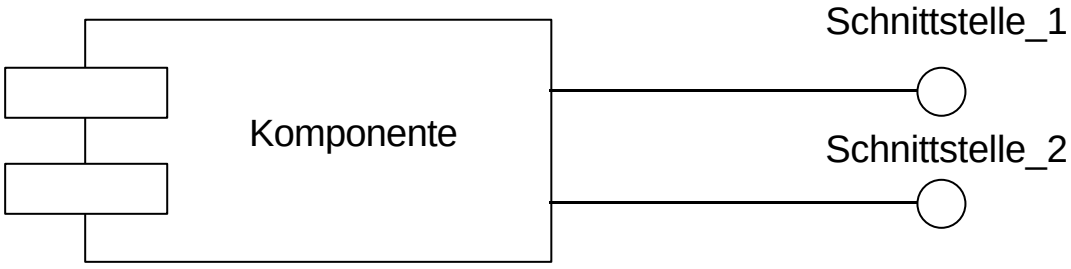
- Zugriffspfad entsprechend der Pakethierarchie
- Paketnamen getrennt durch zwei Doppelpunkte
  - z.B. Paket::Paket_1::Paket_11::Element11
- Referenzen auf Klassen in anderen Paketen
  - zur Nutzung als Klasse
  - zur Nutzung als Attribut



Komponente

Eine Komponente ist ein Software-Modul mit ausführbaren Bestandteilen und definierten Schnittstellen

- vornehmlich in der Implementationsphase
- Modularisierungskonzept



Abhängigkeiten

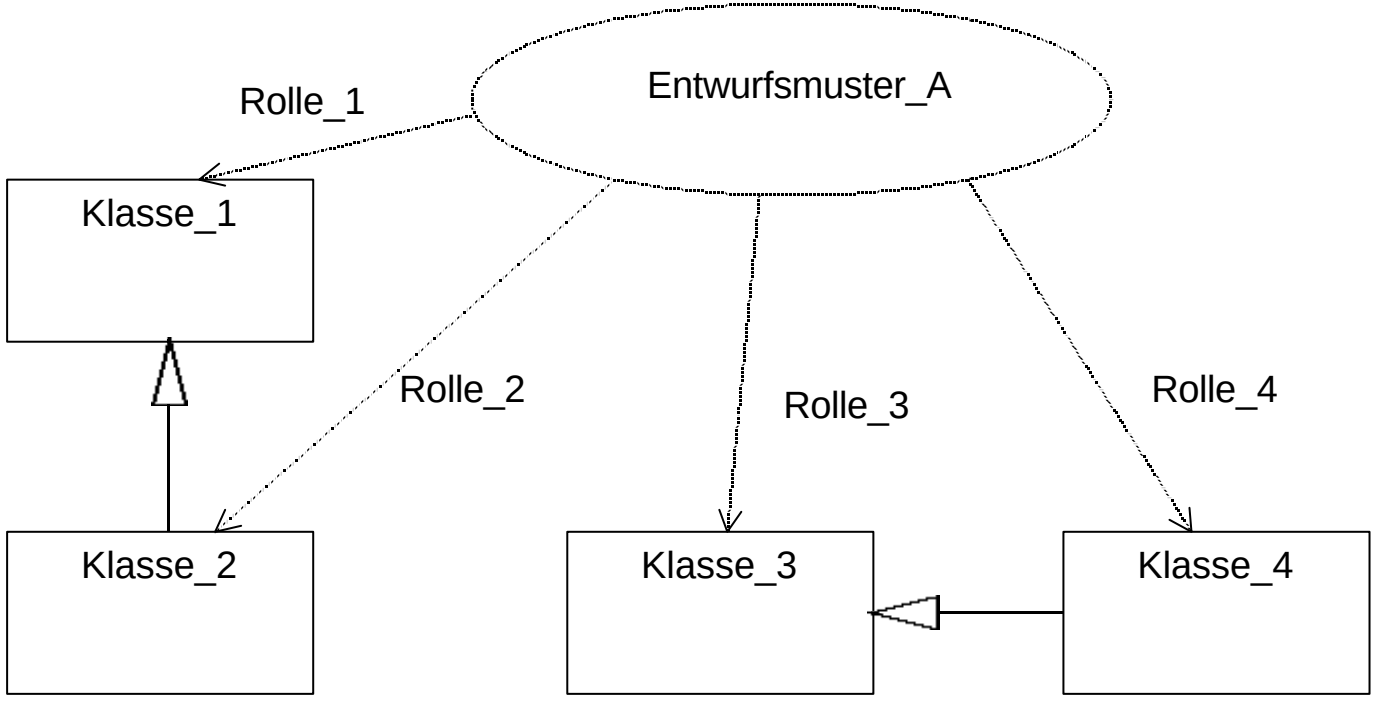
- Abhängigkeit
  ist Beziehung, die bedeutet, dass eine Änderung im ersten Element eine Veränderung im zweiten erfordern kann
  - erstes Element unabhängig
  - zweites Element abhängig
- Abhängigkeitsarten z.B.
  - <<derive>> <<derivation>> Ableitung
  - <<refine>> <<refinement>> Verfeinerung
  - <<use>> <<usage>> Benutzung
- Darstellung
  - gestrichelter Pfeil
  - zwischen verschiedenen Modellelementen möglich, z.B.
    - Methode abhängig von Methode / abhängig von Klasse
    - Klasse abhängig von Klasse / Schnittstelle
    - Paket abhängig von Paket

Entwurfsmuster, Design Patterns

- Entwurfsmuster (Design Patterns)
  sind generalisierte Lösungsideen zu häufig auftretenden Entwurfsproblemen
  - keine fertig codierten Lösungen
  - Beschreibungen des Lösungsansatzes
- Bestandteile eines Entwurfsmusters
  - Name
  - Kontext- und Problemspezifikation
  - Lösung
    - Elemente, Beziehungen, Interaktionen
    - allgemeiner als Algorithmus
  - Folgerungen
  - Beispiele

Entwurfsmuster (2)

- Allgemeine Darstellung
  - gestrichelte Ellipse mit Namen
  - gestrichelte Pfeile zu den Elementen mit Rollen



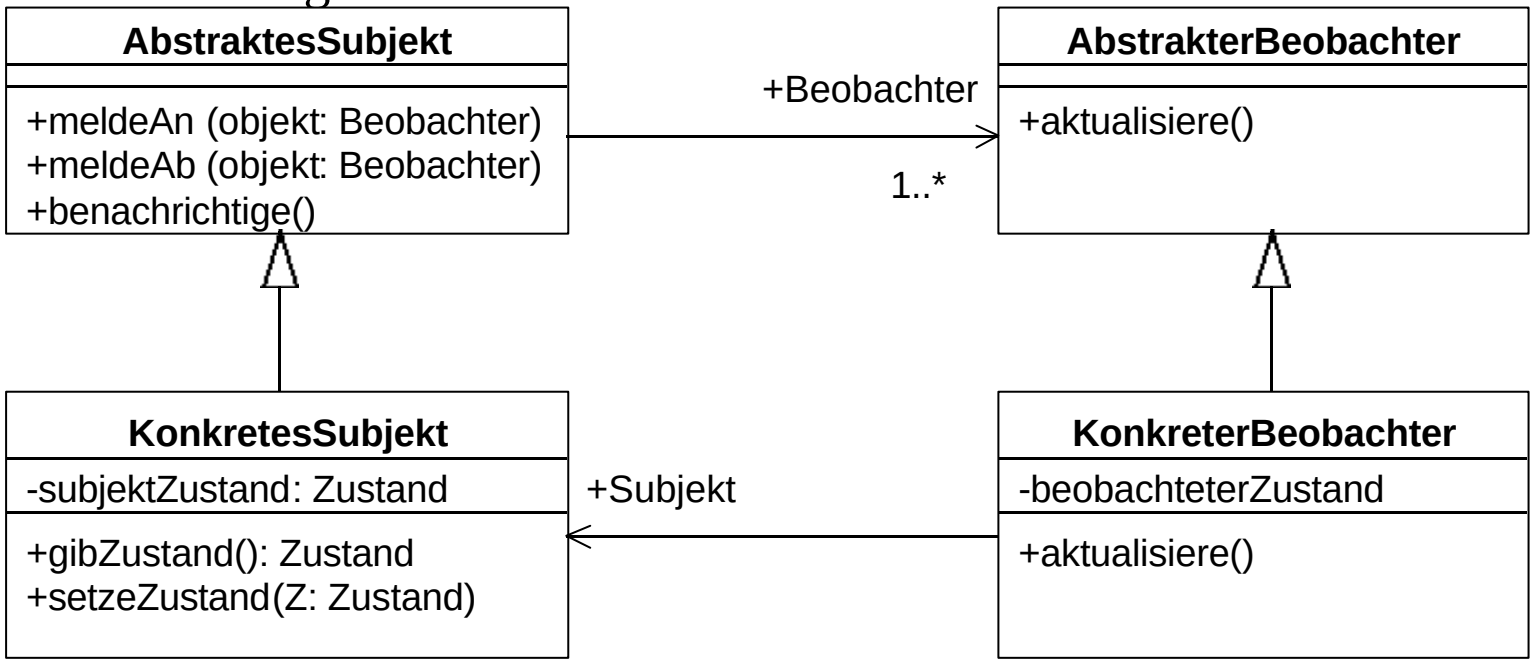
Entwurfsmuster (2)

- Beispiel: Uhr
  - ein Uhrwerk (im Rechner)
  - verschiedene Anzeigen mit verschiedenen Darstellungen
  - jede Anzeige muß zu Beginn beim Uhrwerk angemeldet werden
  - Uhrwerk informiert alle Anzeigen bei Veränderungen
  - Anzeigen können Zustand der Uhrwerks abfragen

  - Anzeigen sind Beobachter des Uhrwerks
  - Uhrwerk ist beobachtetes Objekt
  - Beobachter haben Methode zur Zustandsabfrage
  - beobachtetes Objekt hat Methode zur Anmeldung neuer Beobachter
  - beobachtetes Objekt hat Methode zur Information aller Beobachter

Entwurfsmuster (3)

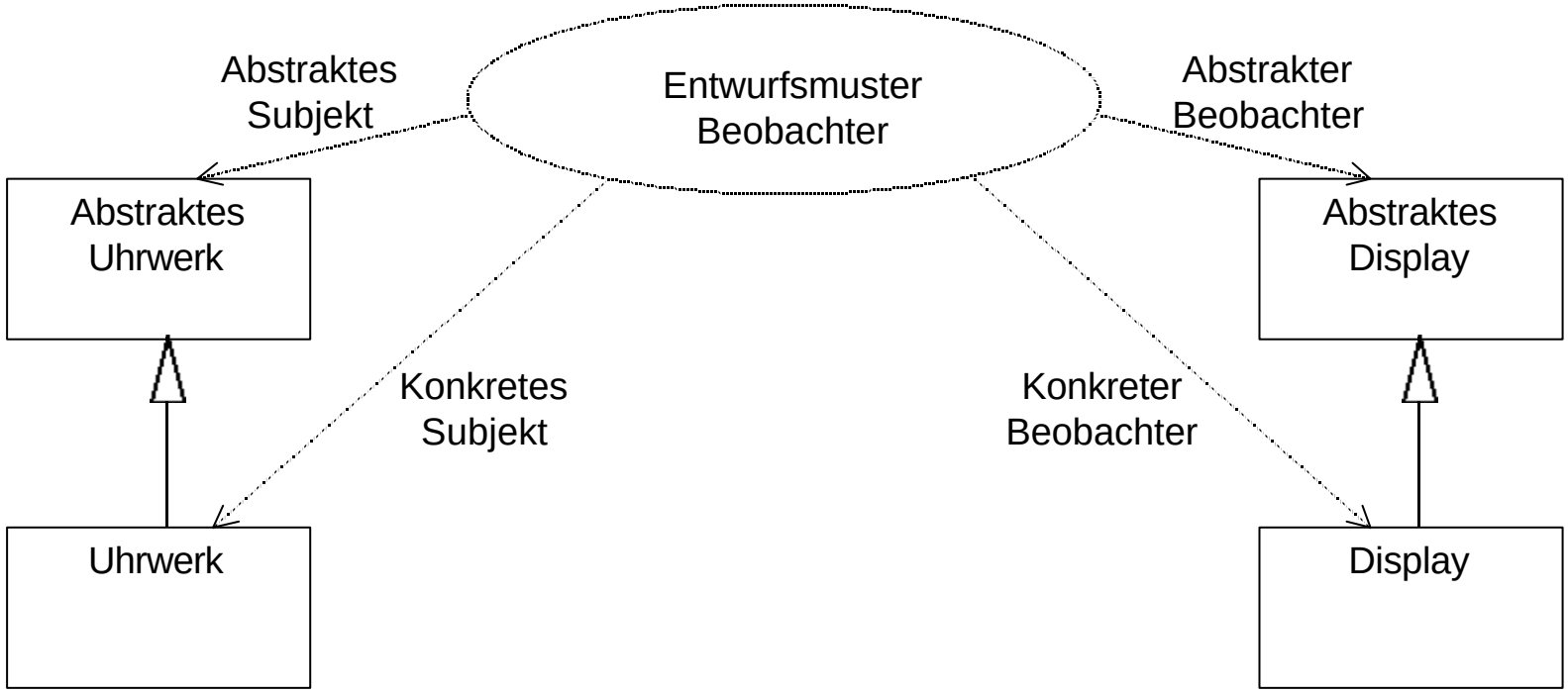
- Verallgemeinerung: Entwurfsmuster Beobachter
  - abstrakte Klassen für zu beobachtende Subjekte und für Beobachter
  - Assoziation zwischen diesen abstrakten Klassen
  - Vererbung zu konkreten Klassen



  - benachrichtige() Für alle b in Beobachter {b.aktualisiere()}
  - aktualisiere() beobachteterZustand = Subjekt.gibZustand()

Entwurfsmuster (4)

- Anwendung des Entwurfsmusters Beobachter für Uhr
  - Klassendiagramm mit Entwurfsmuster
  - dient zur Überprüfung und Kommentierung der Rollen und Methoden



Zustandsdiagramme

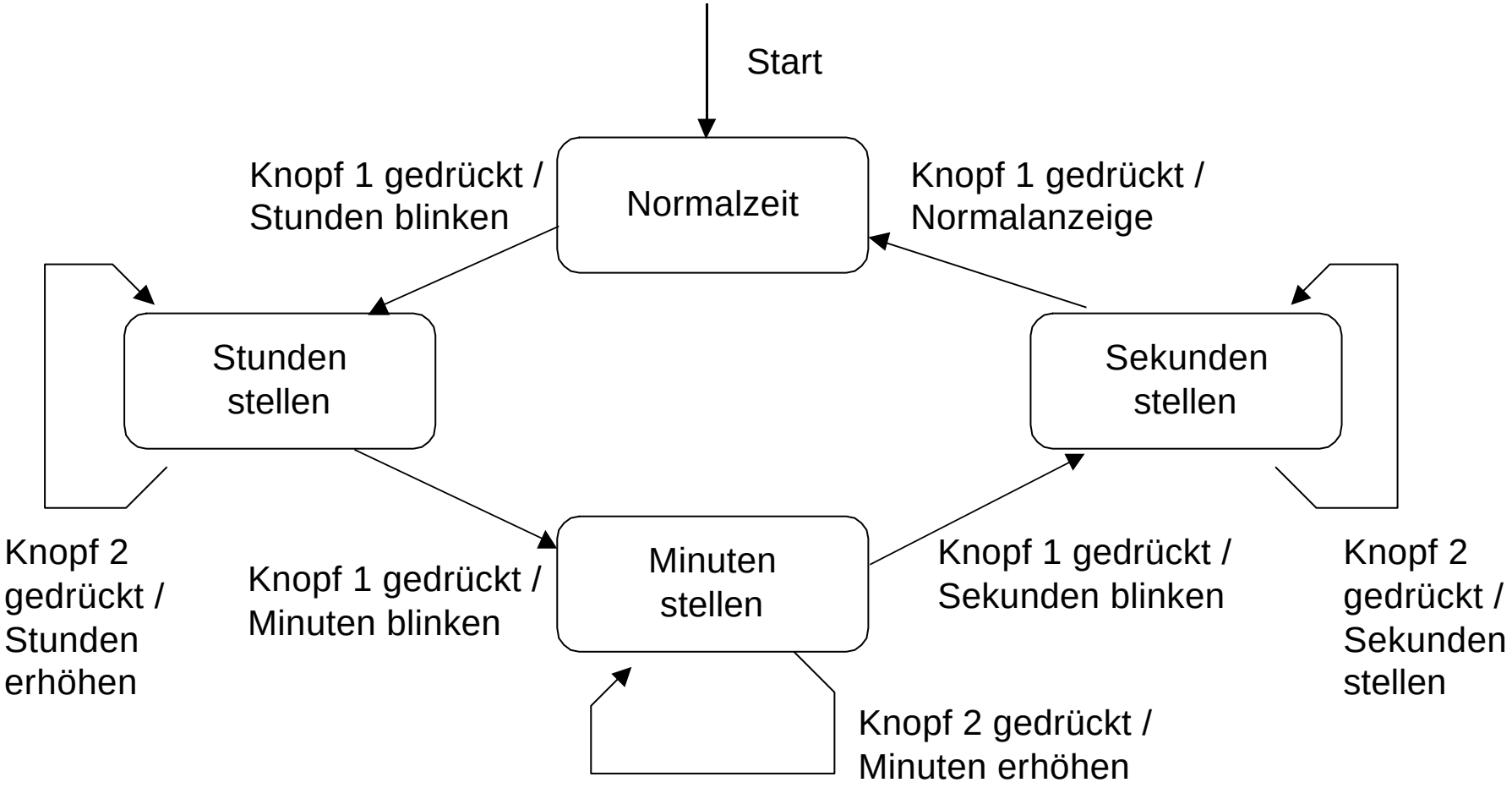
- Zustandsautomat
- Mealy-Automat und Moore-Automat
- Harel-Automat
- Modellierung von Objektzuständen

Zustandsautomat

- Merkmale
  - endliche Anzahl von Zuständen
  - verarbeitet Eingaben / Ereignisse
  - erzeugt Ausgaben / Aktionen
  - Zustände beinhalten Information über bisherigen Ablauf
  - Zustandswechsel durch Eingaben / Ereignisse
- Graphische Darstellung von Zuständen und Übergängen
  - Zustände als abgerundete Rechtecke
  - Endzustände mit doppeltem Rand
  - Startzustand markiert durch Pfeil „Start“
  - Übergänge als Pfeile
  - Ereignisse / Eingaben an den Pfeilen
  - Aktionen an den Pfeilen nach Schrägstrich
  - Bedingungen an den Pfeilen in eckigen Klammern

Zustandsautomat (2)

- Beispiel Digitaluhr als Graphik



Zustandsautomat (3)

- Zustandstabelle
  - Tabelle mit 4 Spalten:
    - Aktueller Zustand
    - Ereignis (Eingabe)
    - Aktion (Ausgabe)
    - Folgezustand
- Beispiel Digitaluhr als Tabelle

Zustand	Ereignis	Aktion	Folgezustand
Normalzeit	Knopf 1	Stunden blinken	Stunden stellen
Stunden stellen	Knopf 1	Minuten blinken	Minuten stellen
Stunden stellen	Knopf 2	Stunden erhöhen	Stunden stellen
Minuten stellen	Knopf 1	Sekunden blinken	Sekunden stellen
Minuten stellen	Knopf 2	Minuten erhöhen	Minuten stellen
Sekunden stellen	Knopf 1	Normalanzeige	Normalzeit
Sekunden stellen	Knopf 2	Sekunden stellen	Sekunden stellen

Zustandsautomat (4)

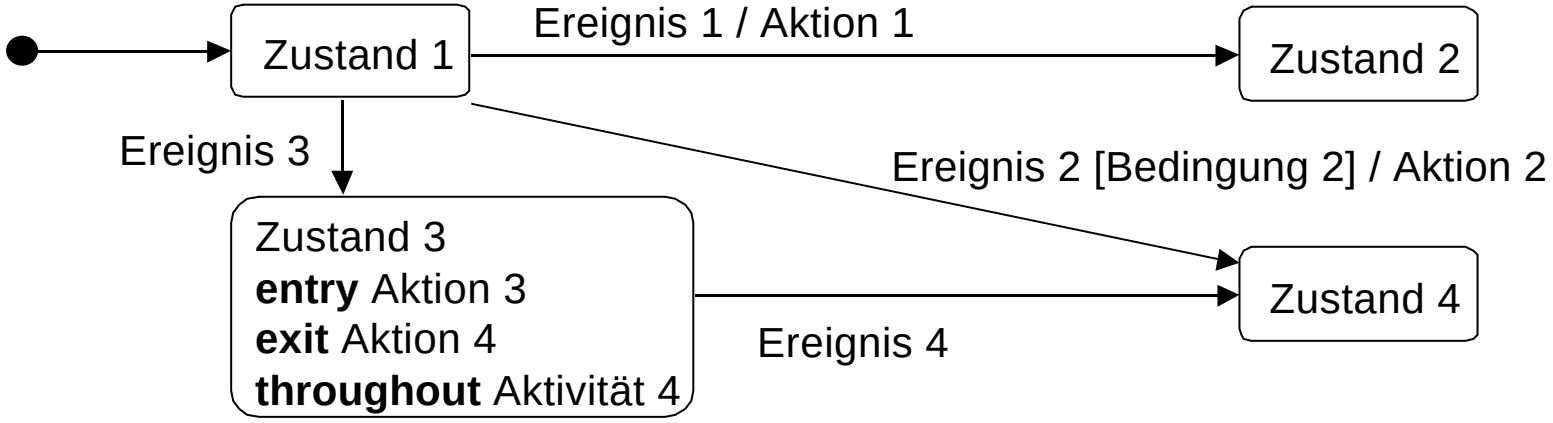
- Matrixdarstellungen
  - Zustandsmatrix
    - Spalten für Zustände
    - Zeilen für Ereignisse (Eingaben)
    - in den Feldern Kombination Aktion / Folgezustand
  - Zustandsmatrix (alternative Darstellung)
    - Ausgangszustände in Zeilen
    - Folgezustände in Spalten
    - in den Feldern Kombination Ereignis / Aktion
  - beide Matrix-Darstellungen erlauben Vollständigkeitsprüfung
  - beide Matrix-Darstellungen oft schwach besetzt

Mealy-Automat und Moore-Automat

- Verschiedene Automatendarstellungen
  - Mealy-Automat
    - Ausgabe beim Zustandsübergang
    - Zustand steht für Ruheperiode
  - Moore-Automat
    - Ausgaben / Aktionen sind an die Zustände gebunden
  - Moore- und Mealy-Automat können ineinander umgewandelt werden
  - Moore- und Mealy-Automat sind deterministisch
    (zu jedem Zustand und jeder Eingabe höchstens ein Übergang)
  - Moore- und Mealy-Automat lassen sich mathematisch spezifizieren

Harel-Automat

- Hybrider Zustandsautomat
  - kombiniert Moore- und Mealy-Automat
  - Aktionen möglich
    - beim Übergang zwischen Zuständen
    - beim Eintreten in einen Zustand
    - beim Verlassen eines Zustands
  - Aktivitäten möglich in Zuständen
  - Zustandsübergänge können von Bedingungen abhängen

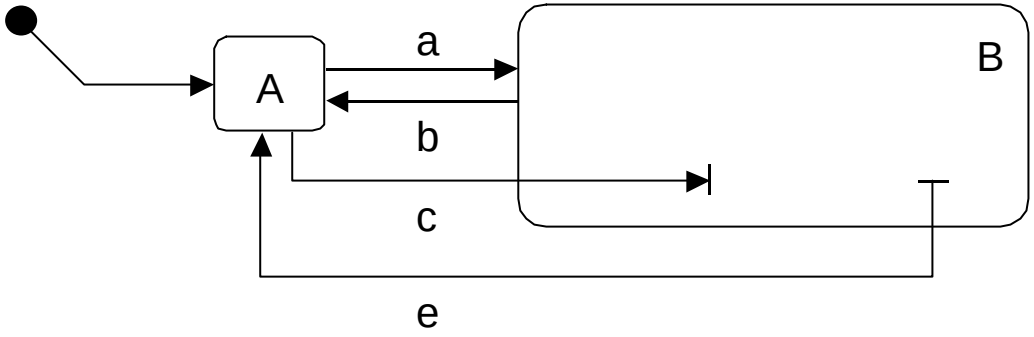


Harel-Automat (2)

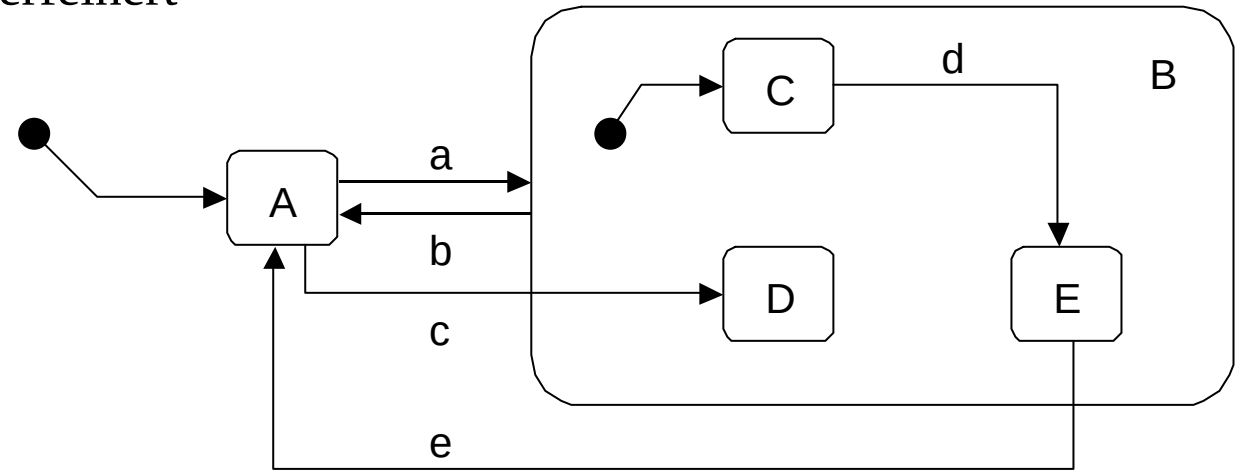
- Hierarchischer Zustandsautomat
  - geschachtelte Zustände zur Zusammenfassung (bottom-up) oder Verfeinerung (top-down)
  - Übergänge möglich
    - zwischen Zuständen gleicher Hierarchiestufe
    - in einen Unterzustand
    - aus einem Unterzustand
    - in einen Oberzustand (d.h. dessen Anfangszustand)
    - aus einem Oberzustand (d.h. aus allen seinen Unterzuständen)

Harel-Automat (3)

- Graphische Darstellung hierarchischer Zustände
  - abstrahiert von Unterzuständen

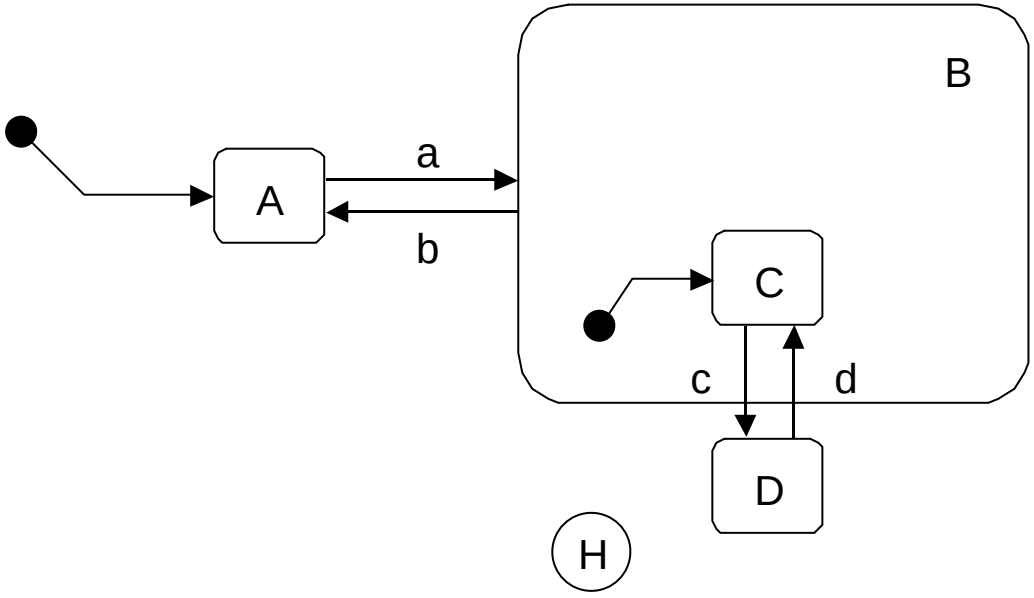


  - verfeinert



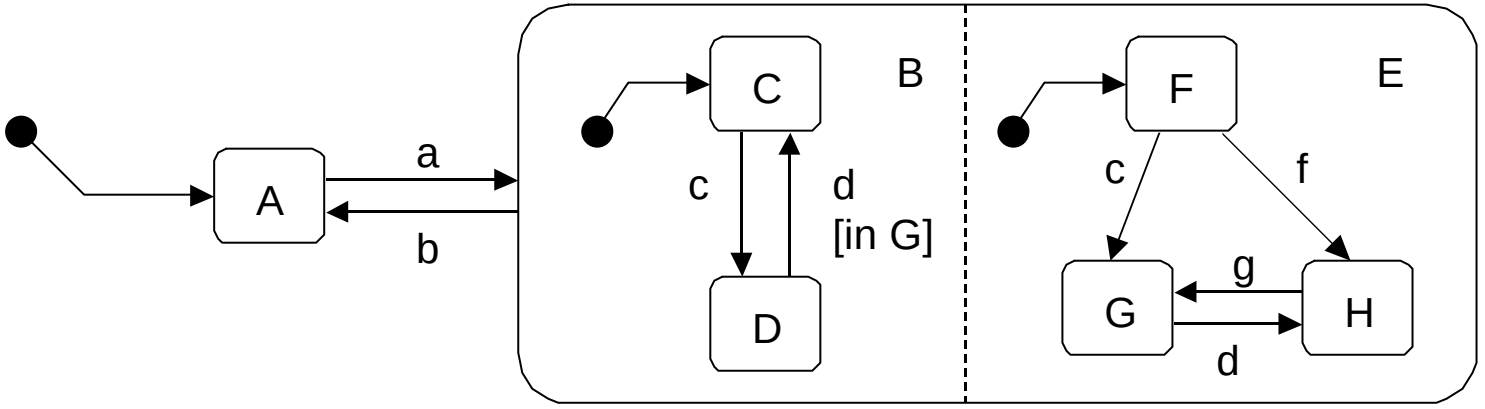
Harel-Automat (4)

- Zustände mit Gedächtnis
  - Zustand mit Unterzuständen
    merkt sich beim Verlassen den jeweiligen Unterzustand
  - beim nächsten Betreten wird nicht der Anfangsunterzustand,
    sondern der zuletzt besuchte aufgesucht



Harel-Automat (5)

- Nebenläufige Zustände
  - Oberzustand mit getrennten Komponenten
  - Automat befindet sich gleichzeitig in zwei Unterzuständen
  - Übergänge können Zustand anderer Komponente als Bedingung haben

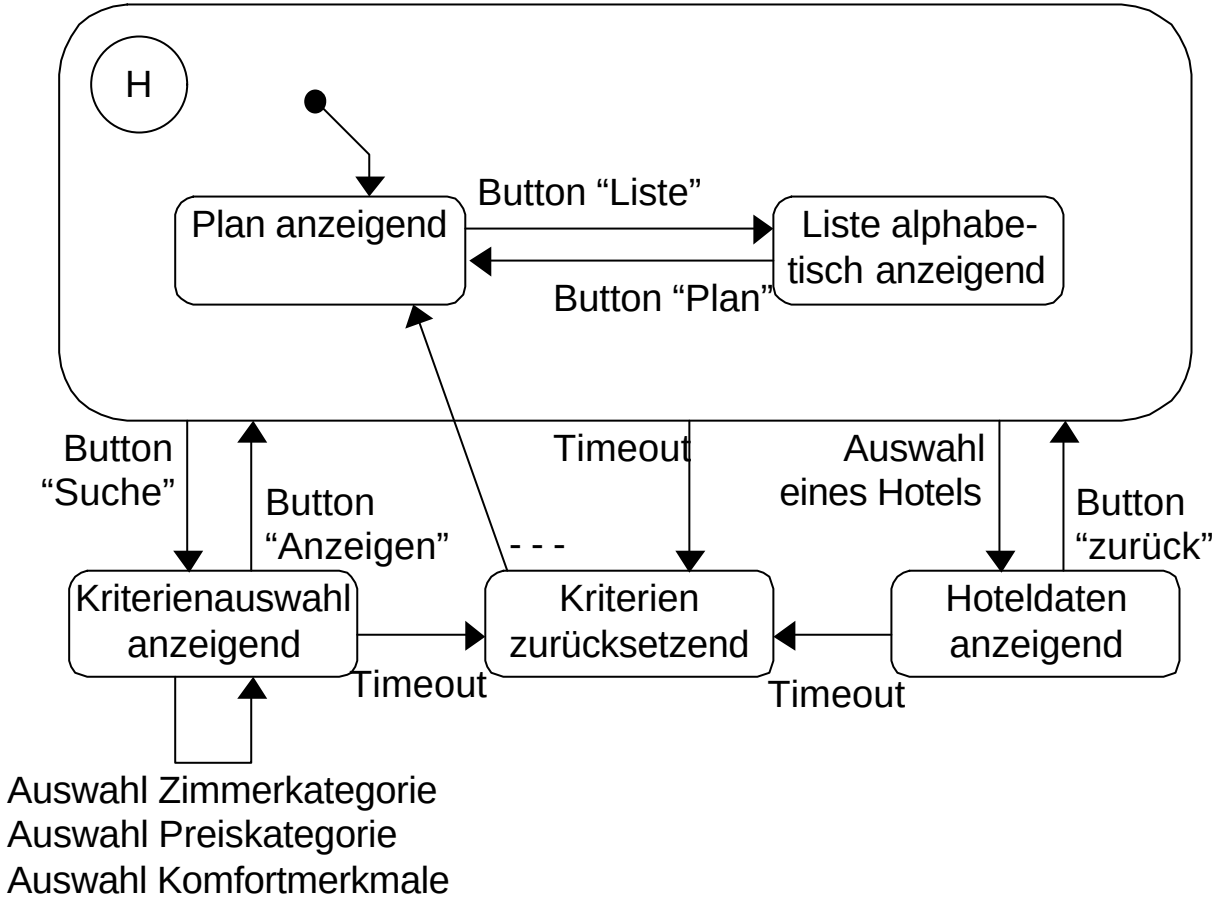


Harel-Automat (6)

- Beispiel
  - Szenario
    Eine Stadt möchte ein Informationssystem einrichten. An verschiedenen Terminals im Stadtgebiet soll es möglich sein, Informationen über Sehenswürdigkeiten, Veranstaltungen, Hotellerie und Gastronomie abzurufen. ... An den Terminals ist es möglich, Hotels anhand einer alphabetischen Liste, anhand von bestimmten Komfortmerkmalen und anhand von Preiskategorien zu suchen. Dabei sollen die Suchergebnisse in einem Stadtplan und als Liste angezeigt werden.
  - Terminal als Harel-Automat
    Das Terminal (Teilsystem Hotellerie) wird beschrieben als Zustands-automat, der im Ruhezustand den Stadtplan mit der Lage aller Hotels zeigt. In jedem Zustand werden die jeweils möglichen Handlungen in Form von Schaltflächen angezeigt.

Harel-Automat (7)

– Zustandsdiagramm des Terminals

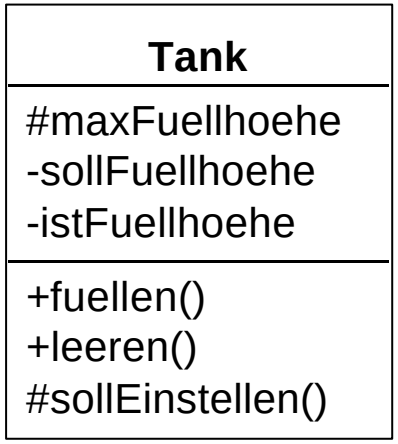


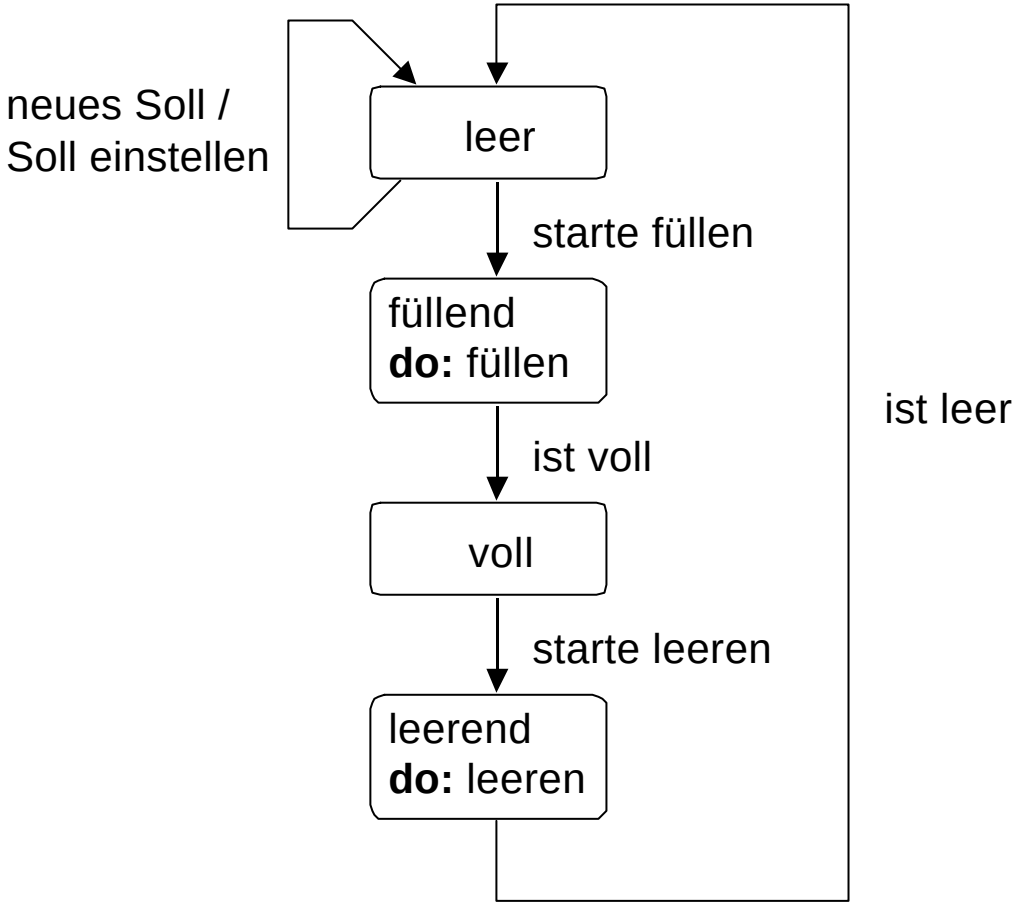
Modellierung von Objektzuständen

- Zustand beschreibt Eigenschaften des Objekts zu einem bestimmten Zeitpunkt
  - Attributwerte
  - Beziehungen zu anderen Objekten
- Zustandsname soll daher kein Verb sein, sondern Adjektiv / Partizip
- Ereignis tritt zu einem bestimmten Zeitpunkt auf, keine Dauer
  - neuer Attributwert
  - Signal (z.B. Abschluß einer Operation)
- Ereignis häufig als Botschaft anzusehen
  - Name kann entfallen, da Aktion (=Operation) gleichnamig

Modellierung von Objektzuständen (2)

- Beispiel eines Objekt-Lebenszyklus





Aktivitätsdiagramme

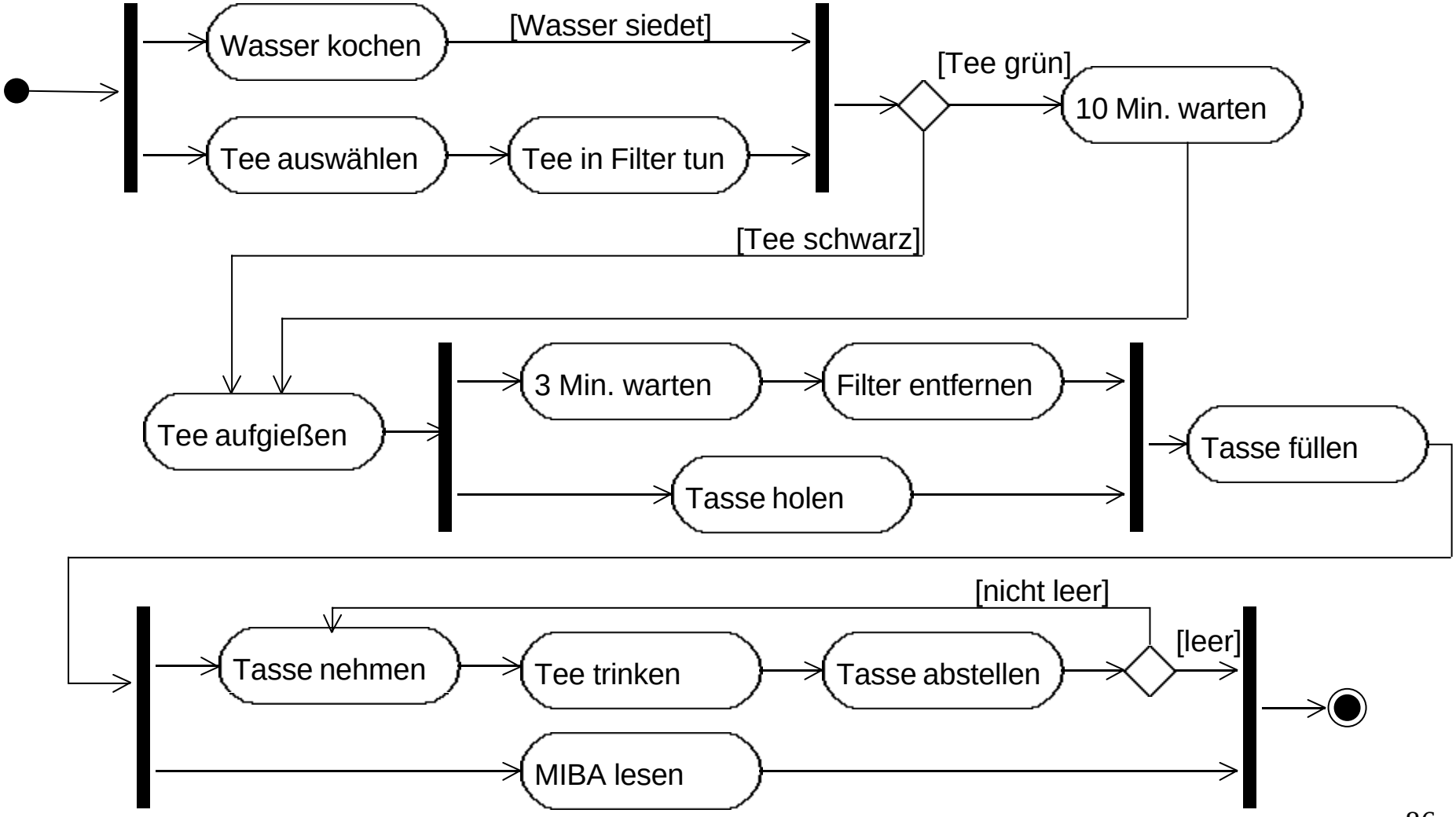
- Aktivitäten und Übergänge
- Verantwortlichkeiten
- Objektzustände
- Signalaustausch
- Ereignis-Prozeß-Ketten

Aktivitäten und Übergänge

- Aktivität
  ist Zustand mit interner Aktion
  - entspricht Zustand bei Moore-Automaten
- Übergänge modellieren Kontroll- und Objektfluß, z.B.
  - bei Anwendungsfällen
  - beim Zusammenspiel verschiedener Methoden
  - zur Beschreibung des Ablaufs in einer Methode
- Übergänge
  - nicht abhängig von Ereignissen
  - Bedingungen möglich
  - Aufteilung und Synchronisation durch Balken
    - zusätzlich Synchronisationsbedingungen möglich

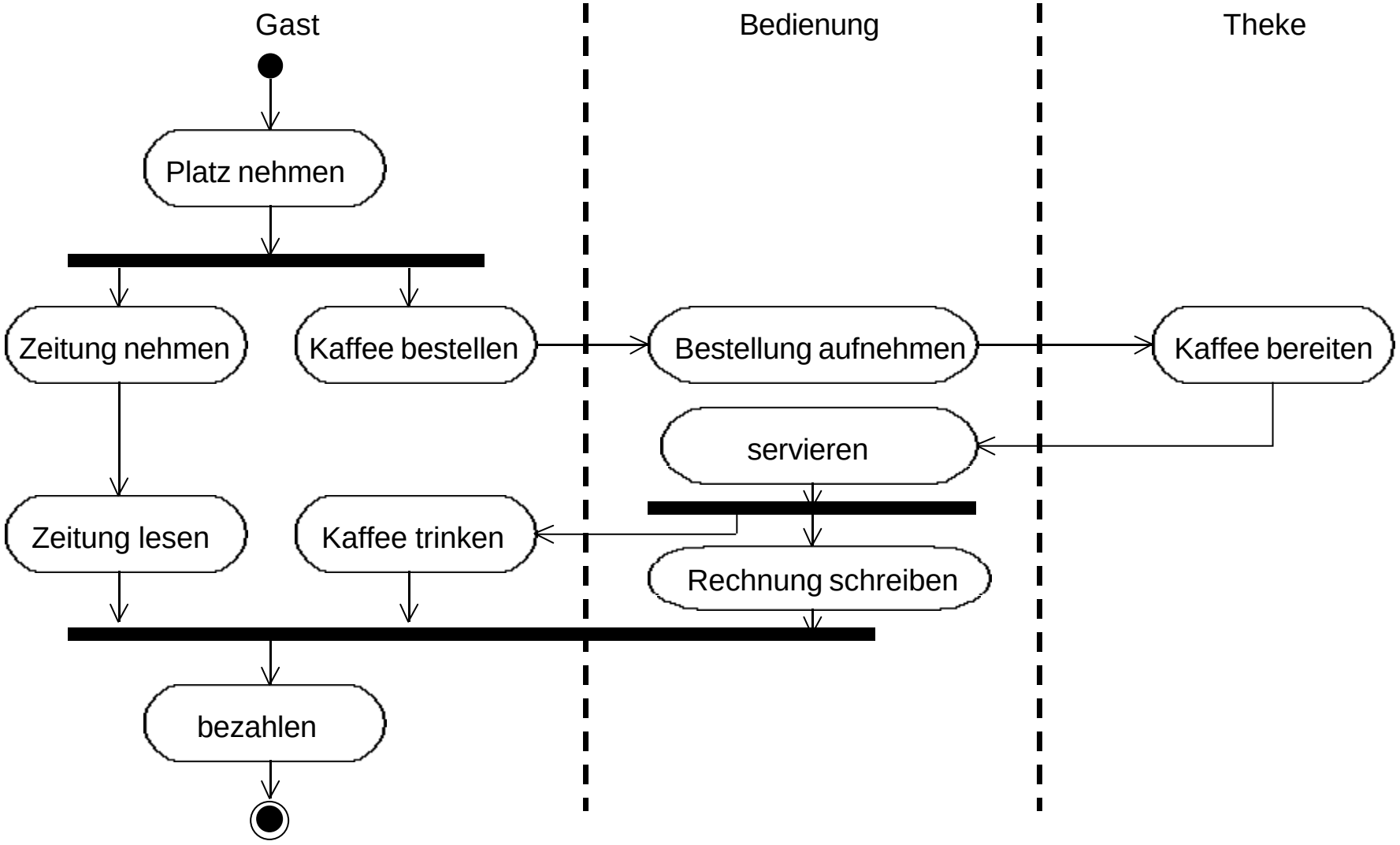
Aktivitäten und Übergänge (2)

- Beispiel: Teepause



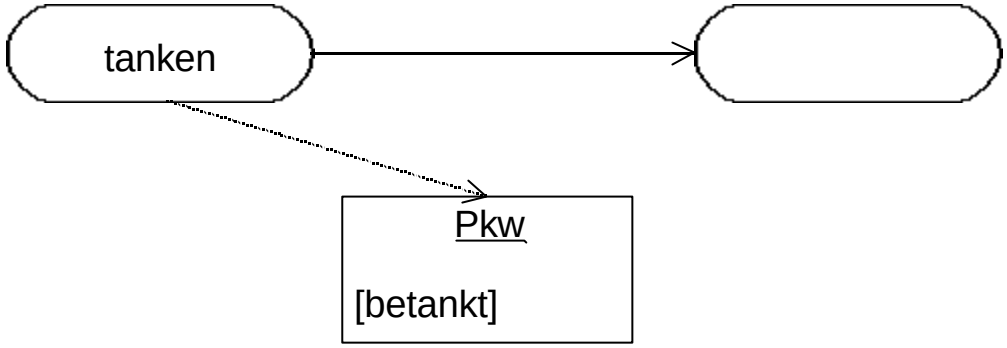
Verantwortlichkeiten

- Verantwortlichkeitsbereiche entsprechen Handelnden

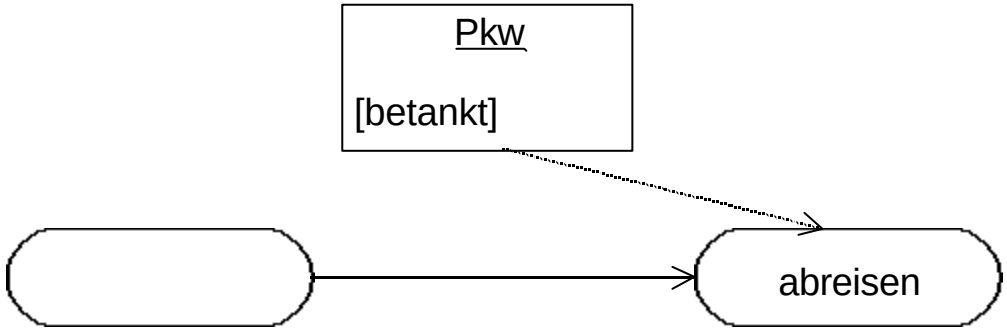


Objektzustände

- Zustand (Attribut) als Ergebnis einer Aktivität

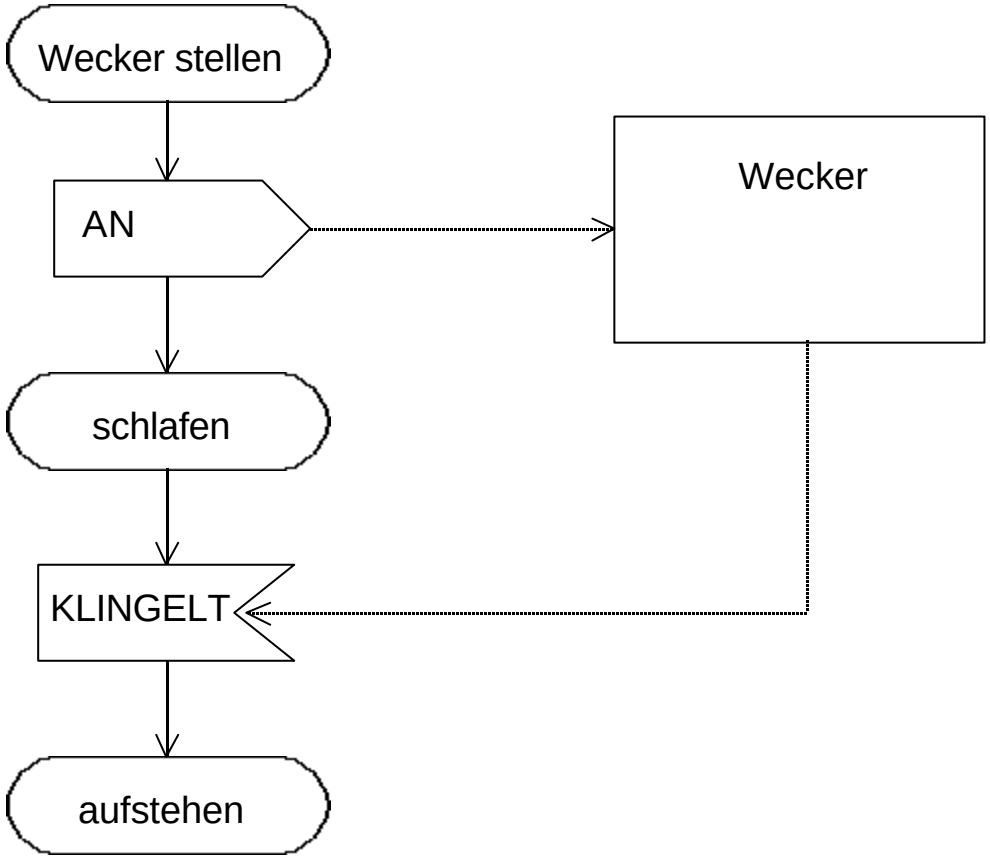


- Zustand (Attribut) als Bedingung einer Aktivität



Signalaustausch

- Signale zur Kommunikation mit externen Objekte



Ereignis-Prozeß-Ketten

- Folge von Ereignissen und Prozessen
  - Prozeß wird durch Ereignis ausgelöst
  - Prozeß erzeugt bei Beendigung Ereignis
  - Prozeß kann Eingabedaten nutzen
  - Prozeß kann Ausgabedaten nutzen
  - Prozeß wird in Verantwortlichkeit einer Organisationseinheit ausgeführt
- Darstellung durch Aktivitätsdiagramme
  - Ereignisse ersetzt durch Zustände
  - Prozeß ist Aktivität
  - Daten als Objekte
  - Organisationseinheiten als Kommentare oder Verantwortlichkeitsspalten

Zusammenfassung

OOA

- verwendet Konzepte aus OO Programmierung:
  - Objekt, Klasse, Vererbung, Polymorphie, späte Bindung, Schnittstellen
- verwendet Konzepte aus ER Modellen:
  - Assoziation, Kardinalität
- und weitere:
  - Rollen
  - Aggregation, Komposition
  - Design Patterns, Muster
  - Anwendungsfälle, Szenarien
- ist verbunden mit UML Diagrammen als Menge von Notationen
  - Klassendiagramm,
  - Aktivitätsdiagramm, Zustandsdiagramm
  - Sequenzdiagramm, Kollaborationsdiagramm
  - Anwendungsfalldiagramm
  - Paketdiagramm

für die verschiedenen Sichten auf ein System