Objektorientierte Modellierung mit der UML



Martin Wolf

Net.ObjectDays 2000

Erfurt, 9. Oktober 2000

Inhalt

- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
- Modellverwertung

Inhalt

- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
- Modellverwertung

Grundlage der oo Modellierung

- Objekttechnologie

  stellt Mittel zur Verfügung, um Systeme (Realität bzw. Software) derart zu beschreiben, daß sie als Menge kooperierender Objekte organisiert werden können

- Metamodell der Objekttechnologie

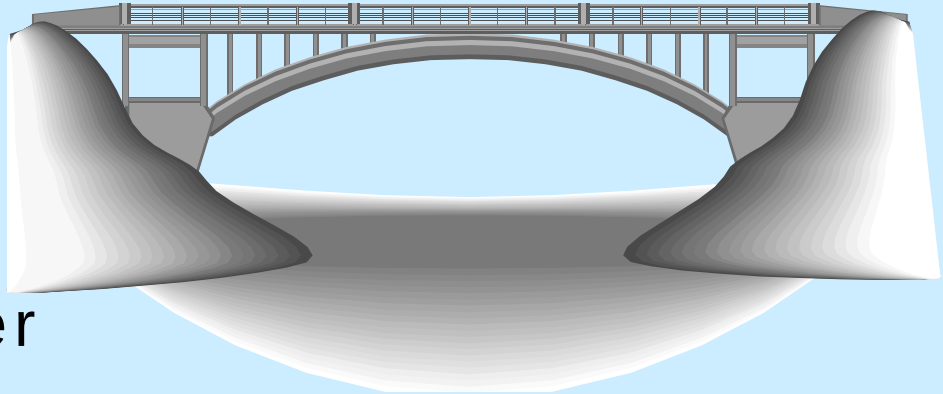
  Beschreibung der Konzepte (Basiselemente und Zusammenhänge)

- Modellierungssprache
  - Darstellung der Elemente
  - basierend auf dem Metamodell

Modellierung



Auftraggeber





Entwickler



Anwender

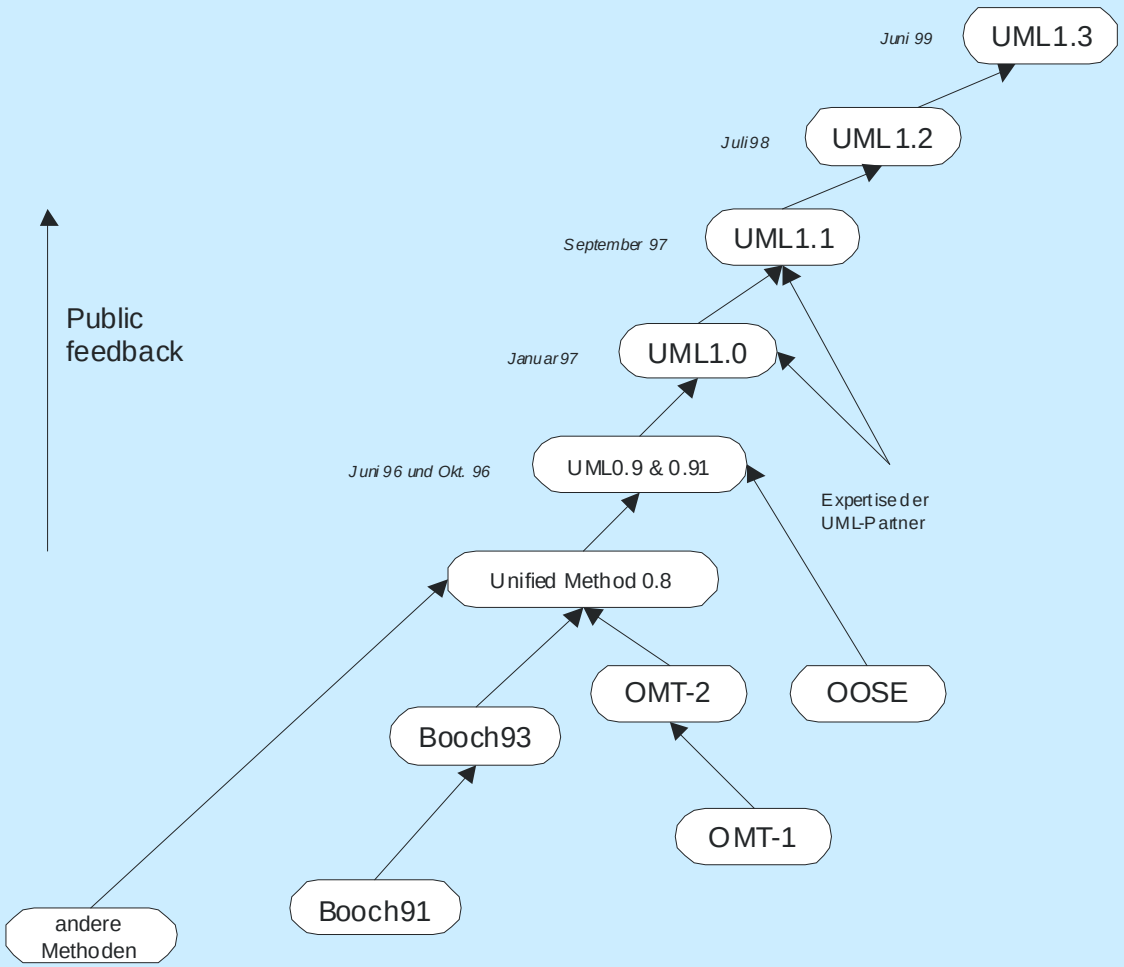
Unified Modeling Language

- Standard der Object Management Group (OMG)
- Graphische Modellierungssprache
  - Notation von Elementen
  - Notation von Diagrammtypen
- Strukturierung eines Modells
  - unterschiedliche Sichtweisen
  - Verminderung der Komplexität
- Anpassungen
  - Nutzergruppen

Unified Modeling Language

- Modellierungssprache
  - Spezifikation
  - Konstruktion
  - Visualisierung
  - Dokumentation
- Anwendungsgebiete
  - Software-Systeme
  - Geschäftsmodelle
  - andere „Nicht Software-Systeme“

Entwicklung der UML



Inhalt der UML

- UML Semantics
  - abstrakte Syntax
  - Wohlformuliertheitsregeln
  - Semantik
- UML Notation Guide
- UML Standard Profiles
- UML CORBAfacility Interface Definition
- UML XMI DTD Specification
- Object Constraint Language Definition

Ziele der UML

- Ausdrucksstarke Modellierungssprache zur Entwicklung und zum Austausch von Modellen
- Erweiterungs- und Spezialisierungsmöglichkeiten
- Unabhängigkeit der Spezifikation von Programmiersprache und Entwicklungsprozeß
- Formale Basis für das Verständnis der Modellierungssprache
- Entwicklung des Werkzeugmarktes
- Unterstützung von High-Level Konzepten

Abgrenzung UML

- Programmiersprachen
- Werkzeuge
- Entwicklungsprozeß

Literatur

- Martin Fowler, Kendall Scott, **UML konzentriert, Strukturierte Einführung in die Standard-Objektmodellierungssprache**, Addison-Wesley, 2000
- Grady Booch, James Rumbaugh, Ivar Jacobson, **Das UML-Benutzerhandbuch**, Addison-Wesley, 1999
- Bernd Oestereich, **Objektorientierte Softwareentwicklung, Analyse und Design mit der Unified Modeling Language**, Oldenbourg, 1998
- Rainer Burkhardt, **Die Unified Modeling Language, Objektorientierte Modellierung für die Praxis**, Addison-Wesley, 1999
- Horst A. Neumann, **Objektorientierte Softwareentwicklung mit der Unified Modeling Language (UML)**, Hanser, 1998
- Bernd Kahlbrandt, Software Engineering. **Objektorientierte Software-Entwicklung mit der Unified Modeling Language**, Springer, 1998

Verweise

- Object Management Group

  http://www.omg.org/uml
- UML Revision Task Force

  http://uml.shl.com/
- UML auf gut Deutsch

  http://www.system-bauhaus.de/uml/
- Übersicht

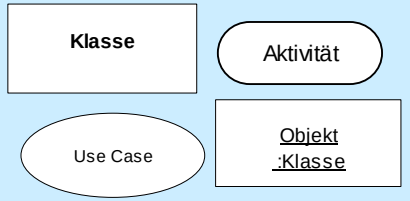
  z.B. http://www.jeckle.de/unified.htm

Inhalt

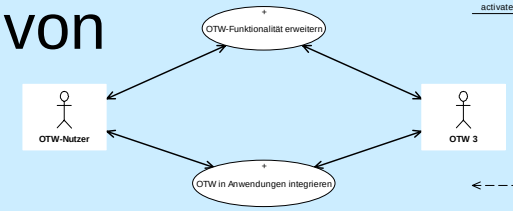
- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
- Modellverwertung

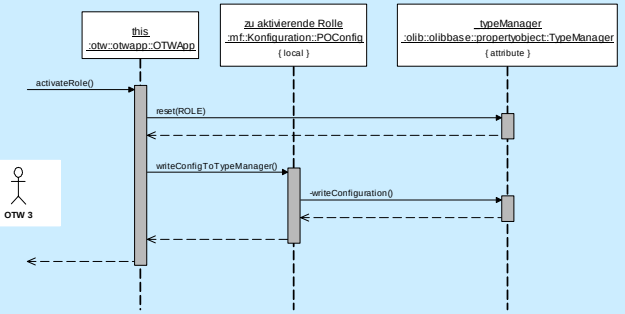
Beschreibungselemente

- Grundelemente
  - Basiselemente der Objekttechnologie

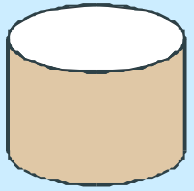


- Diagramme
  - Darstellung ausgewählter Aspekte
  - Zusammenfassung von Grundelementen

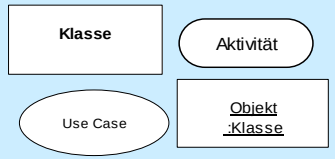




- Gesamtmodell
  - Grundelemente sind Bestandteile des Gesamtmodells
  - Diagramme definieren Sichten auf das Gesamtmodell



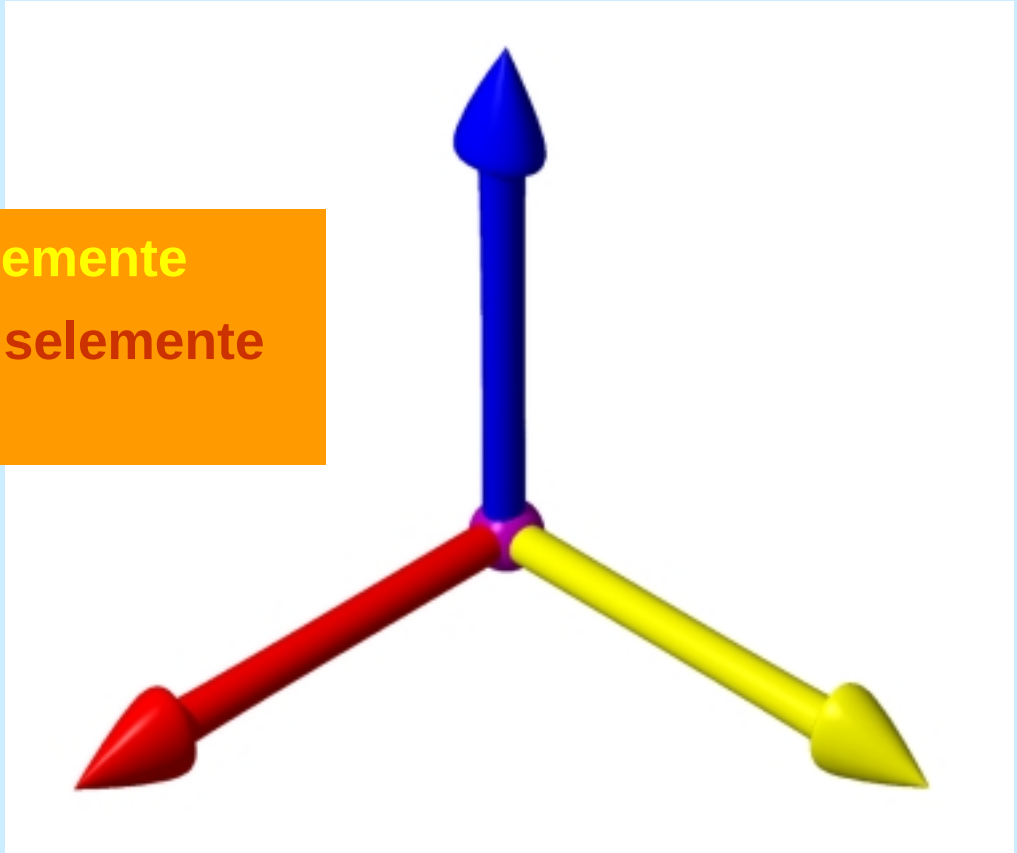
Grundelemente



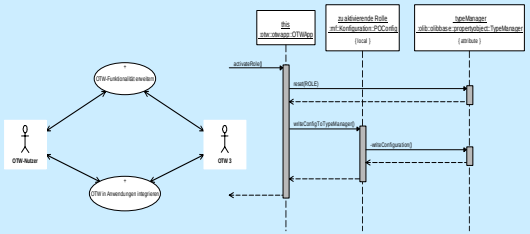
Statische Beschreibungselemente

Dynamische Beschreibungselemente

Architekturelemente



Diagramme



- Use-Case-Diagramm
- Sequenzdiagramm
- Kollaborationsdiagramm
- Aktivitätsdiagramm
- Zustandsdiagramm
- Klassendiagramm
- Komponentendiagramm
- Verteilungsdiagramm

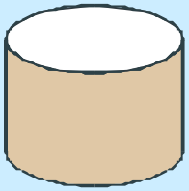
Use-Case-Diagramm

Verhaltensdiagramme

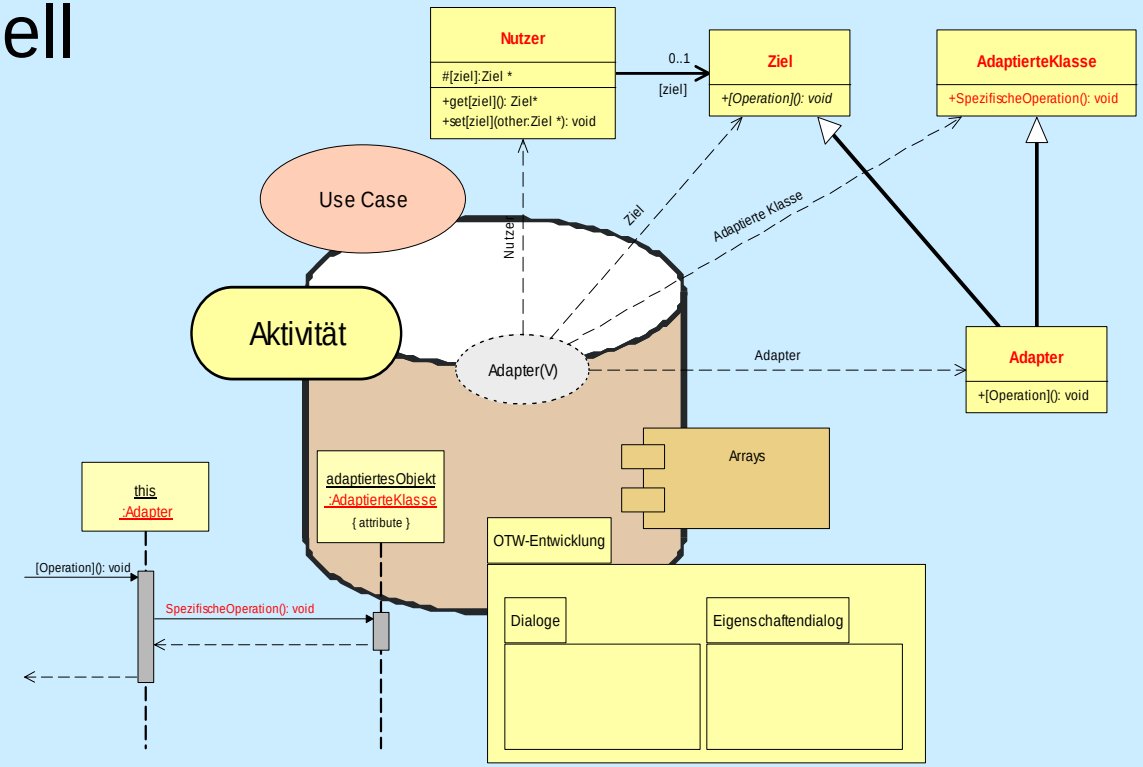
Strukturdiagramm

Architekturdiagramme

Gesamtmodell

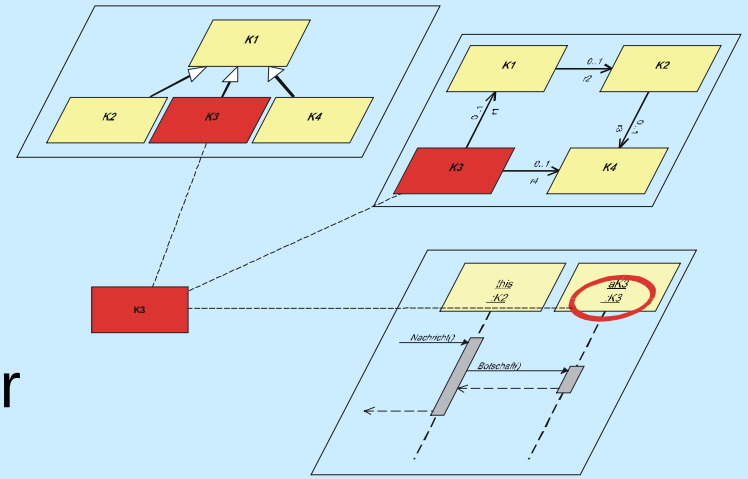


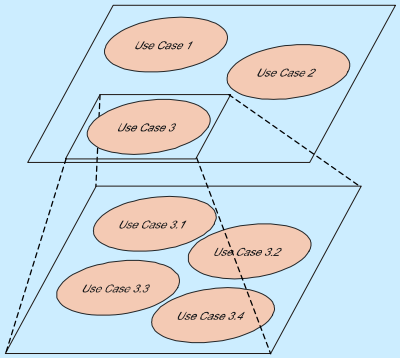
- Objektmodell
- Repository
- Konsistenz



Beziehungen

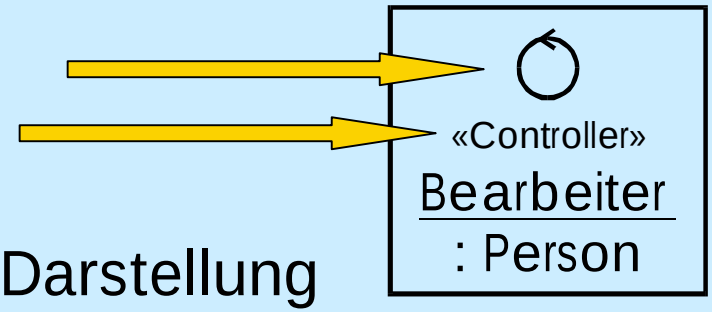
- Grundelemente
  - verschiedene Diagramme
  - verschiedene Diagrammtypen
  - Beziehungen entsprechend der Objekttechnologie
- Diagramme
  - eingebettete Grundelemente
  - Verfeinerung
- Gesamtmodell





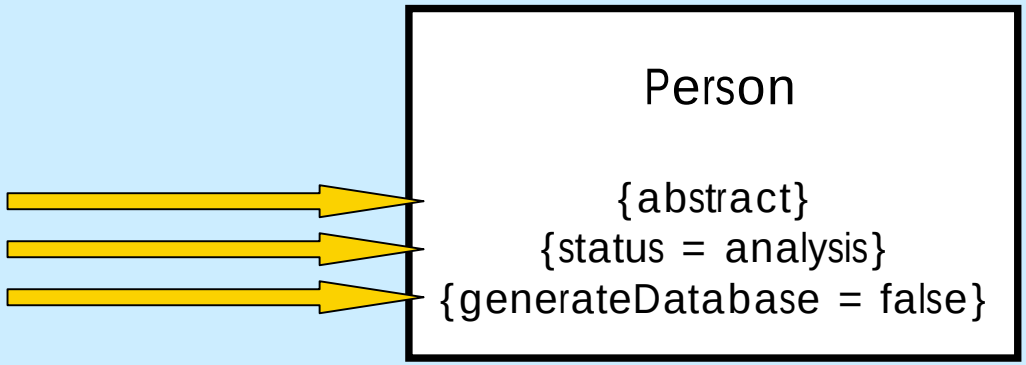
Erweiterungsmöglichkeiten - 1

- Stereotype
  - Meta-Modellelement
  - Zuordnung zu bestehenden Grundelementen
  - Einsatz während der Modellierung
  - Zusätzliche Restriktion, gleichbleibende Eigenschaft
  - Anpassung an Domänen
  - graphische oder textuelle Darstellung



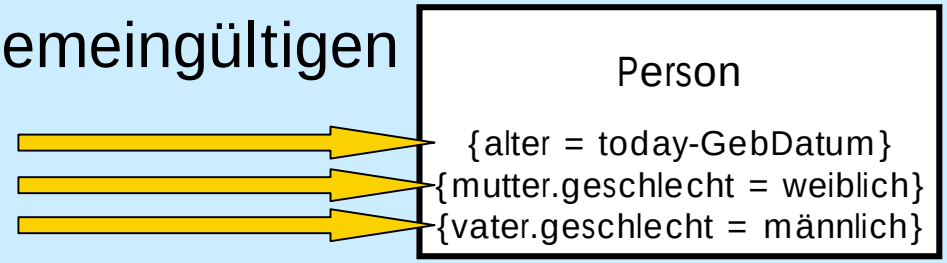
Erweiterungsmöglichkeiten - 2

- Eigenschaftswerte (Tagged Values)
  - Eigenschaften als Schlüssel-Wert-Paar
  - Zuordnung zu bestehenden Grundelementen



Erweiterungsmöglichkeiten - 3

- Zusicherungen - Constraints
  - Zuordnung zu bestehenden Grundelementen
  - Einschränkung des Allgemeingültigen
    - Bedingung
    - Integritätsregel
  - Notation: { Zusicherung }
    - frei formuliert
    - OCL-Ausdruck (Object Constraint Language)
  - Beispiel
    - Attribut - Wertmenge
    - Operation/Methode - Vor- und Nachbedingungen, Invariante



Inhalt

- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
- Modellverwertung

Diagrammtypen der UML

- Use-Case-Diagramm

Use-Case-Diagramm

- Sequenzdiagramm
- Kollaborationsdiagramm
- Aktivitätsdiagramm
- Zustandsdiagramm

Verhaltensdiagramme

- Klassendiagramm

Strukturdiagramm

- Komponentendiagramm
- Verteilungsdiagramm

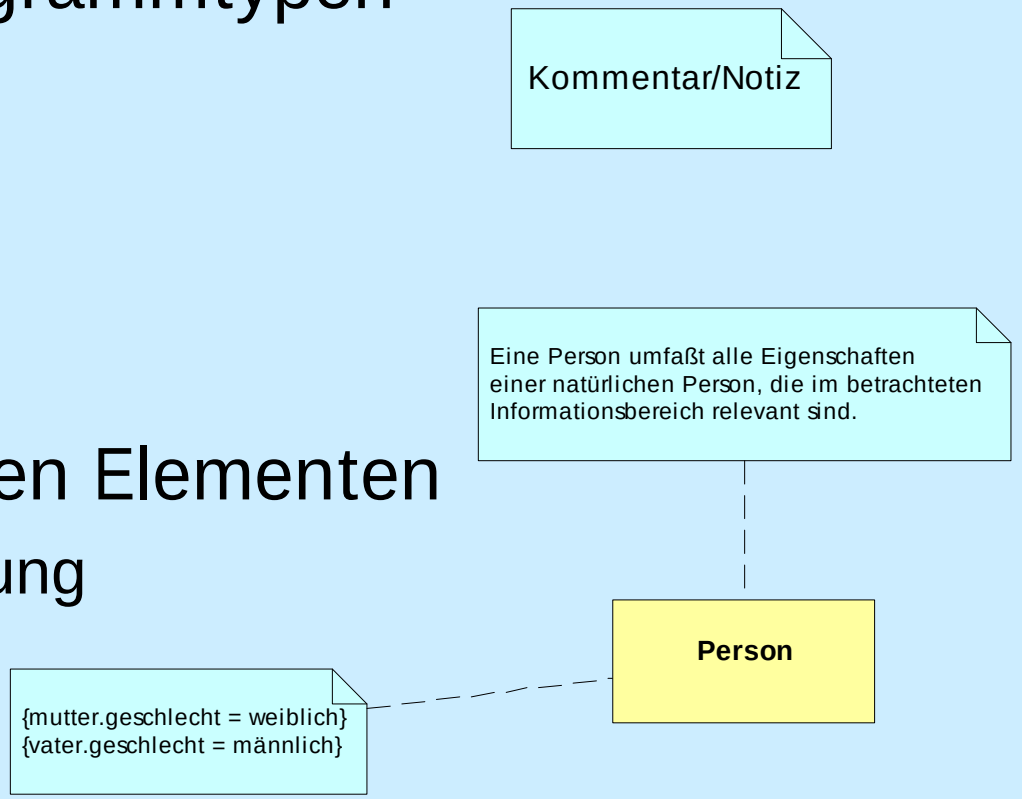
Architekturdiagramme

Strukturierung

- Zweck
- Graphische Darstellung
- Bestandteile
  - Elemente
  - Beziehungen
- Beispiel

Notiz

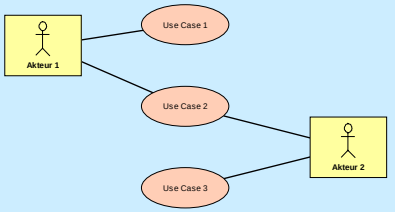
- Element in allen Diagrammtypen
- Ziel
  - Dokumentation
  - Anmerkungen
  - Zusicherung
- Zuordnung zu anderen Elementen
  - Abhängigkeitsbeziehung



Inhalt

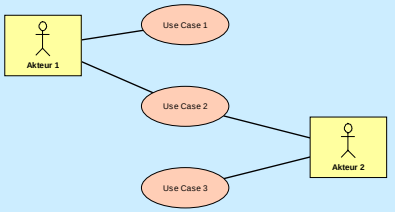
- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
- Modellverwertung

Use-Case-Diagramm - 1



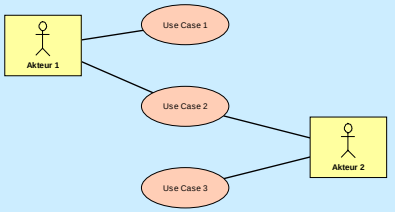
- Anwendungsfälle des Systems
  - Welche Aufgaben erfüllt das System?
  - Welche Dienstleistungen erbringt das System?
- Beteiligte am System
  - Welche Personen bzw. andere Systeme sind zu berücksichtigen?
  - Welche Beteiligte haben Einfluß auf Anwendungsfälle?
  - Welche Beteiligten werden von Anwendungsfällen beeinflußt?

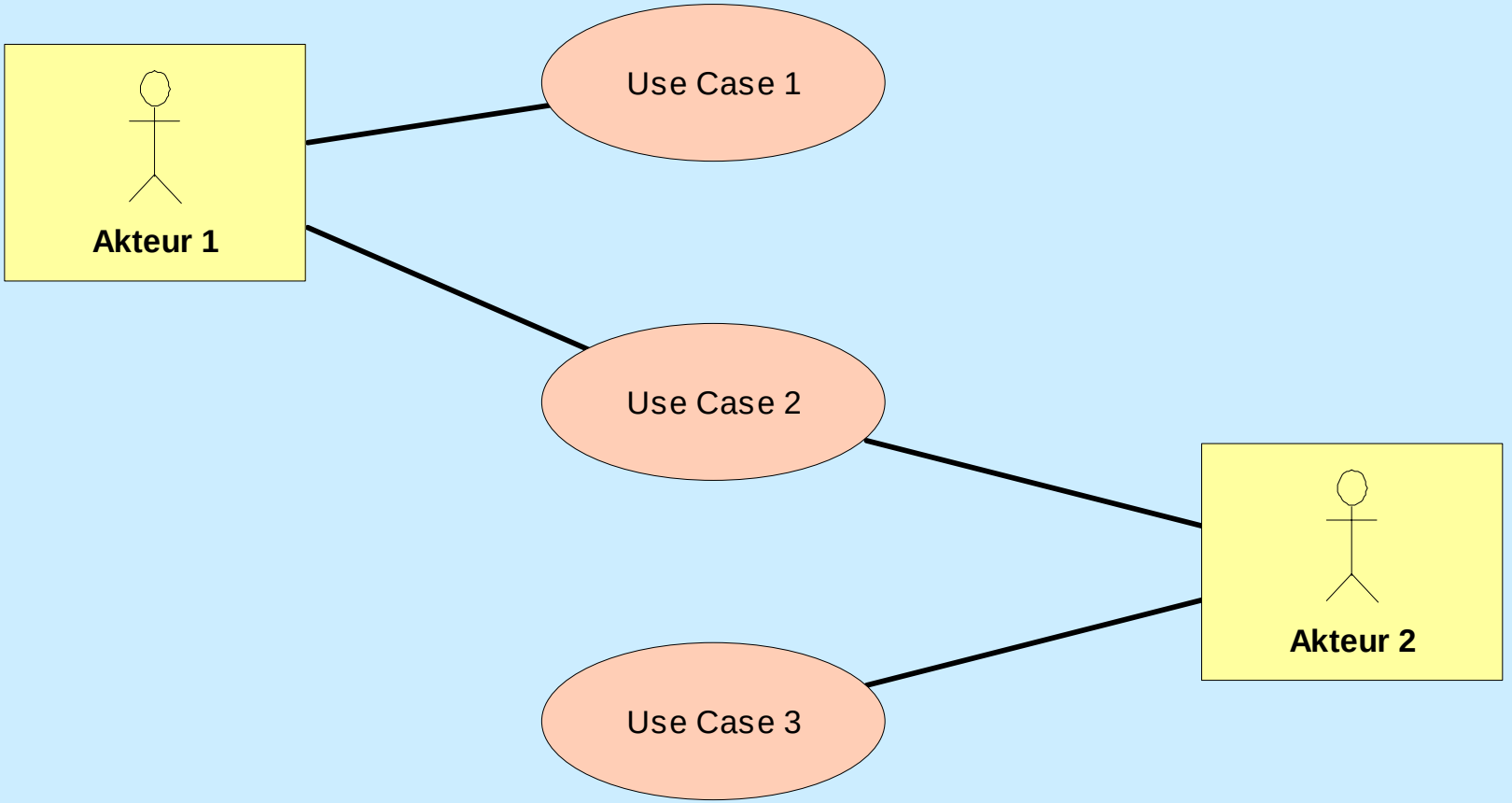
Use-Case-Diagramm - 2



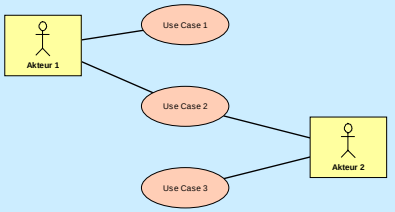
- Schnittstelle Anwender - System
  - klar
  - verständlich
  - übersichtlich
  - überschaubare Anzahl von Elementen
- Spezifikation mit Auftraggeber/Fachabteilung

Graphische Darstellung

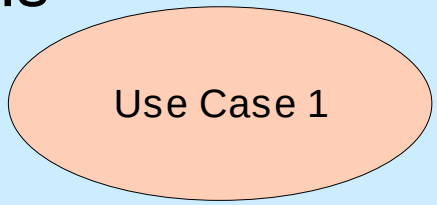




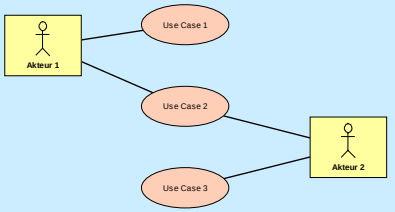
Use Case



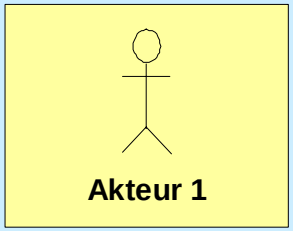
- Möglichkeit der Nutzung des Systems (Anwendungsfall)
- Repräsentation eines Arbeitsgangs
- „Systemverhalten aus der Sicht und in der Sprache der Anwender“
- Regeln:
  - Beschreibung einer Tätigkeit
  - Beschreibung des „Was“, nicht des „Wie“



Akteur

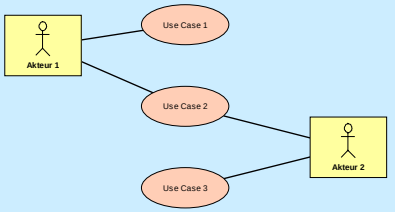


- Einheit, die im Zusammenhang mit einem Anwendungsfall eine Rolle spielt
  - Mensch
  - Maschine
  - System



«Actor»
Akteur 1

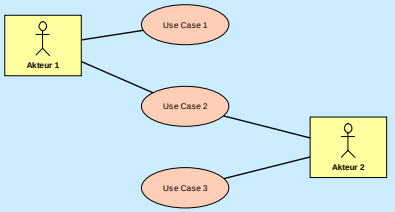
- Regeln:
  - Verwendung eines Substantivs
  - Informationsgeber oder -empfänger



Beziehung Use Case - Akteur

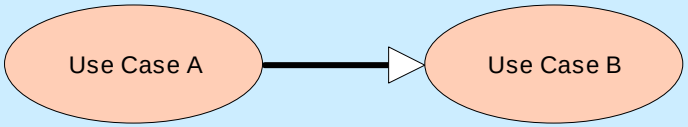
- Assoziation
  - Beteiligung eines Akteurs an einem Use Case
  - Kommunikation - Informationsaustausch (ggf. mit Angabe der Richtung)



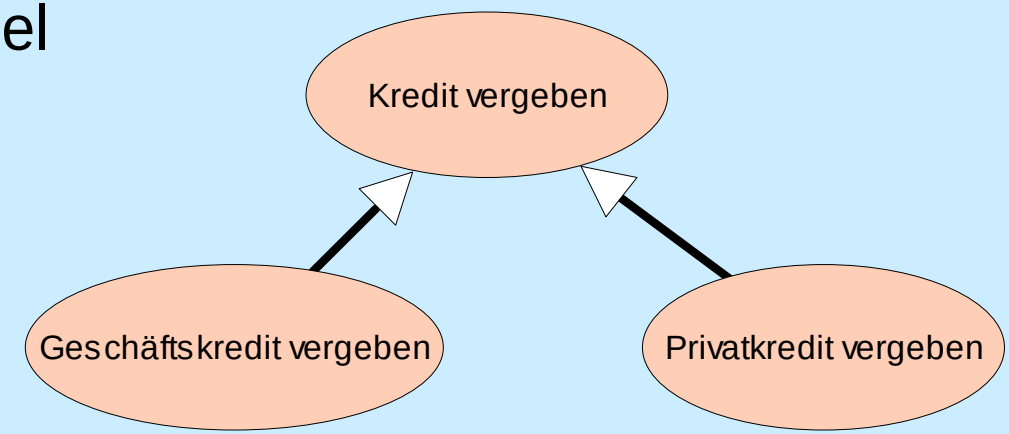


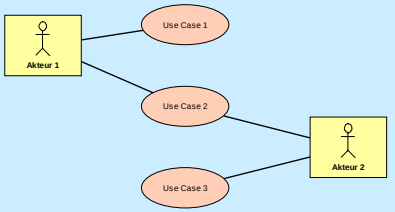
Beziehungen der Use Cases

- Generalisierung
  - Spezialisierung eines Anwendungsfalls
  - „A spezialisiert Verhalten von B“



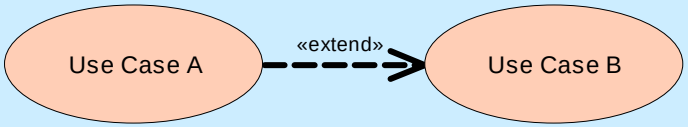
  - Beispiel



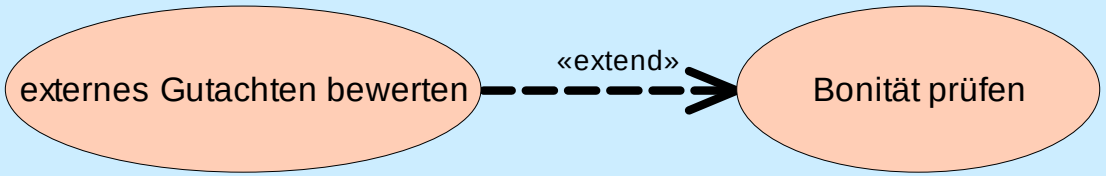


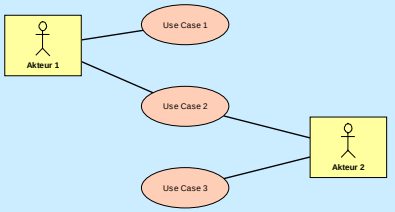
Beziehungen der Use Cases

- Extend
  - Erweiterung eines Anwendungsfalls
  - „B wird durch A erweitert“
  - Verwendung von Erweiterungs-punkten



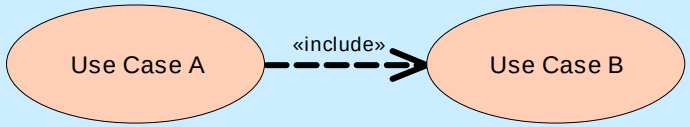
  - Beispiel



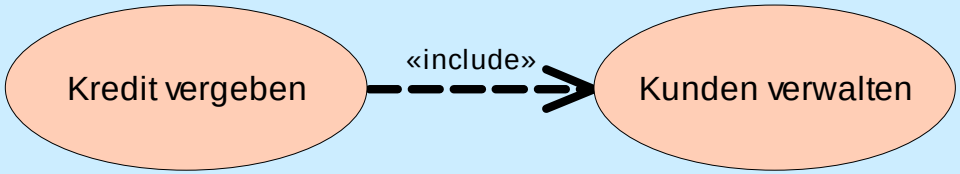


Beziehungen der Use Cases

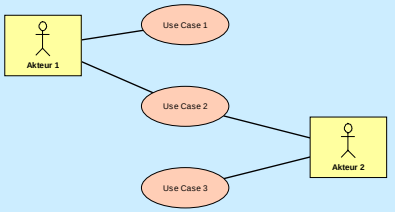
- Include
  - Einbindung eines Anwendungsfalls
  - „A enthält Verhalten von B“



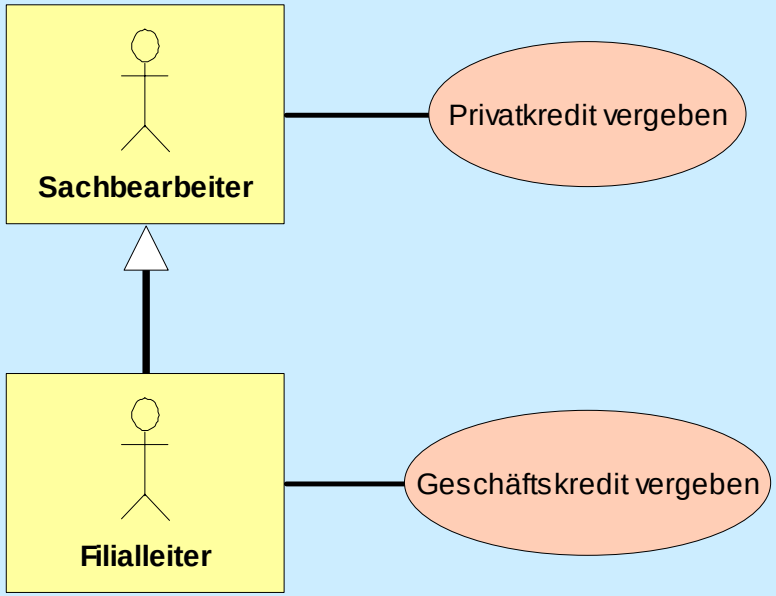
  - Beispiel



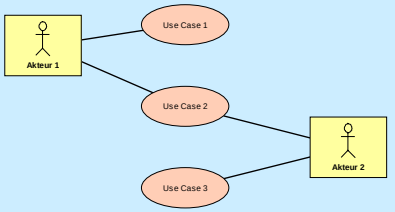
Beziehung Akteur - Akteur



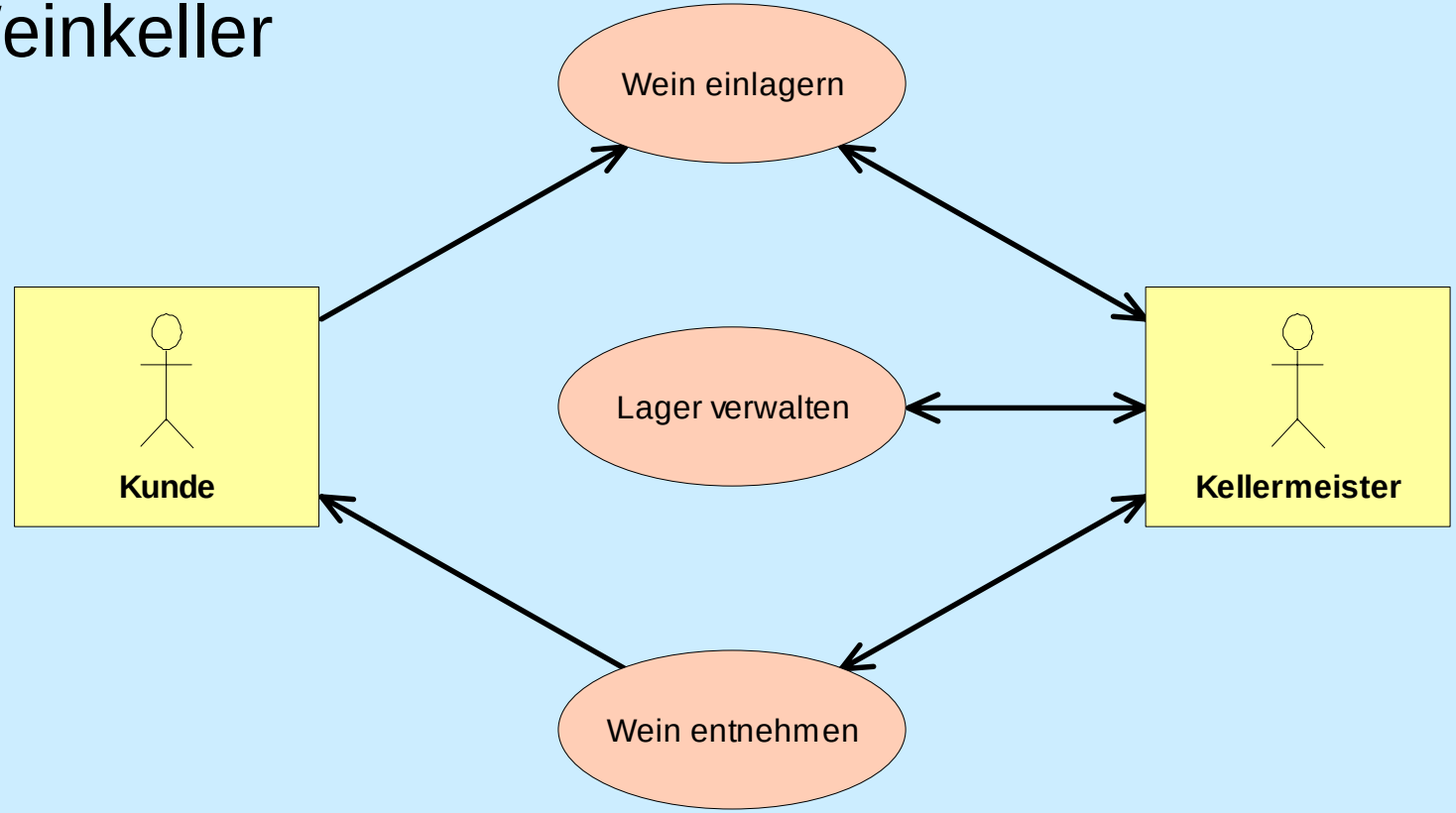
- Generalisierung
  - Spezialisierung (entsprechend Klassen)
  - auch der speziellere Akteur kann die Rolle des allgemeineren einnehmen



Beispiel



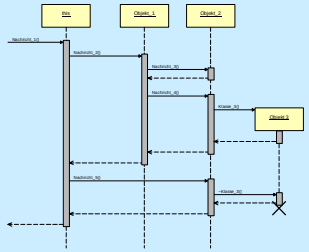
- Weinkeller



Inhalt

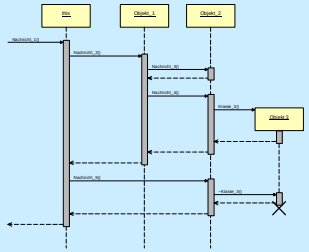
- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
    - Sequenzdiagramm
    - Kollaborationsdiagramm
    - Aktivitätsdiagramm
    - Zustandsdiagramm
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
- Modellverwertung

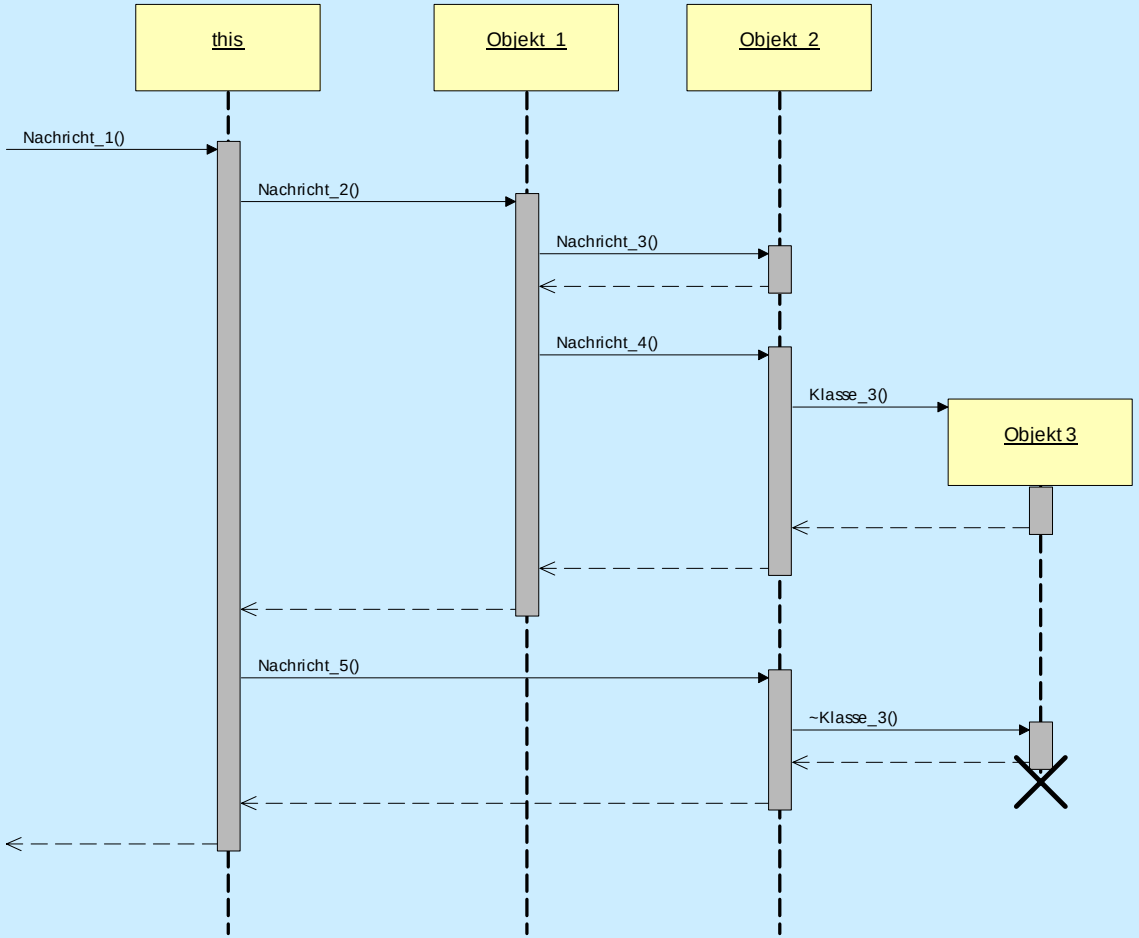
Sequenzdiagramm



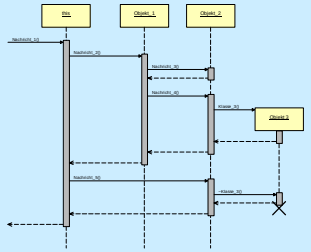
- Darstellung eines Szenarios
  - beteiligte Objekte mit Lebenslinie
  - Austausch von Nachrichten zwischen Objekten
  - Aktivierung von Methoden
  - Darstellung von Zusammenhängen des Nachrichtenaustauschs
  - zeitliche Reihenfolge der Kommunikation
  - Spezifikation von Zeiten (Restriktionen)
  - Fokus: sequentielle Abläufe

Graphische Darstellung





Objekt

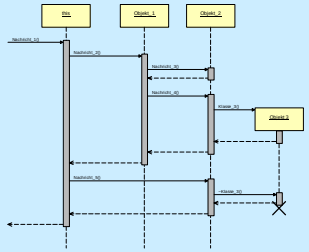


- Instanz einer Klasse
- Identität
- beinhaltet Attributwerte

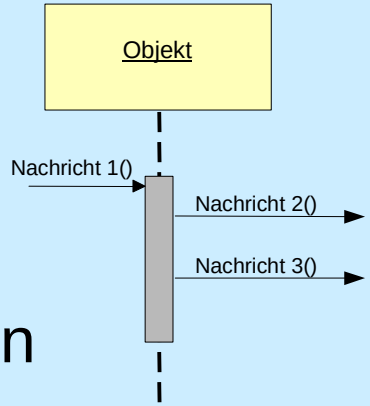
- beschreibt eine Rolle, die ein Objekt in einem Szenario spielt

Objekt	Sachbearbeiter
Objekt :Klasse	Sachbearbeiter :Person
Objekt :Klasse Attribut = Wert	Sachbearbeiter :Person Alter = 42

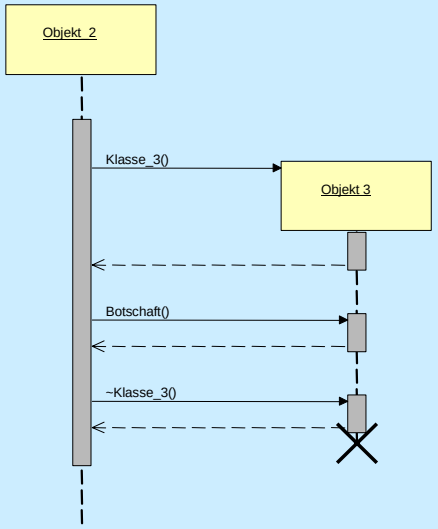
Objekt - Lebenslinie



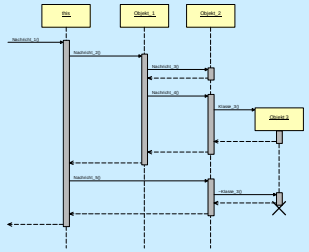
- Objekt
  - Lebenslinie
  - Aktivierung (Steuerungsfokus)
  - eingehende, abgehende Nachrichten



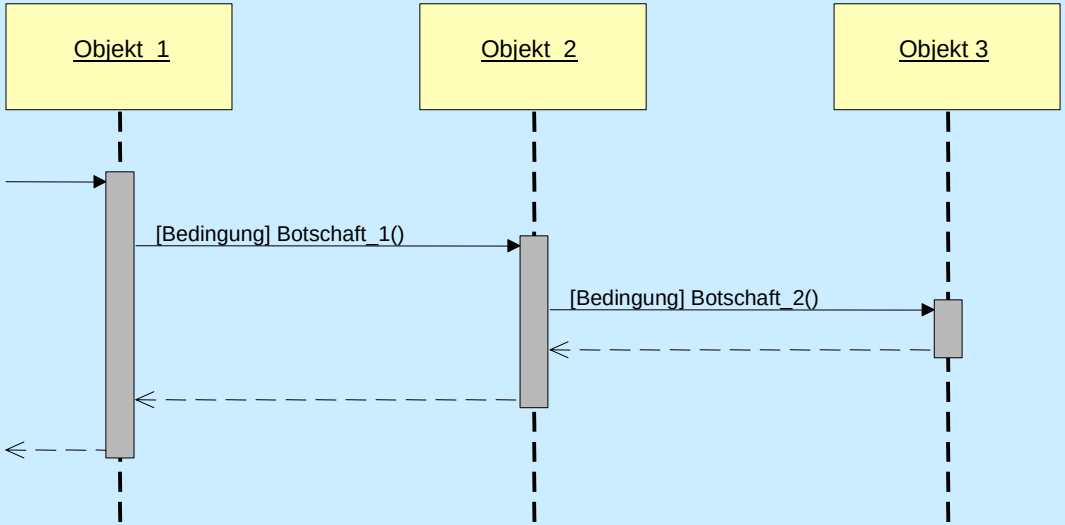
- Objektlebenszyklus
  - Erzeugung
  - Zerstörung



Botschaft/Nachricht

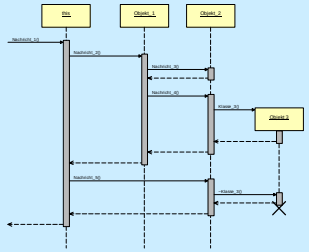


- Botschaft

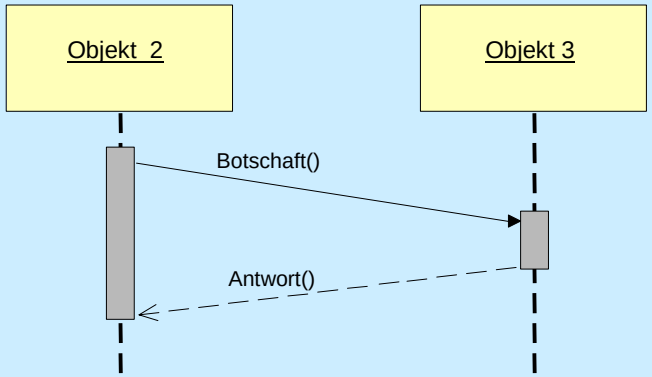


  - Senden einer Nachricht an ein Objekt
  - Angabe von Bedingungen
  - Reaktion durch Ausführung einer Methode
  - Antwortnachricht möglich

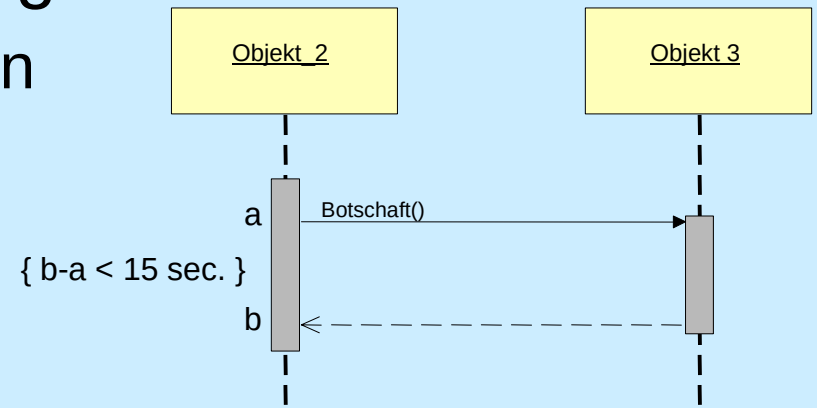
Zeitbedingungen



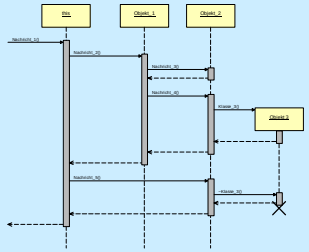
- Verzögerungen
  - Laufzeit von Botschaften



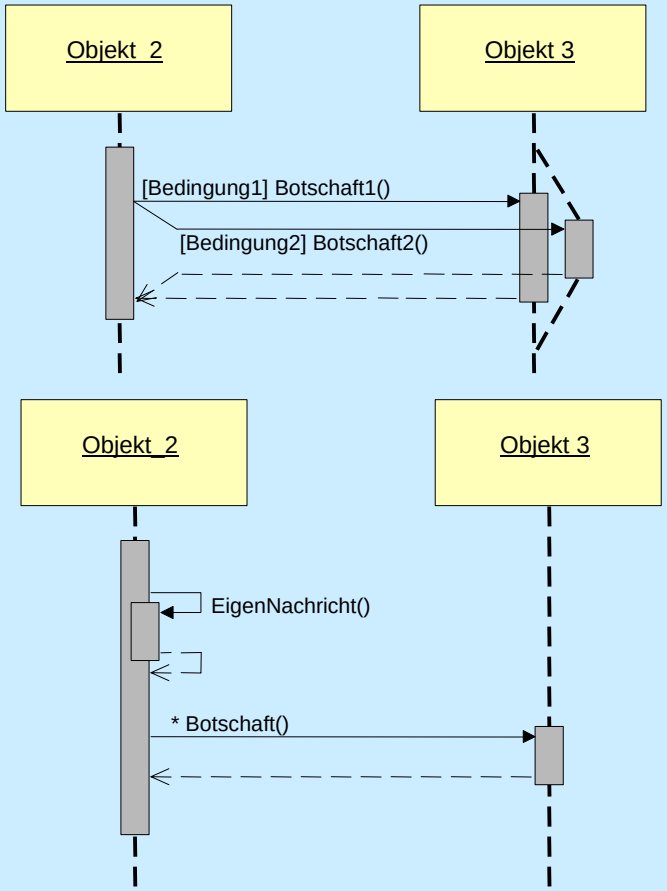
- Angabe von Zusicherungen
  - Abhängigkeiten zwischen verschiedenen Botschaften



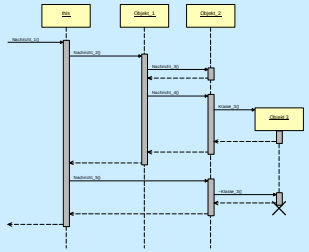
Spezielle Botschaften



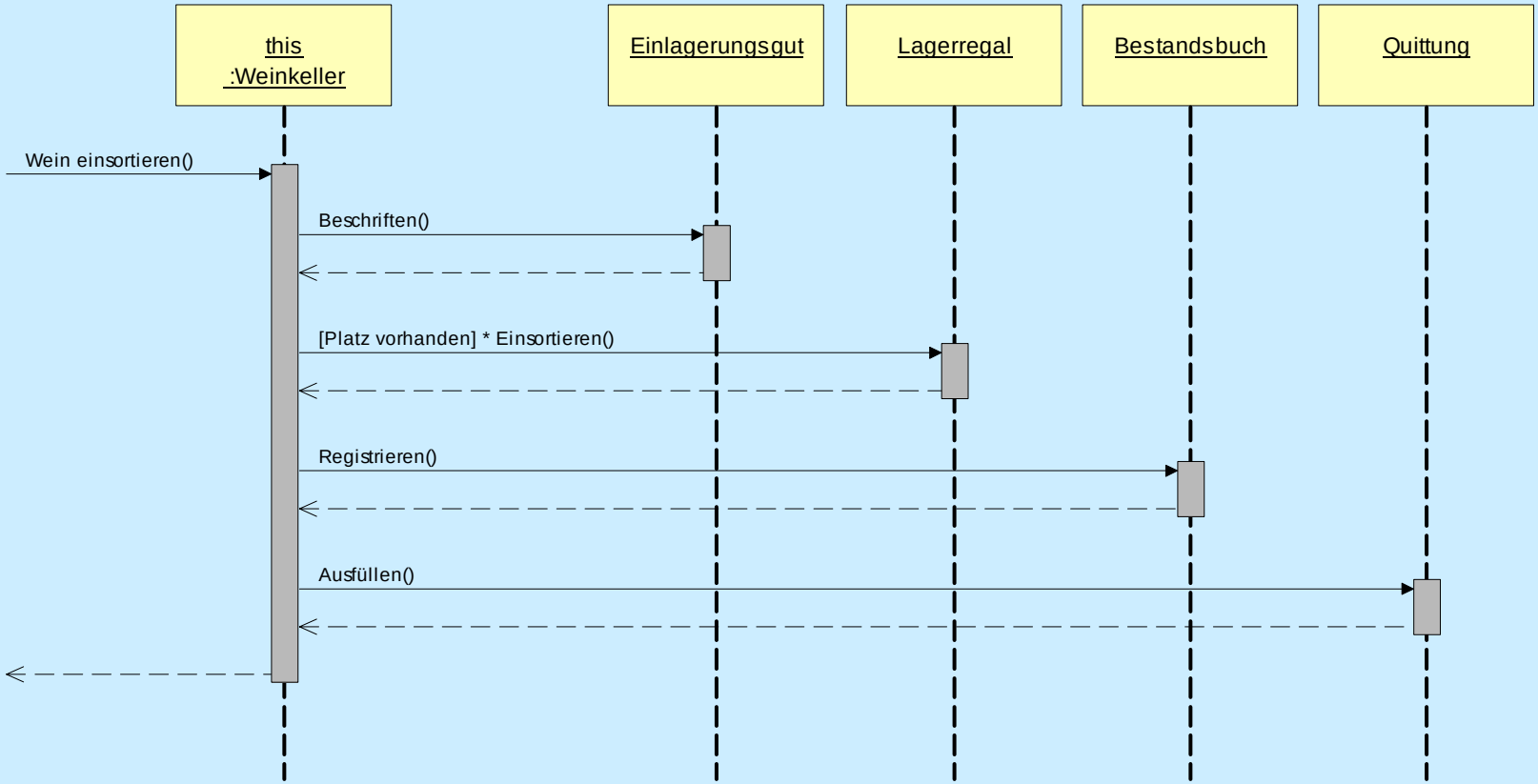
- Aufspaltung der Lebenslinie
  - Alternative Abläufe
  - Parallele Abläufe
- Selbstdelegation
  - Versenden einer Botschaft an das eigene Objekt
- Iteration



Beispiel



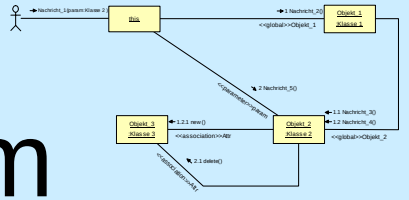
Weinkeller - Wein einsortieren



Inhalt

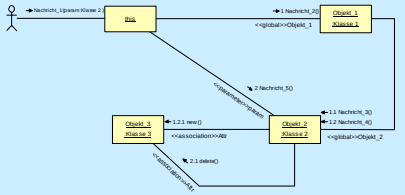
- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
    - Sequenzdiagramm
    - Kollaborationsdiagramm
    - Aktivitätsdiagramm
    - Zustandsdiagramm
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
- Modellverwertung

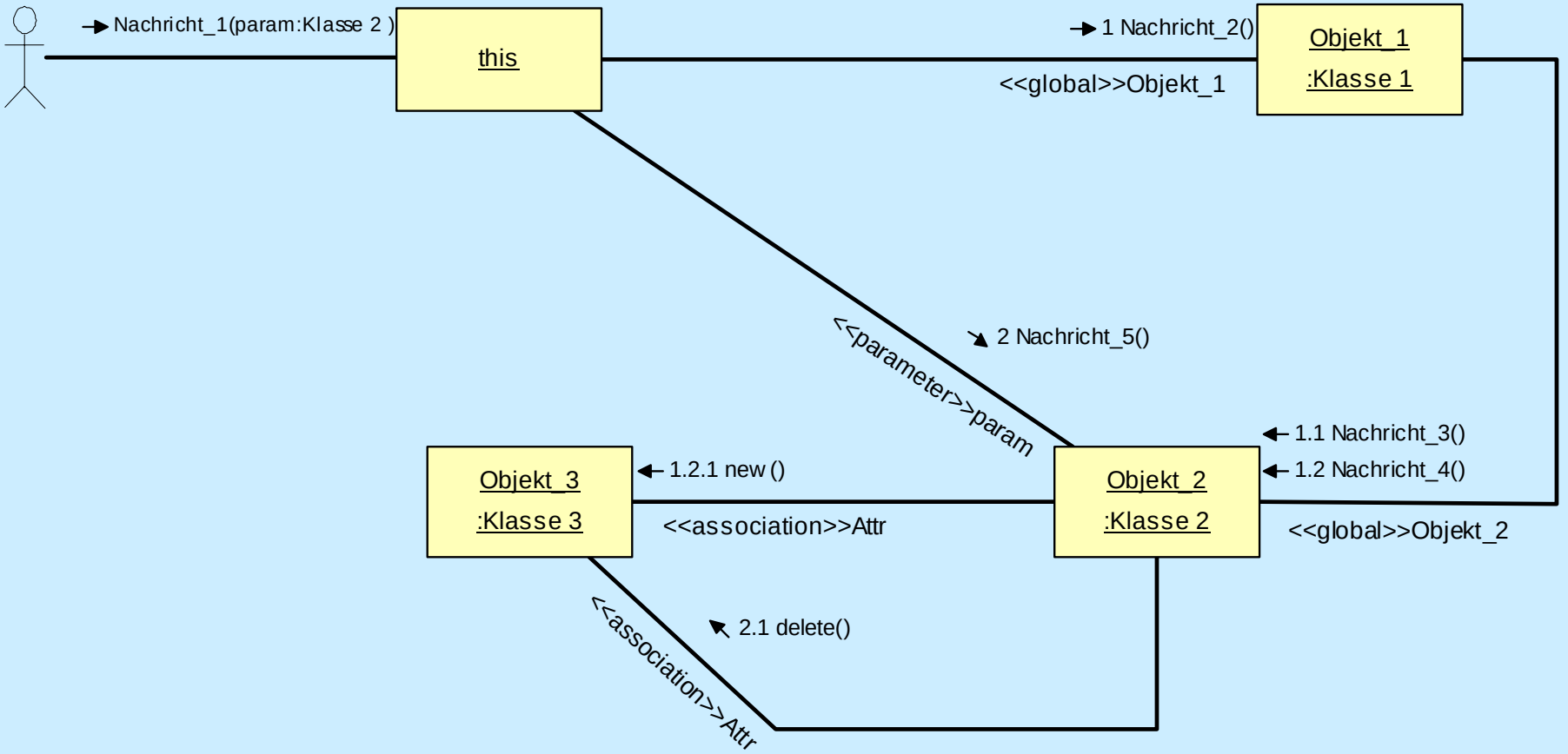
Kollaborationsdiagramm



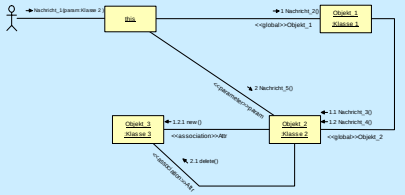
- Darstellung einer Szenarios
  - beteiligte Objekte
    - Zugriffspfade, Links
  - aktuelle Beziehungen zwischen Objekten
    - Rollen in einer Interaktion
  - Austausch von Nachrichten zwischen Objekten
    - Zusammenarbeit - Kollaboration
    - keine explizite Zeitachse
  - Bezüge zu Sequenzdiagramm
    - Interaktionsmodell (Sequenz- und Kollaborationsdiagramm)
    - Zeitbezug - Objekte, die bestimmte Rollen ausfüllen

Graphische Darstellung



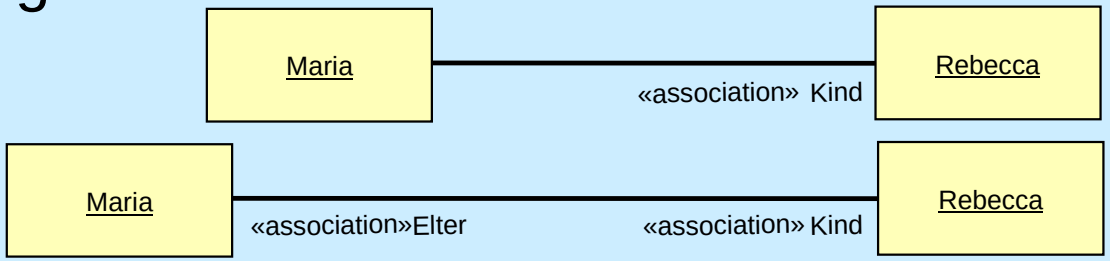


Objekt



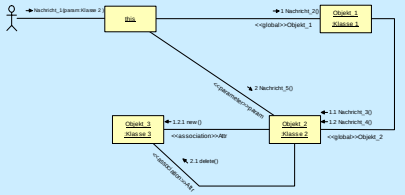
- Objekte
- Objektverbindungen
  - Zugriffspfade der Objekte
  - Sender - Empfänger einer Nachricht
  - Unidirektional
  - bidirektional



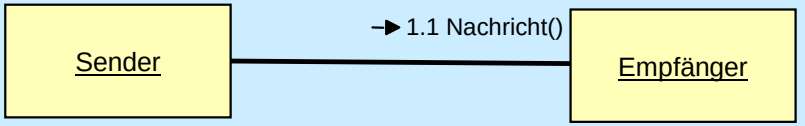


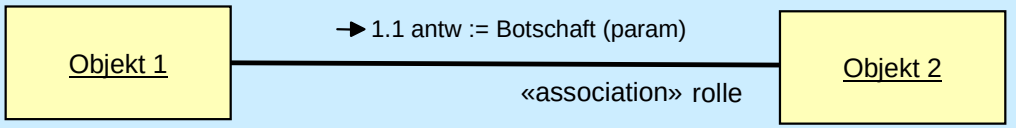
  - Kennzeichnung der Rolle
    - «association», «global», «local», «parameter», «self»

Interaktion

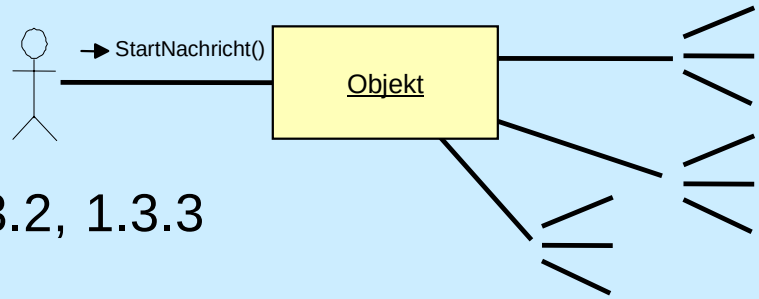


- Senden von Botschaften
  - Sender -> Empfänger
  - Entlang von Zugriffspfaden
  - Parameter
  - Rückgabewerte

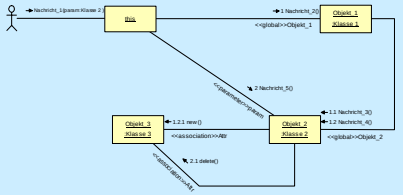




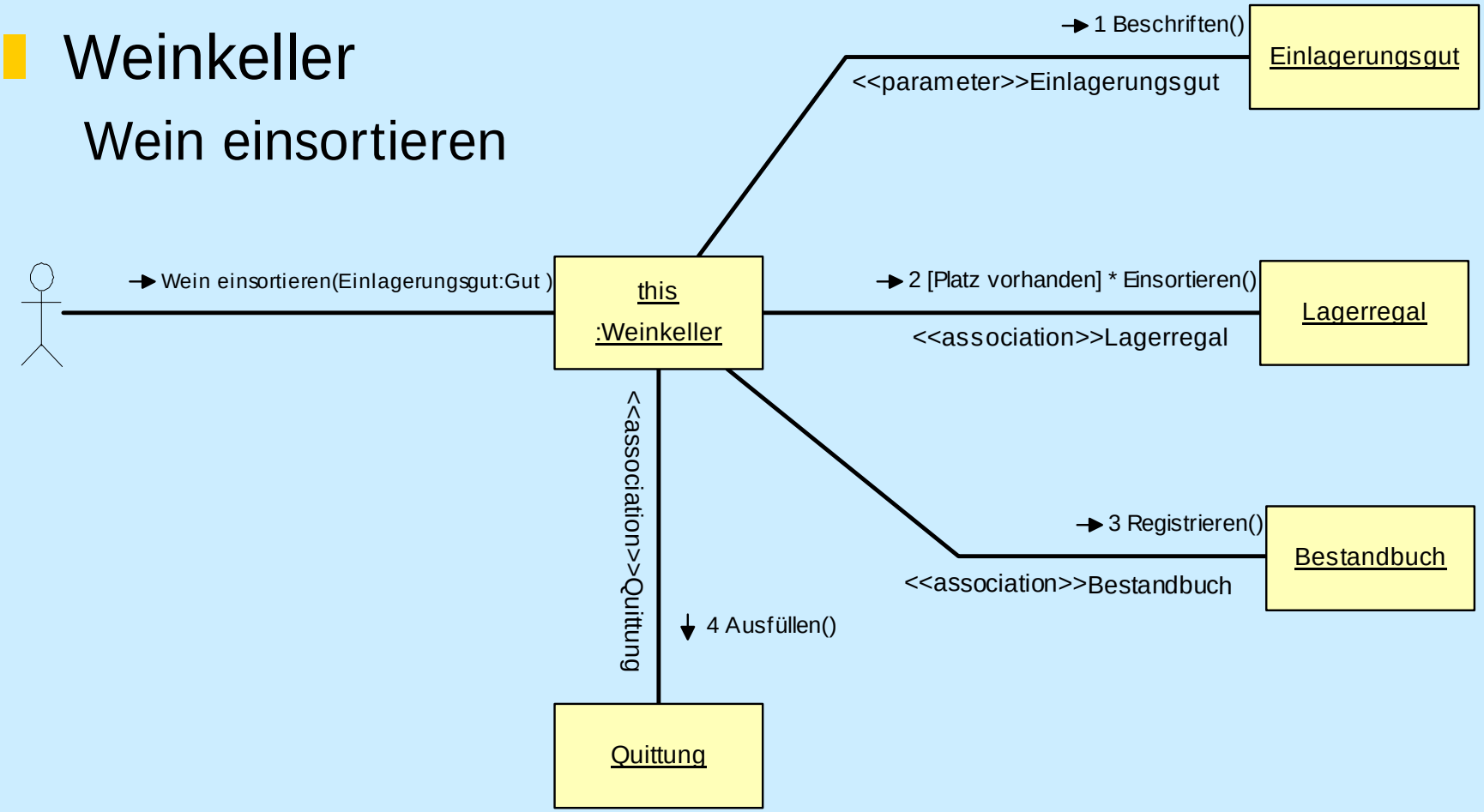
- Reihenfolge
  - Kaskadierung nach Startnachricht
    - 1, 2, 3
      - 1.1, 1.2, 1.3, 2.1, 2.2, ...
        - 1.1.1, 1.1.2, 1.3.1, 1.3.2, 1.3.3



Beispiel



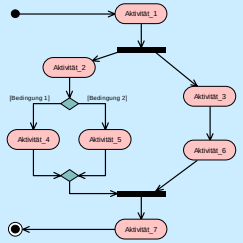
- Weinkeller
  - Wein einsortieren



Inhalt

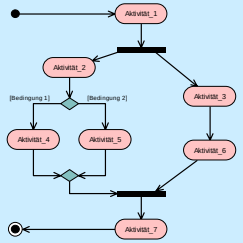
- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
    - Sequenzdiagramm
    - Kollaborationsdiagramm
    - Aktivitätsdiagramm
    - Zustandsdiagramm
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
- Modellverwertung

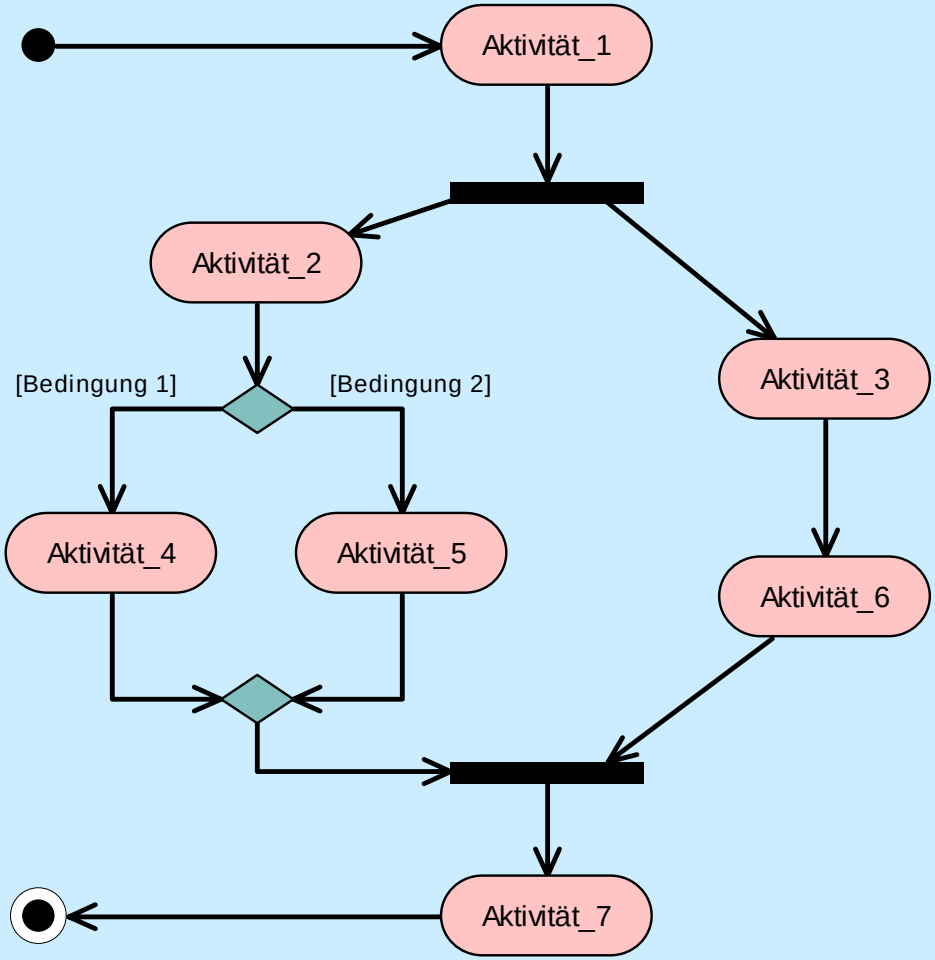
Aktivitätsdiagramm



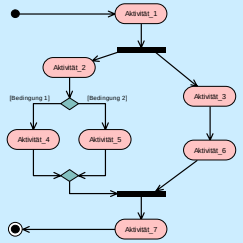
- Darstellung von Abläufen
  - Schritte eines Ablaufs
  - Abhängigkeiten der Aktivitäten
  - „Flußdiagramm“
  - Beschreibung z.B. einer Methode
  - Fokus:
    - Parallelitäten/Nebenläufigkeiten
    - Sequenzen
    - Verzweigungen
    - Schleifen

Graphische Darstellung



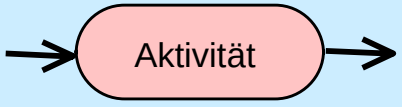


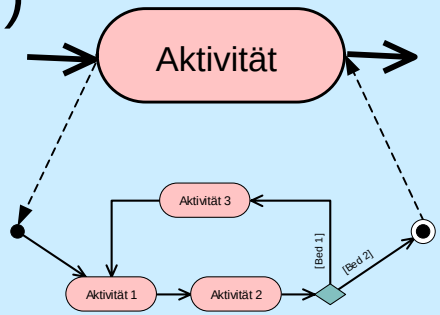
Aktivität



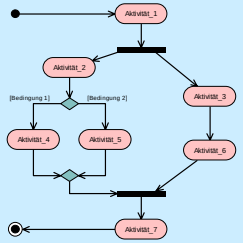
- Ausführungsschritt eines Algorithmus‘
- z.B. Aktivierung einer Methode für ein Objekt

- Differenzierung
  - Aktivitäten (Action State)
  - Subaktivitäten (Subactivity State)

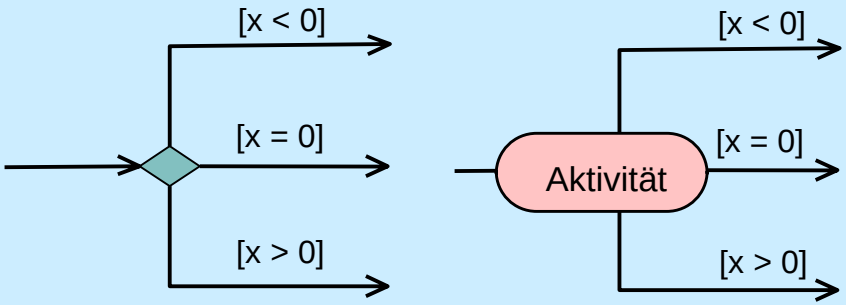




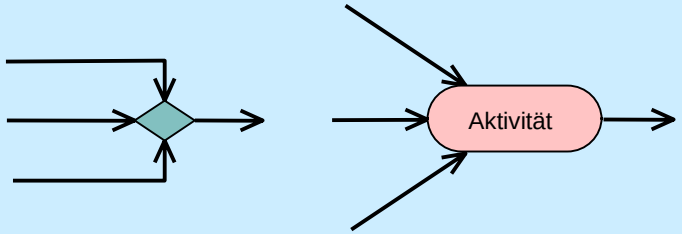
Bedingungen



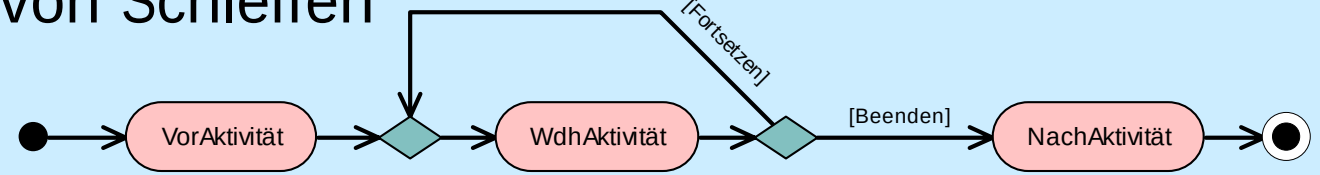
- Verzweigung-Alternativen
  - boolesche Ausdrücke als Bedingungen



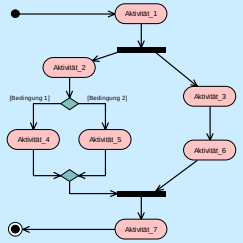
- Zusammenführung von Verzweigungen



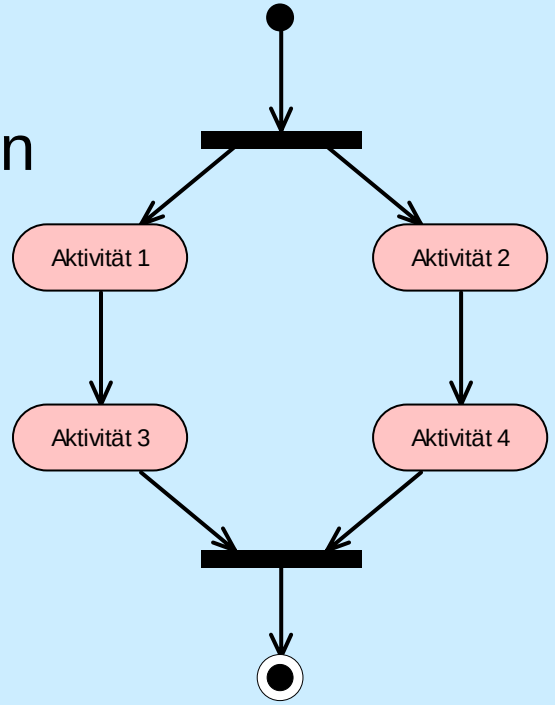
- Darstellung von Schleifen



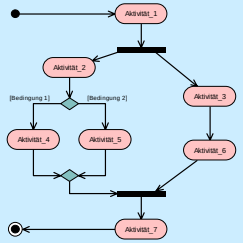
Nebenläufigkeit



- Unabhängigkeit/Parallelität von Aktivitäten
- Splitting
  - Aufspaltung des Ablaufes
  - Unabhängigkeit weiterer Aktivitäten
- Synchronisation
  - Zusammenführung eines zuvor aufgespalteten Ablaufes
  - Fortsetzung nach Abschluß aller unabhängigen Aktivitäten

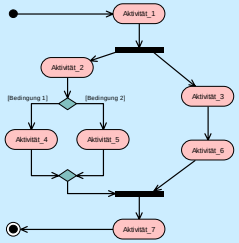


„Schwimmbahnen“ - 1

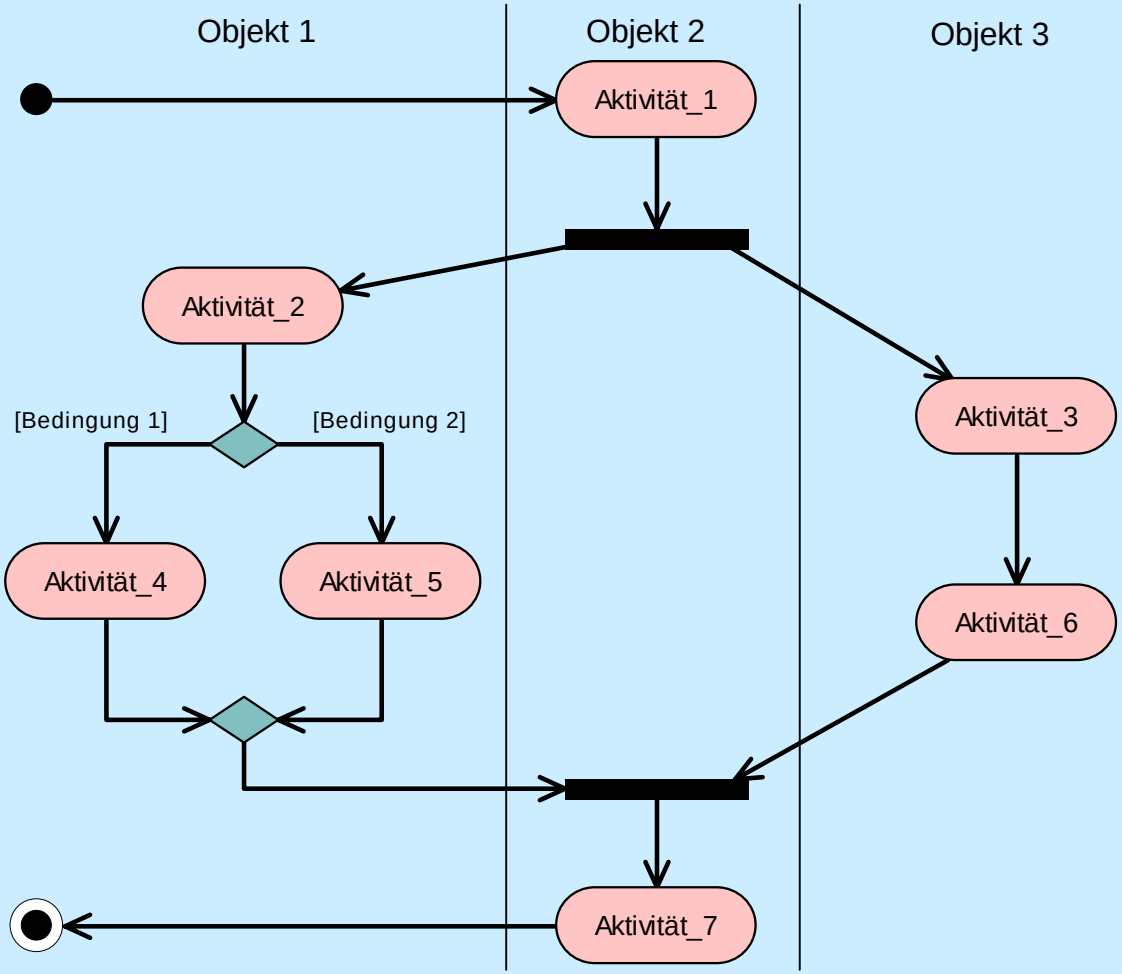


- Verantwortlichkeitsbereiche
- Festlegung von Zuständigkeiten
- mögliche Interpretation:
  - Zugehörigkeit von Aktivitäten zu Objekten
- Werkzeugunterstützung:
  - eingeschränkte Darstellungsmöglichkeiten
  - kreative Weiterentwicklungen

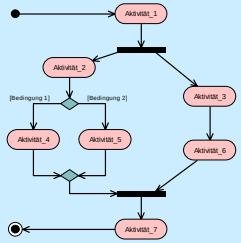
„Schwimmbahnen“ - 2



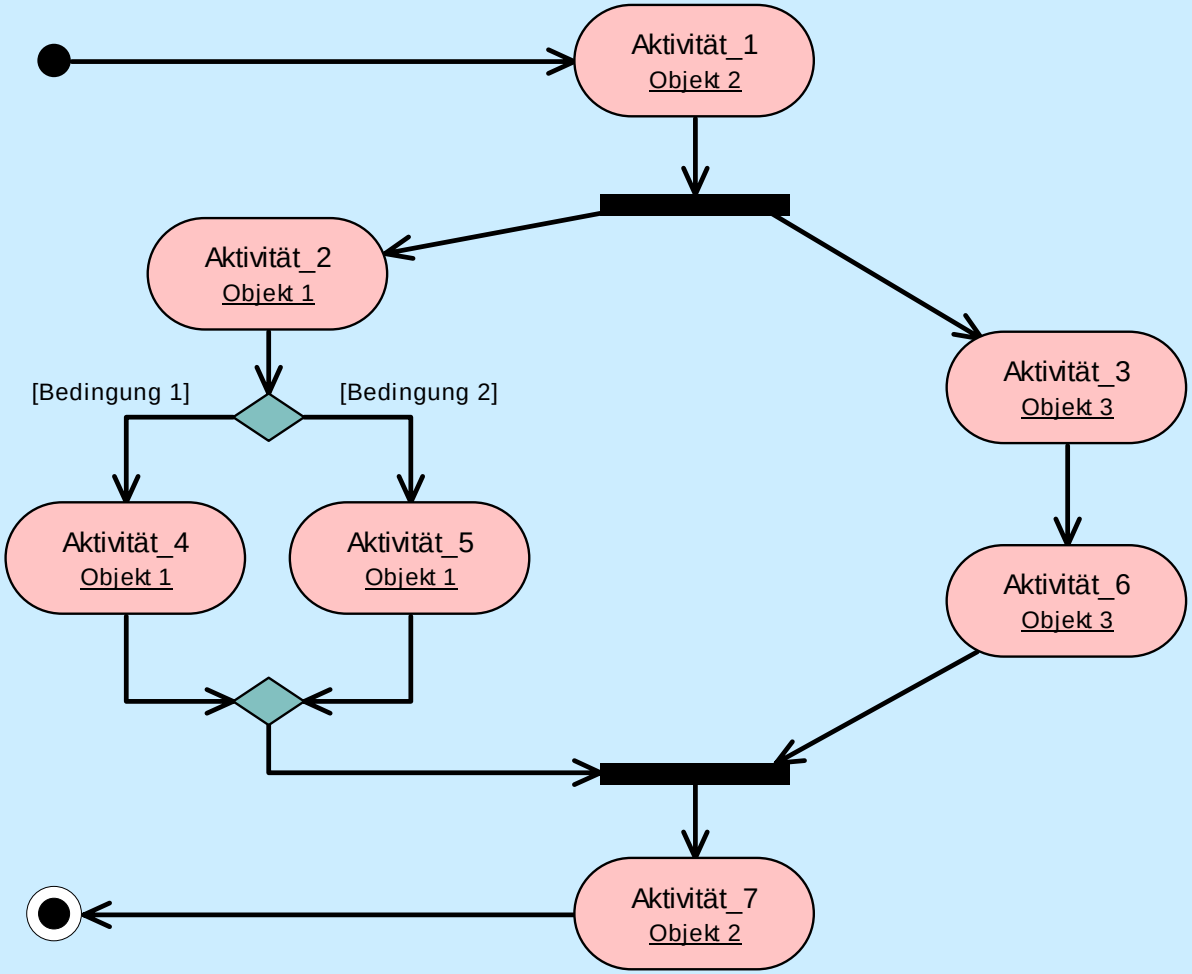
- Beispiel nach UML-Darstellung



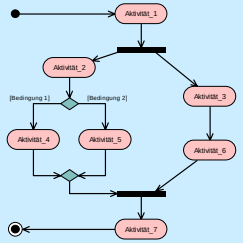
„Schwimmbahnen“ - 3



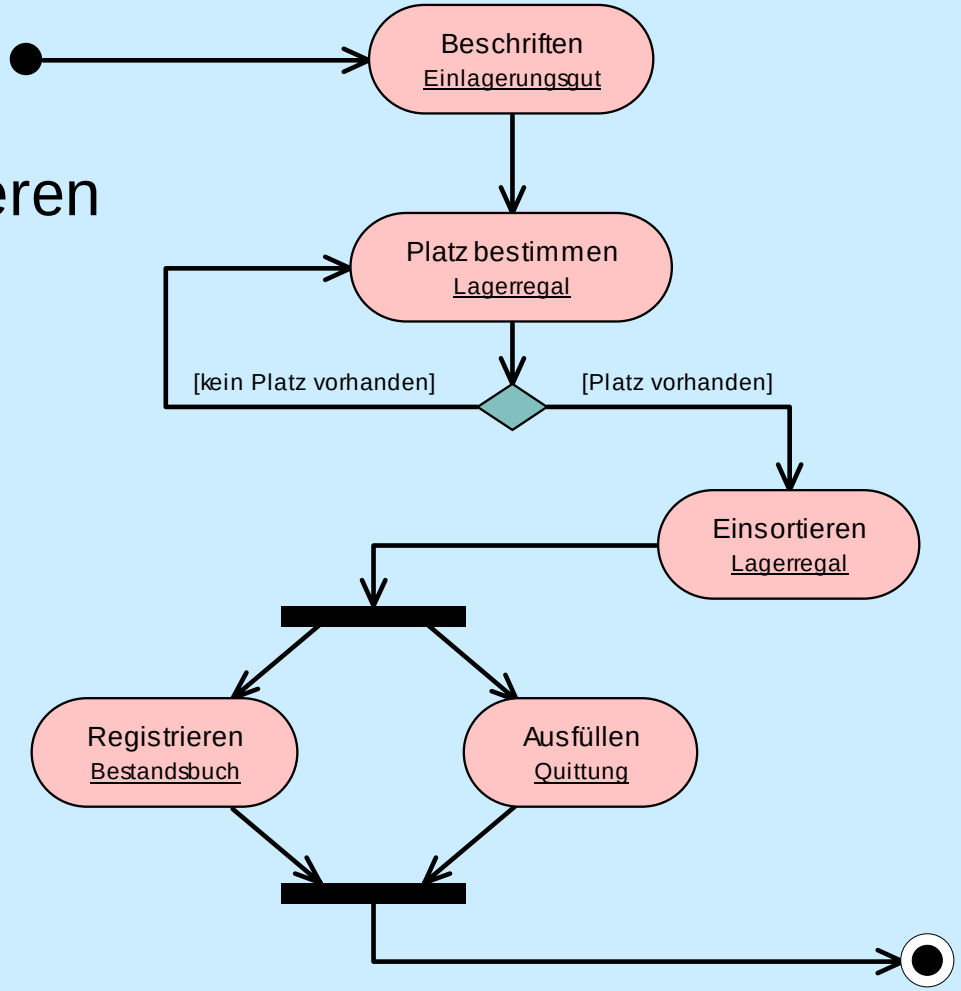
- Beispiel nach „weiter-entwickelter“ Darstel-lung



Beispiel



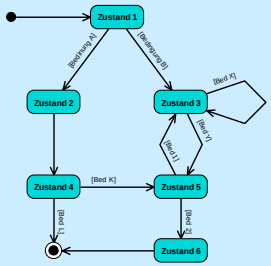
- Weinkeller
  - Wein einsortieren



Inhalt

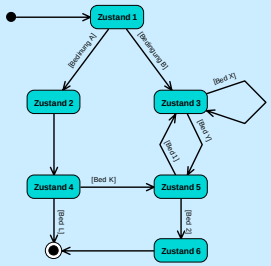
- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
    - Sequenzdiagramm
    - Kollaborationsdiagramm
    - Aktivitätsdiagramm
    - Zustandsdiagramm
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
- Modellverwertung

Zustandsdiagramm - 1



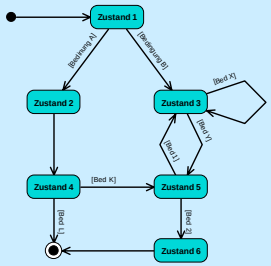
- Verhaltensbeschreibung
  - System
  - Klasse
  - Objekt
- Lebenszyklus
  - Erzeugung
  - Folge von Zuständen
  - Zerstörung

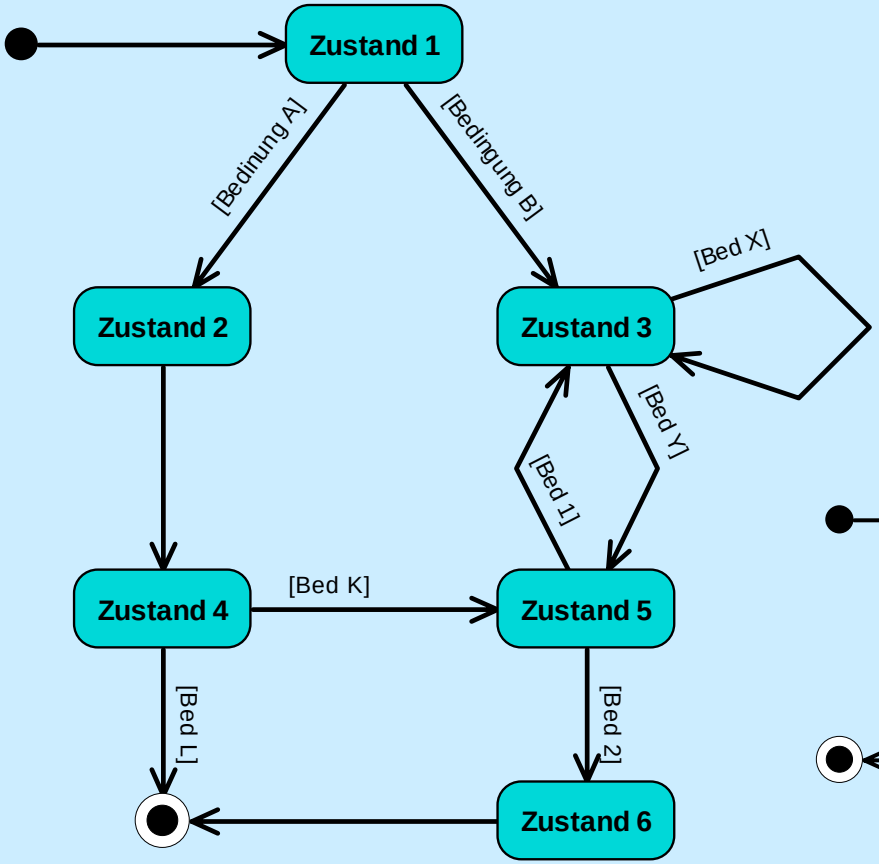
Zustandsdiagramm - 2



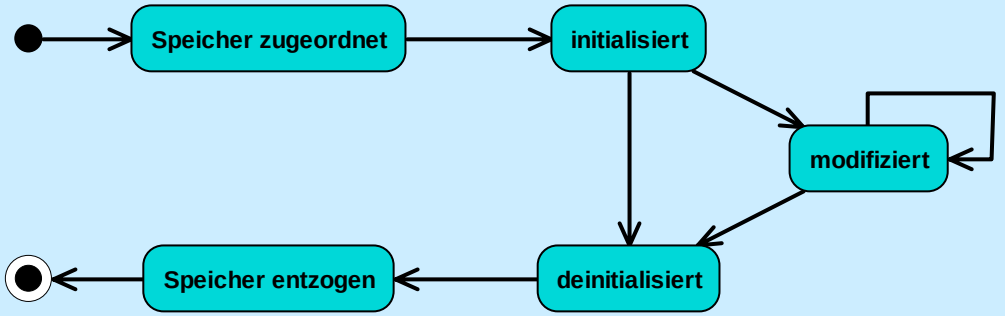
- Zustände eines Elements
  - Welche Zustände kann ein Element (Objekt, Klasse) annehmen?
- Folge von Zuständen
  - Wie gelangt man ausgehend von einem Zustand in einen anderen?
- Aktionen, die den Zustand verändern
  - Reaktion auf eingehende Ereignisse
  - Auslösen von Ereignissen

Graphische Darstellung

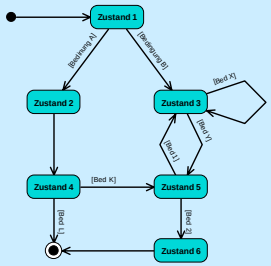




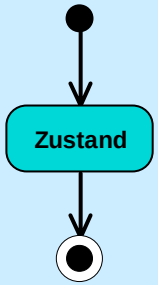
- Lebenszyklus eines Objektes

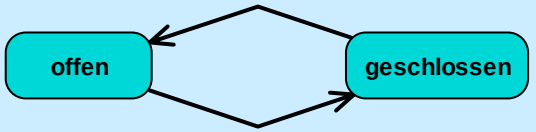


Zustand

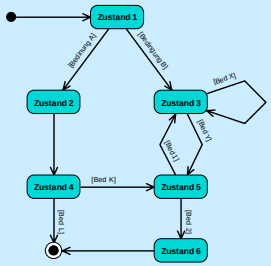


- Startzustand
- Endzustand
- Zustand
  - Charakterisierung einer Situation
    - Bedingung erfüllt
    - Ausführung einer Aktion
    - Warten auf ein Ereignis
  - z.B. Zusammenfassung einer Menge von Attributwerten
- Zustandsübergang



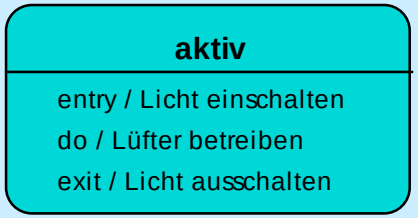


Ereignis - Aktion

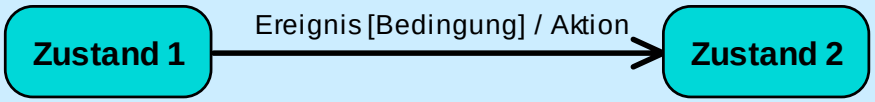


- Zustand
  - Ausführung von Aktionen im Zustand
  - Ereignis [Bedingung] / Aktion
    - entry
    - do
    - exit

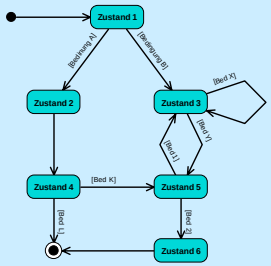




- Zustandsübergang
  - Ausführung von Aktionen während Zustandsübergangs
  - Ereignis [Bedingung] / Aktion

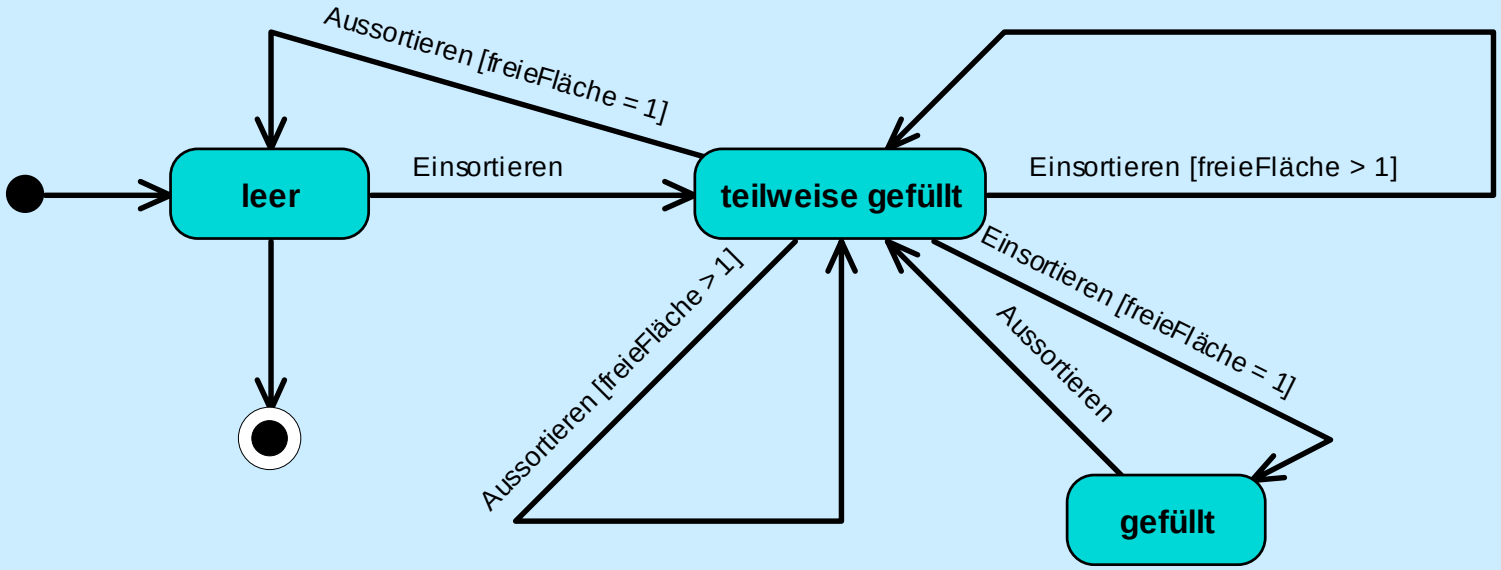


Beispiel



- Weinkeller

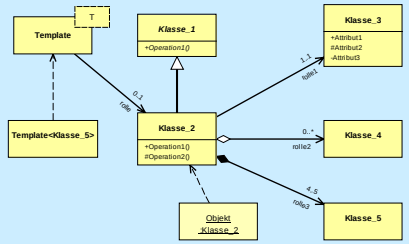
  Zustand des Lagerregals



Inhalt

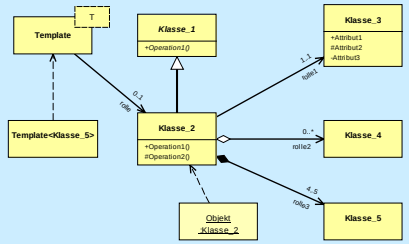
- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
- Modellverwertung

Strukturdiagramme



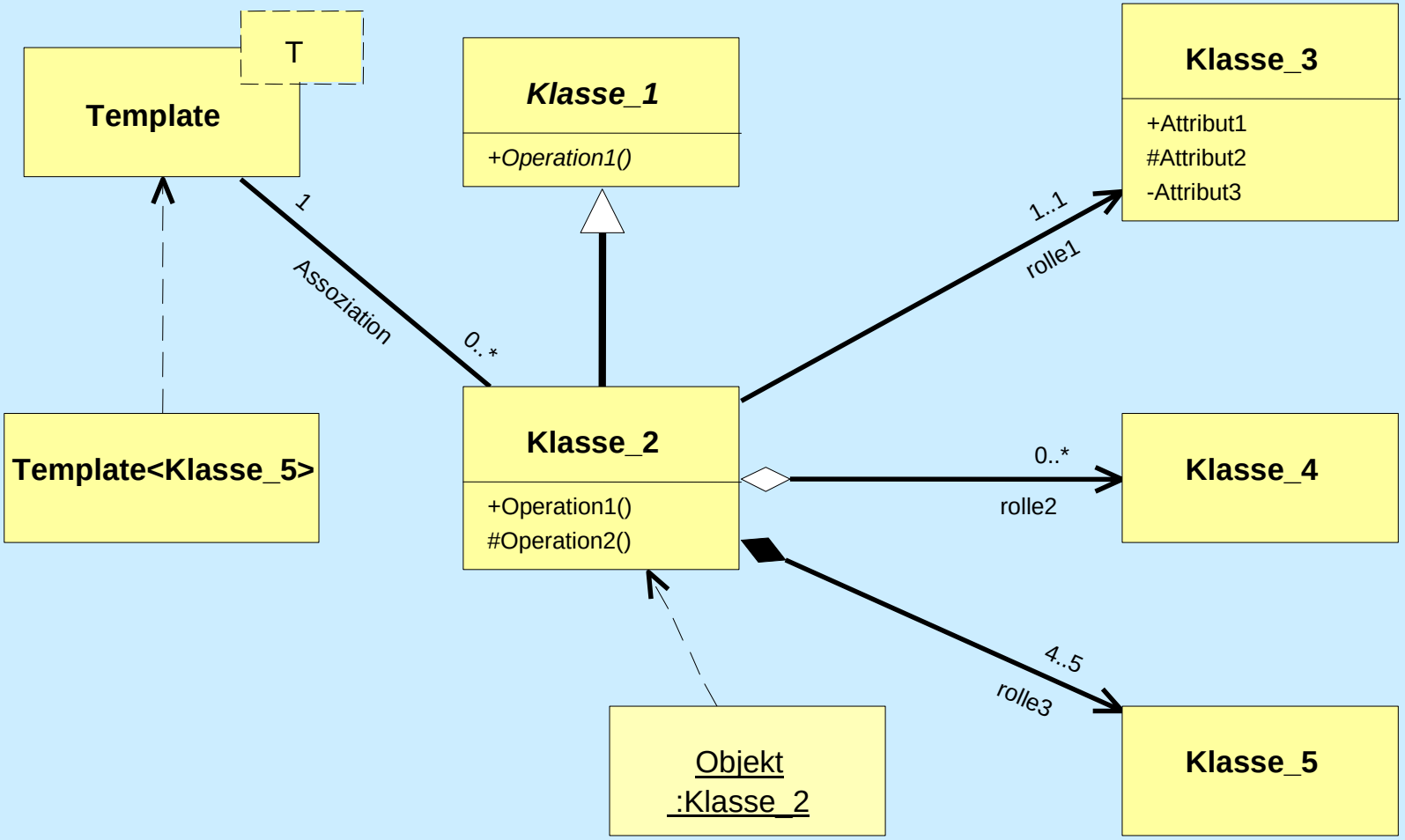
- Klassendiagramm
  - Statische Struktur
    - Existierende Elemente (Klassen, Typen)
    - Interne Struktur eines Systems
    - statische Beziehungen der Strukturelemente
  - keine zeitabhängigen Informationen
  - logische Architektur
  - Name
    - „Statisches Strukturdiagramm“
    - „Klassendiagramm“

Strukturdiagramme

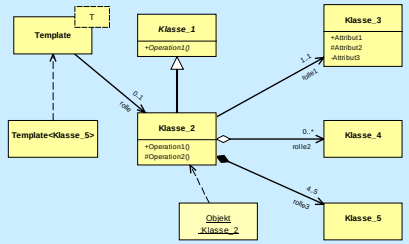


- Objektdiagramm
  - Betrachtung der Instanzen
    - Objekte
    - Datenwerte
  - Kompatibilität zur Klassenstruktur
  - „Schnappschuß“ zu einem Zeitpunkt
  - detaillierter Zustand eines Systems
  - „Static Object Diagram“
  - Empfehlung für Werkzeuge
    - Zusammenfassung mit Klassendiagrammen

Graphische Darstellung



Klasse

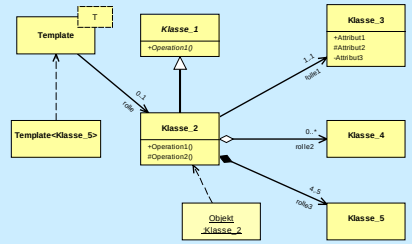


- Vorlage für Objekte
  - Struktur - Attribut
  - Verhalten - Operation/Methode
  - Beziehungen
- Datentyp
- Instanziierung von Objekten





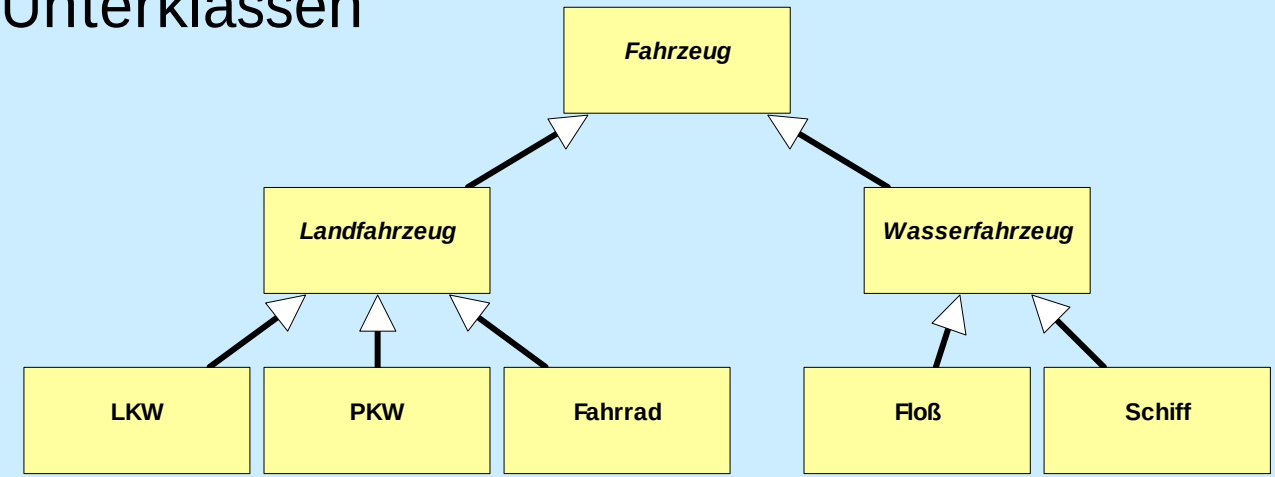
Abstrakte Klasse



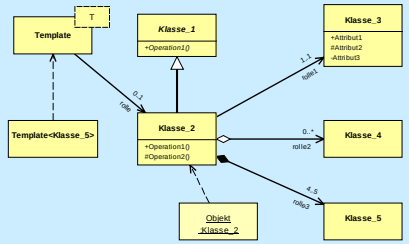
- Abstrakte Klasse
  - keine Objekte instanziierbar
  - Allgemeinbegriff, Verallgemeinerung
  - Definition gemeinsamer Eigenschaften
  - Basis für Unterklassen





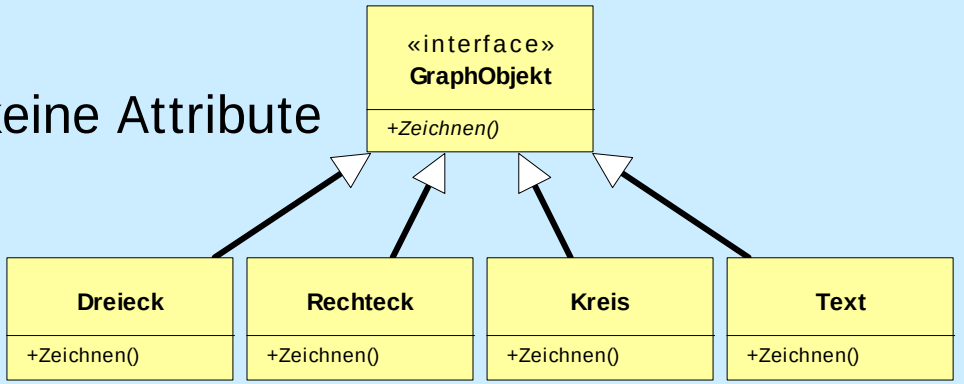


Schnittstelle

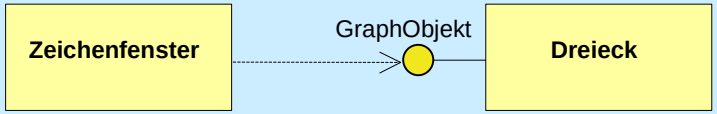


- Schnittstelle/Schnittstellenklasse/Interface
  - abstrakte Klasse (Typ)
  - extern sichtbares Verhalten
    - Signatur einer Operation
    - keine interne Struktur - keine Attribute
  - Stereotyp «interface»

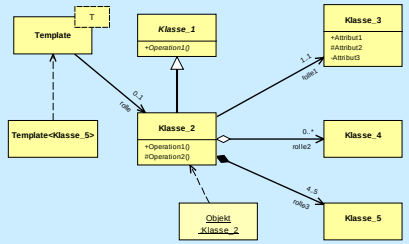


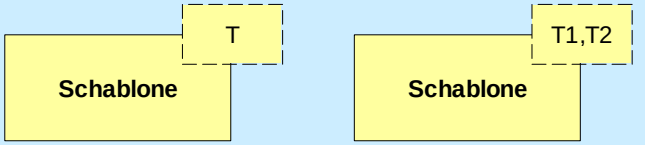


  - Einsatz
    - Erfüllung von Schnittstellen
    - Zugriff über Schnittstelle

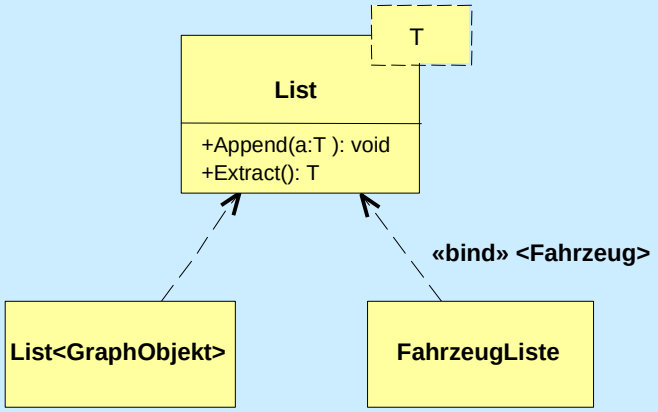


Parametrisierte Klasse

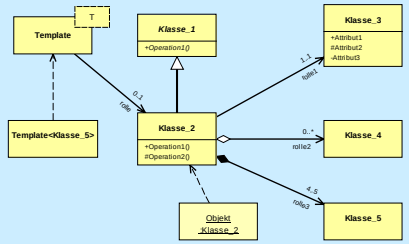




- Schablone/Template
- Vorlage zur Erzeugung einer konkreter Klassen
  - ungebundene Parameter
  - Wiederverwendung
- Familie von Klassen
  - Einsatz der Parameter für Attribute und in Operationen
- Erzeugung konkreter Klassen
  - Zuordnung von aktuellen Werten zu ungebundenen Parametern



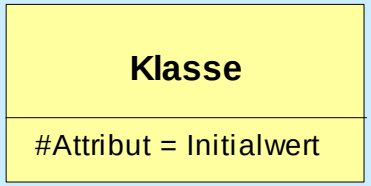
Attribut - 1

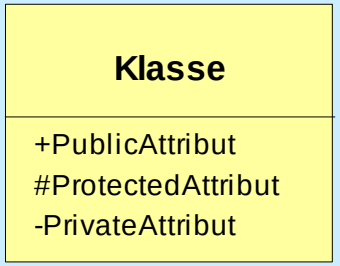


- Eigenschaftselement der Klasse - Datenelement
- in jedem Objekt der Klasse enthalten
- beschreibt Wert/Zustand

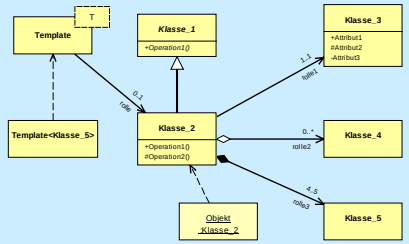


  - Initialwert
  - Sichtbarkeit des Attributs
    - public
    - protected
    - private





Attribut - 2



- Kardinalität/Multiplizität
  - Auftretenshäufigkeit
- Klassenattribut
  - gleicher Wert für alle Instanzen
- abgeleitete Attribute
  - Berechnung
  - basierend auf anderem Attribut

Klasse

#Attribut[4..5]:AttributKlasse

Klasse

#StatischesAttribut:AttributKlasse

Klasse

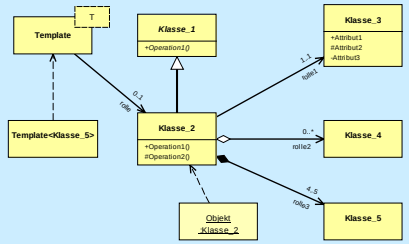
#/AbgeleitetesAttribut:AttributKlasse

- Beispiel

Fahrrad

+Vorderrad:Rad
+Hinterrad:Rad
+/Wert:EinheitGeld
#MinGroesse:EinheitZoll = 20
#MaxGroesse:EinheitZoll = 28
-Groesse:EinheitZoll = 26

Operation/Methode - 1

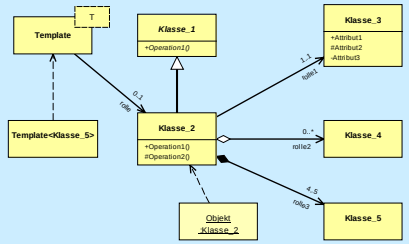


- Klasse zugeordnete Funktionalität
  - Dienstangebot der Klasse
  - Operation -> Methode
- In Objekten der Klasse verfügbar
- Zugriff und Veränderung der Attribute
  - Sichtbarkeit der Methode
    - public
    - protected
    - private





Operation/Methode - 2



- Abstrakte Operationen
  - Operation, die überschreiben wird
- Klassenoperation
  - operiert nicht auf Objekt, sondern auf der Klasse
- Return-Wert
  - Typ des Ergebnisses der Operation
- Parameter
  - Name, Typ, Standardwert
- Beispiel

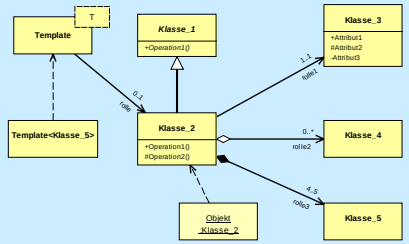
Klasse
+AbstrakteOperation()

Klasse
+KlassenOperation()

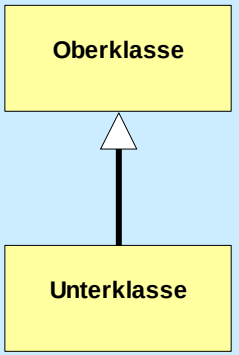
Klasse
+Operation(p1:K1= initialP1, p2:K2 =initialP2): ReturnK

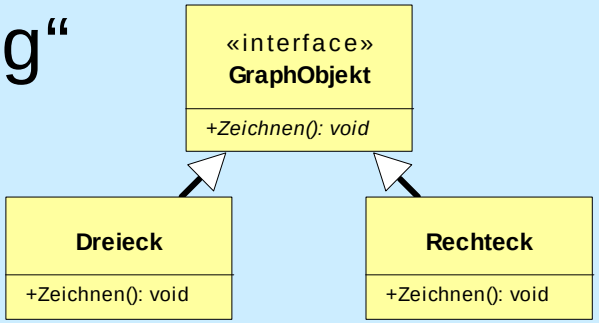
Fahrrad
+Create(gr:EinheitZoll): Fahrrad +getGroesse(): EinheitZoll -setGroesse(other:EinheitZoll): bool

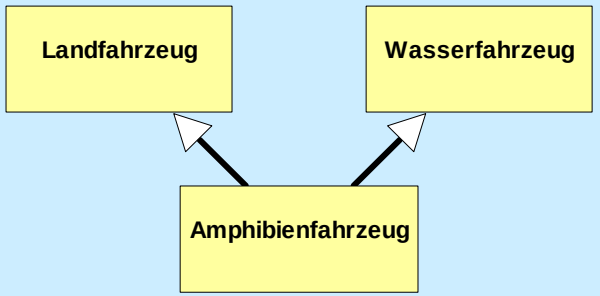
Vererbung - 1



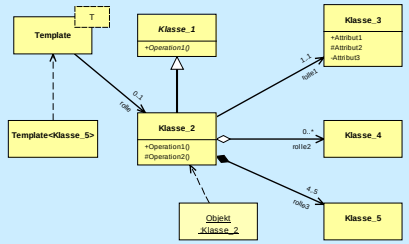
- Beziehung zwischen allgemeinem und speziellem Element
  - Spezialisierung - Generalisierung
  - Erweiterung des Dienstangebotes
  - Kombination
- Übernahme von Attributen und Operationen
- „Is-A-Beziehung“



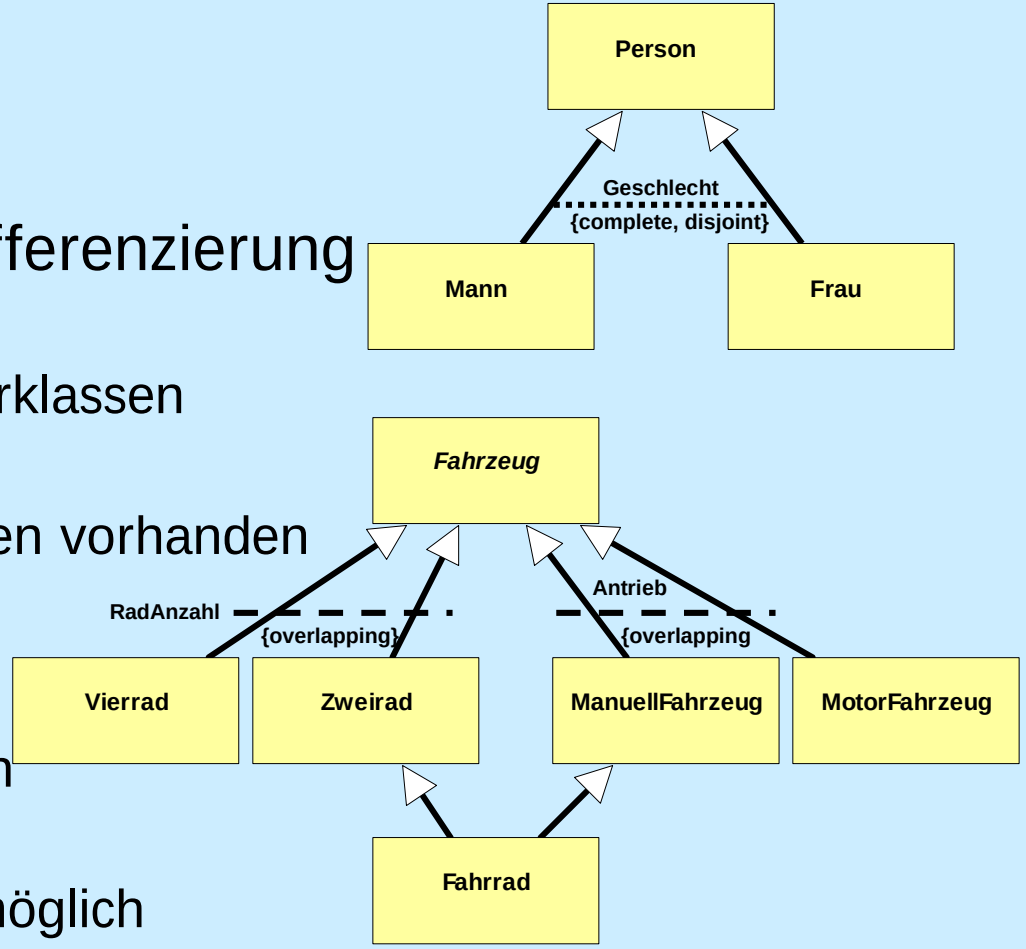




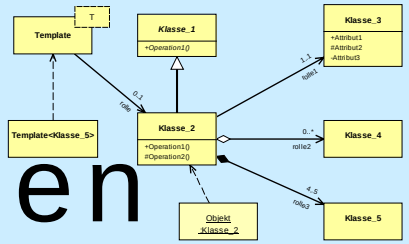
Vererbung - 2



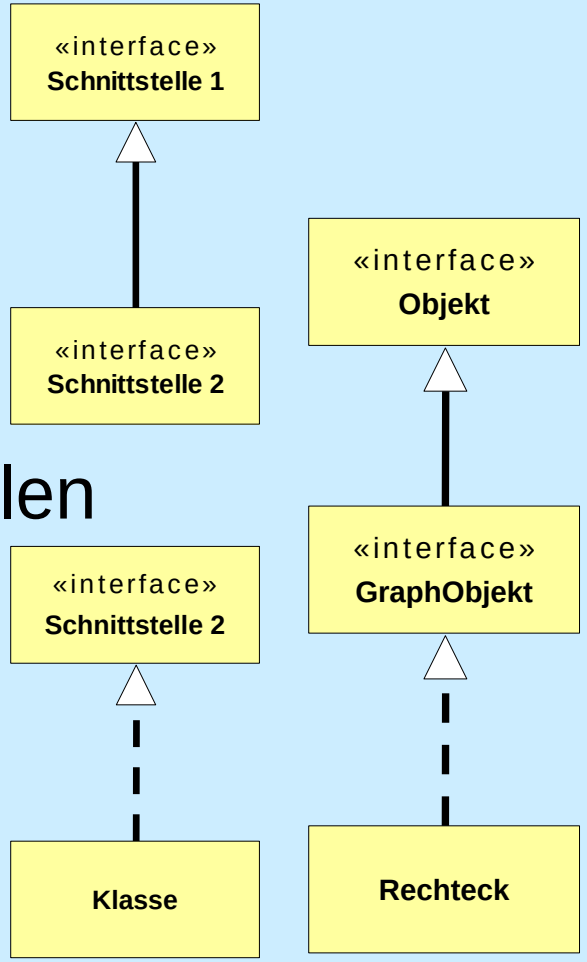
- Diskriminatoren
  - Absicht
  - Vollständigkeit der Differenzierung
    - *complete*
      ➥keine weiteren Unterklassen
    - *incomplete*
      ➥weiteren Unterklassen vorhanden
  - Kombinierbarkeit
    - *overlapping*
      ➥Kombination möglich
    - *disjoint*
      ➥Kombination nicht möglich



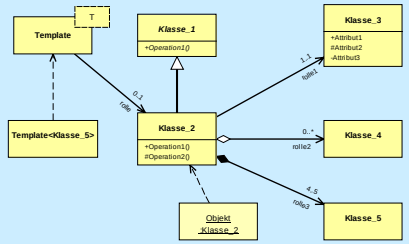
Vererbung - Schnittstellen



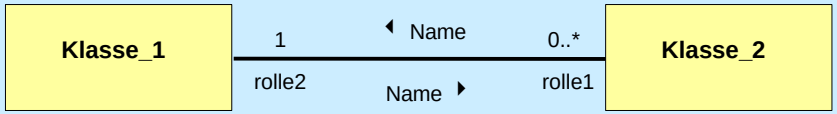
- Vererbung von Schnittstellen
  - Übernahme von Eigenschaften
  - Definition einer neuer Schnittstelle
- Implementierung von Schnittstellen
  - Erfüllung von Schnittstellen
  - Implementierung der Methoden

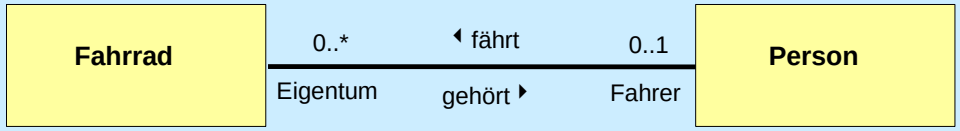


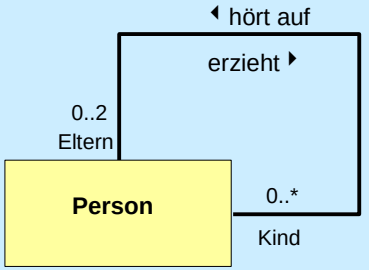
Assoziation - 1



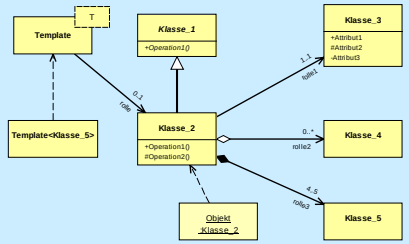
- Relation zwischen Klassen
- gemeinsame Struktur und Semantik einer Menge von Objektverbindungen
- „Has-A-Beziehung“



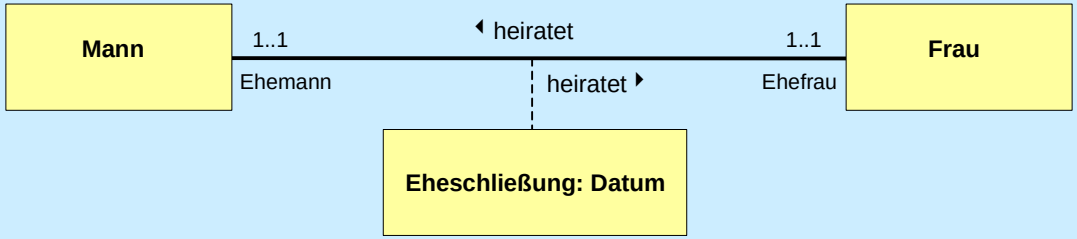




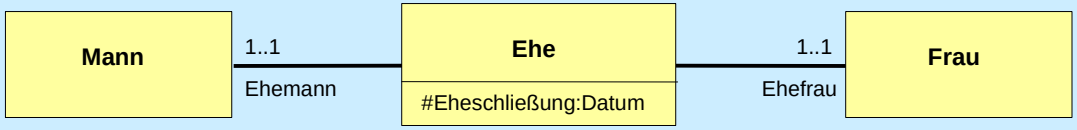
Assoziation - 2



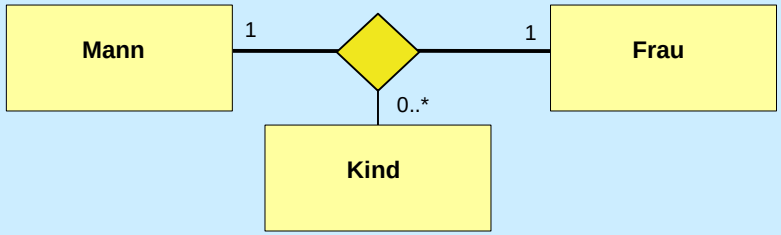
- Assoziationsattribute



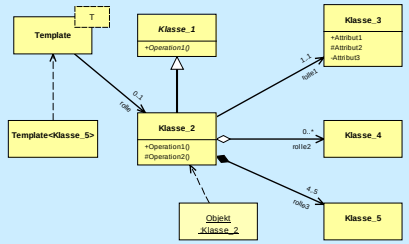
- Assoziationsklasse



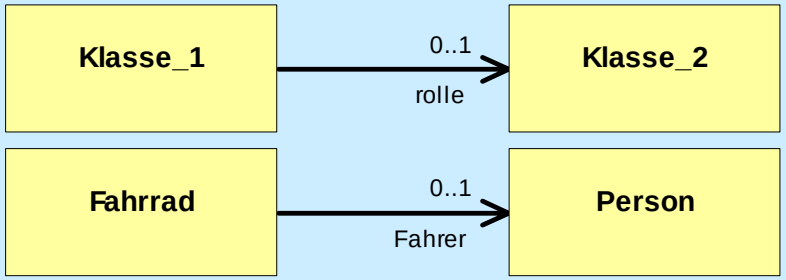
- Mehrgliedrige Assoziation



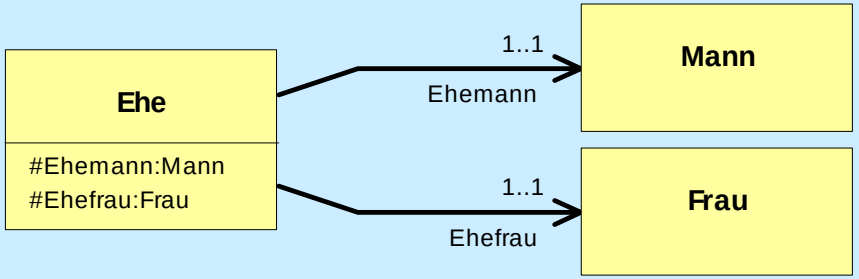
Assoziation - 3



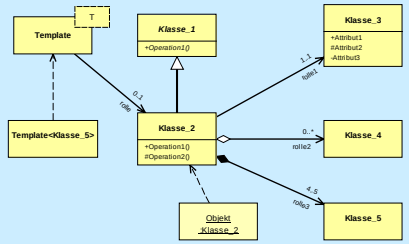
- Gerichtete Assoziation
  - Navigierbarkeit
- Rolle
  - 0 oder 1 Person spielt die Rolle Fahrer für das Fahrrad



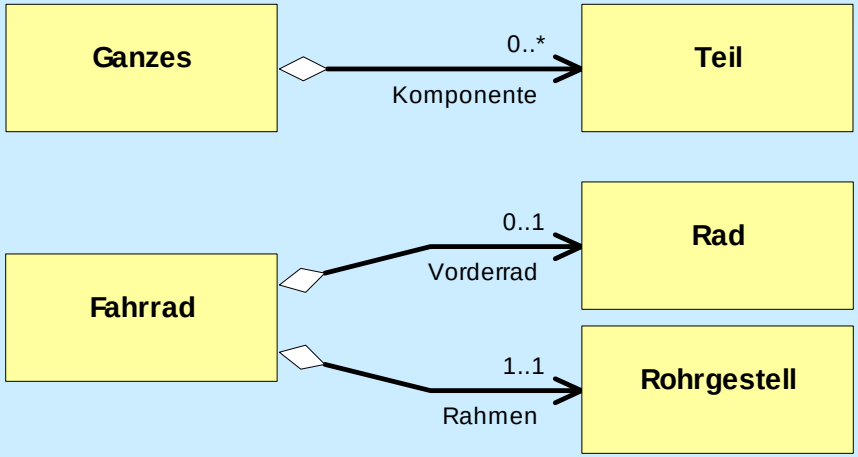
- Umsetzung als Attribut
  - Klasse
  - Datentyp
  - Kardinalität



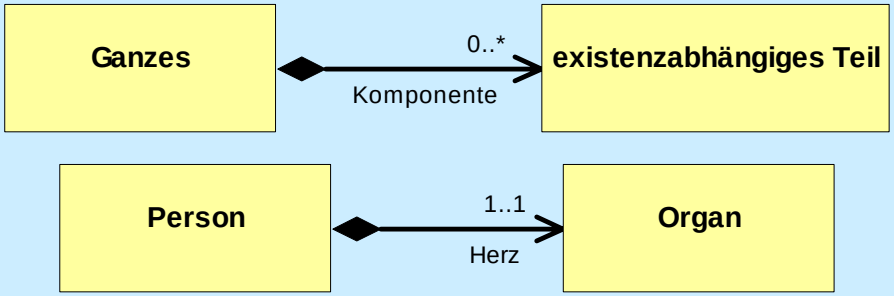
Assoziation - 4



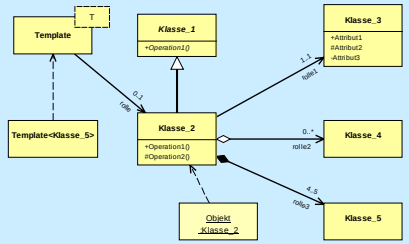
- Aggregation
  - Ganzes-Teile-Hierarchie
  - Beschreibung der Zusammengehörigkeit



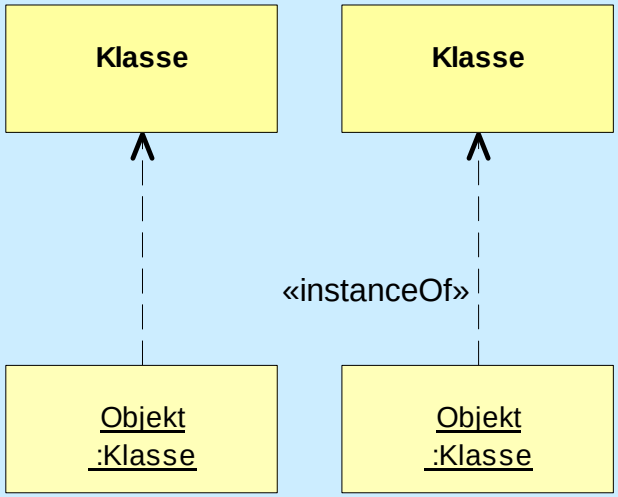
- Komposition
  - strenge Form der Aggregation
  - Existenzabhängigkeit



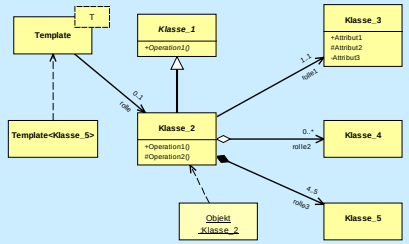
Instanziierung



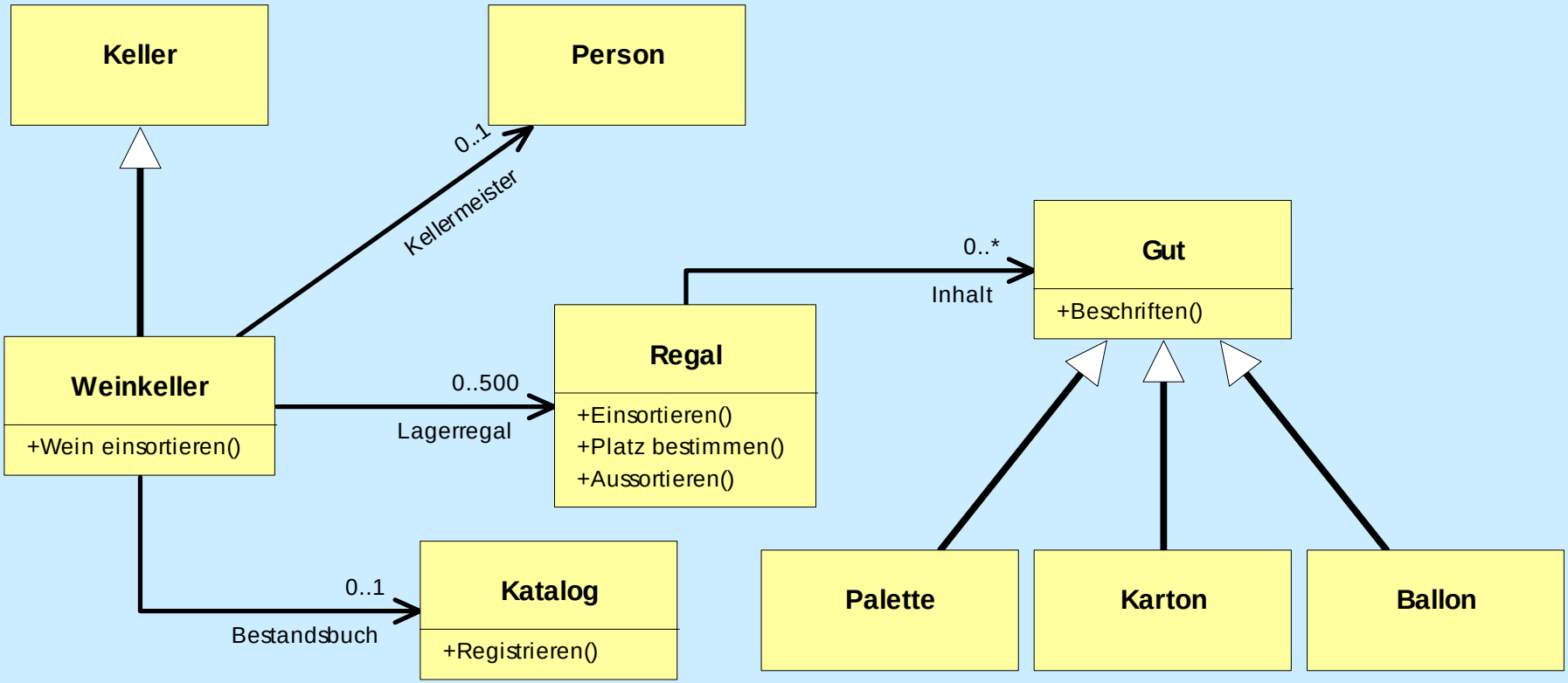
- Beziehung Klasse-Objekt
- Objekt ist Instanz einer Klasse
- Klasse
  - Attribute
  - Methoden
- Objekt
  - Attributwerte (alle Attribute einer Klasse)
  - Nachrichten -> Methoden/Operationen



Beispiel



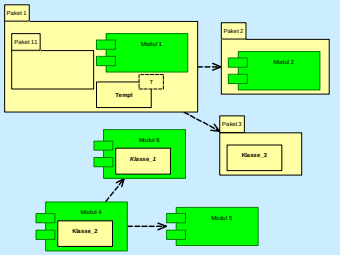
Weinkeller



Inhalt

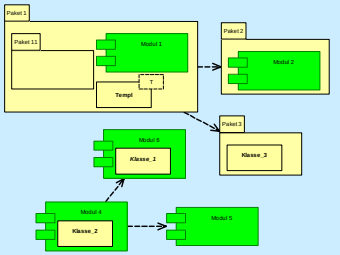
- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
    - Komponentendiagramm
    - Verteilungsdiagramm
- Modellverwertung

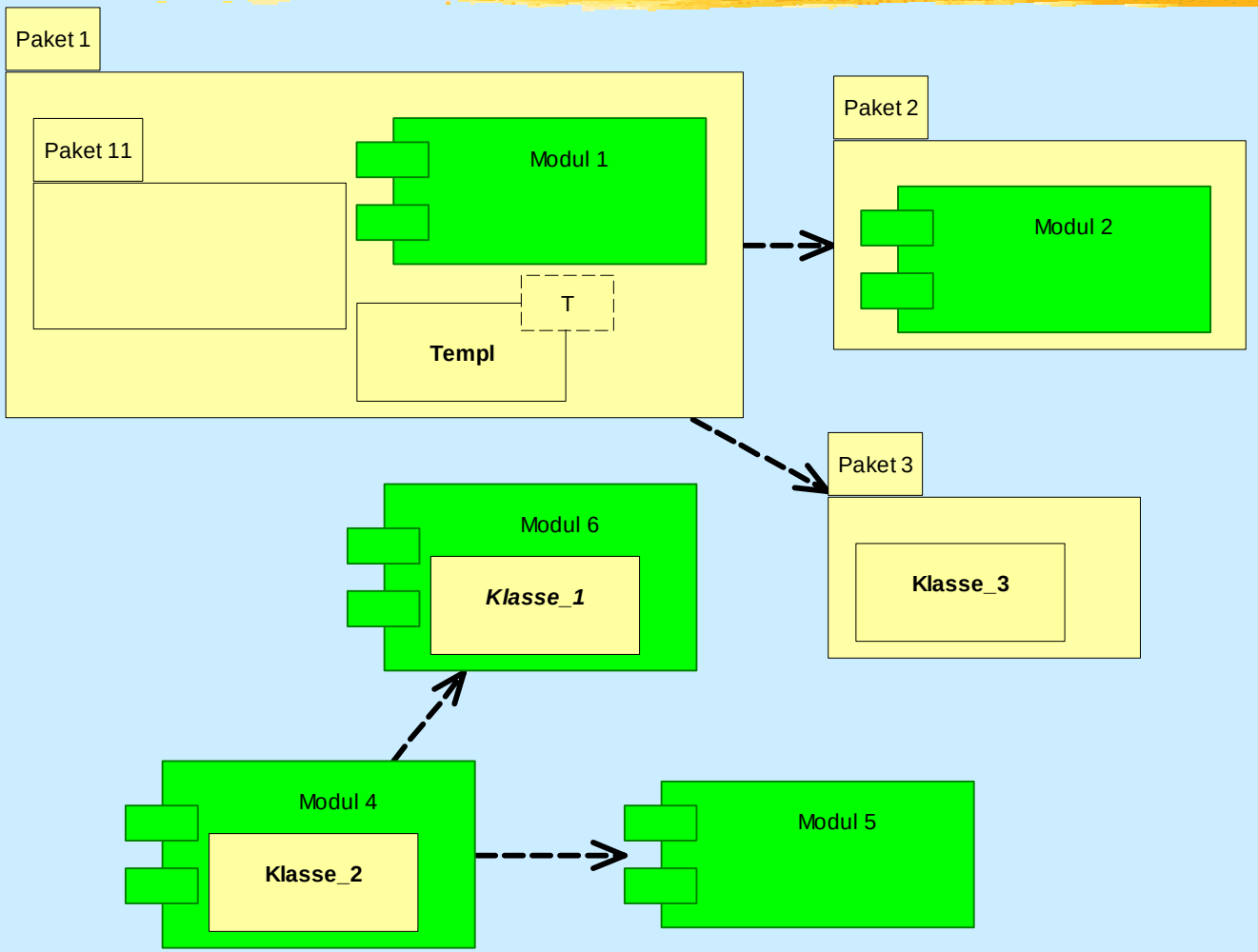
Komponentendiagramm



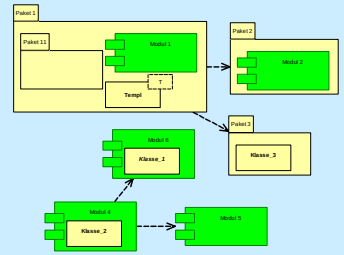
- Physische Architektur
  - Modularisierung eines Systems
- Softwarekomponenten
  - Quelltext
  - Binärcode
  - Ausführbare Komponenten
- Beziehungen der Komponenten
  - Abhängigkeiten (Compile-Zeit, Link-Zeit, Laufzeit)
- Typen, keinen Instanzen

Graphische Darstellung

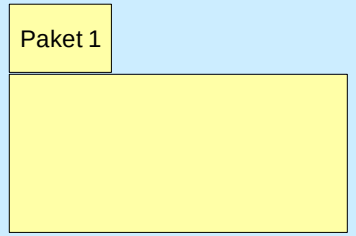


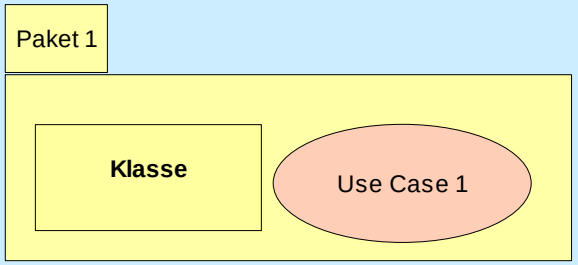


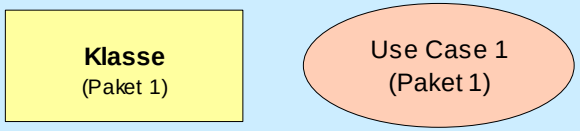
Komponente - Paket - 1

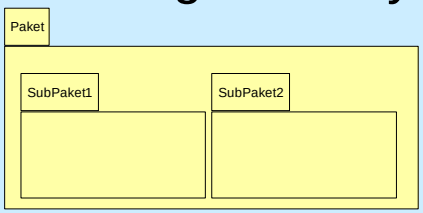


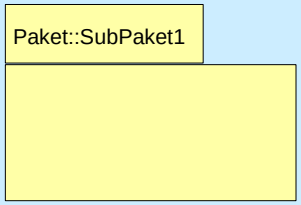
- Strukturierungsmittel für Modelle
- Zusammenfassung von Bestandteilen
  - z.B. Module, Klassen, Use Cases
- Zugehörigkeit zu Paketen
  - Kennzeichnung im Modellelement
  - Einsatz auch in anderen Diagrammen
- Hierarchie von Paketen
  - Aufteilung eines Systems im Subsysteme

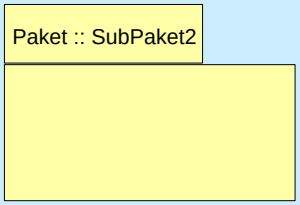




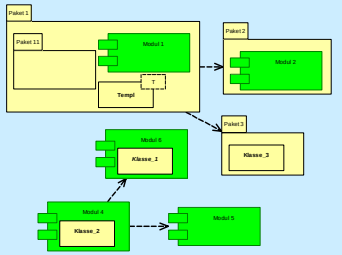




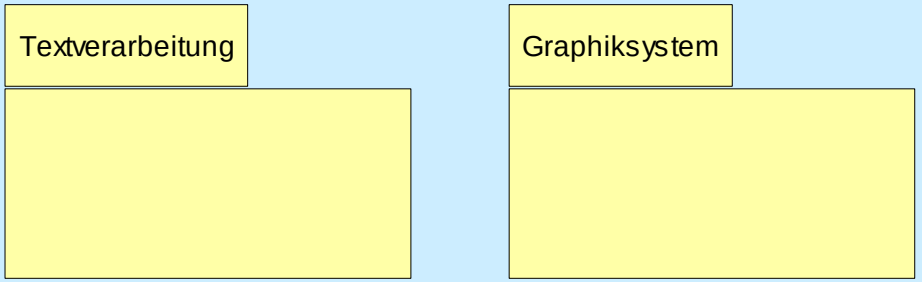




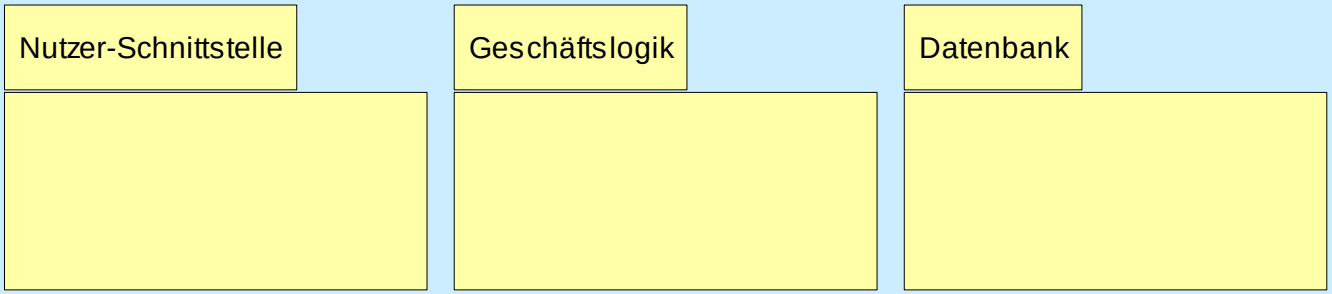
Komponente - Paket - 2



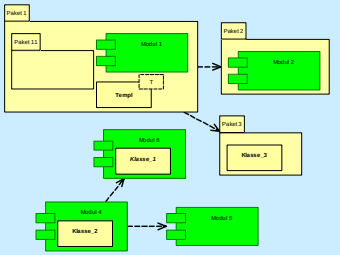
- Einsatz von Paketen
  - Ausführbare Programme
  - z.B. .exe



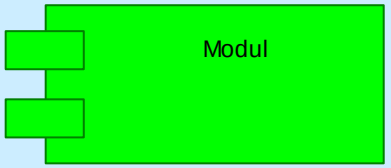
  - Bibliotheken
  - z.B. .dll, .lib



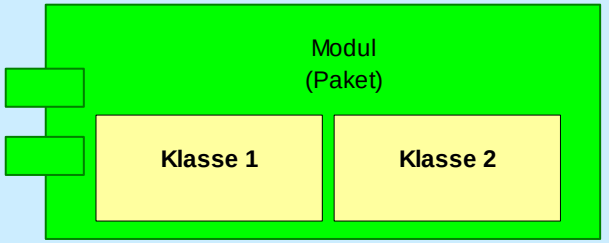
Komponente - Modul - 1



- Interpretationen
  - Implementierungseinheit eines Systems
  - Ausführbares Softwaremodul



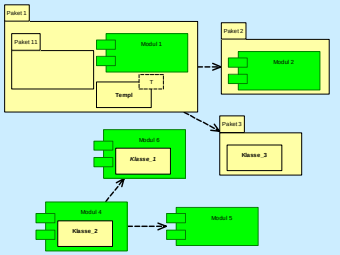
- Zusammenfassung von Bestandteilen
  - z.B. Klassen



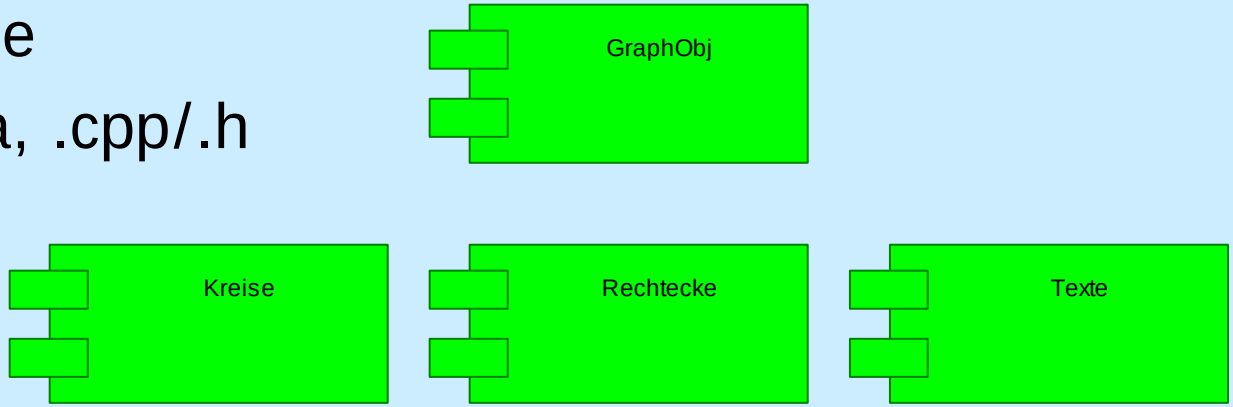
- Erfüllung von Schnittstellen
  - Dienstangebot



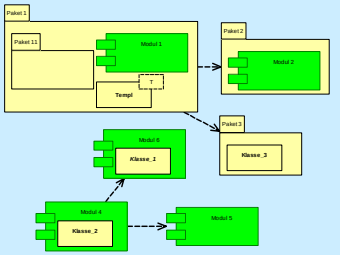
Komponente - Modul - 2



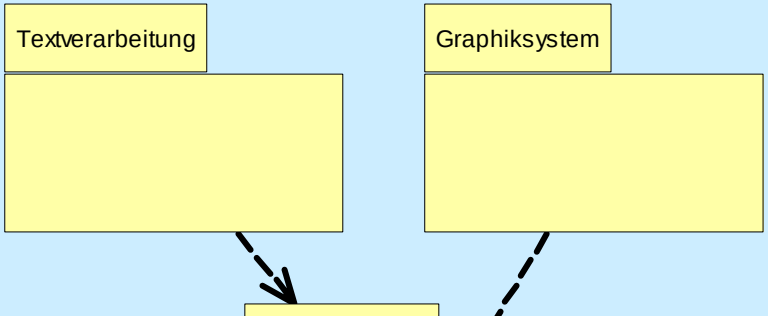
- Einsatz
  - Quelltexte
  - z.B. .java, .cpp/.h



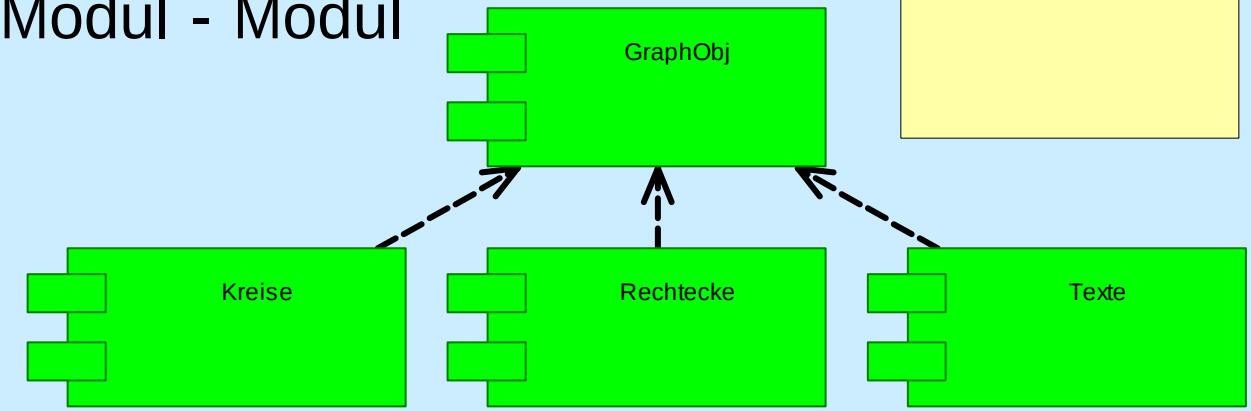
Beziehungen



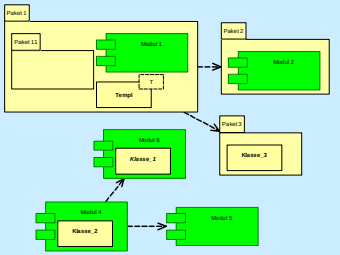
- Abhängigkeiten (Benutzt-Beziehungen)
  - Paket - Paket



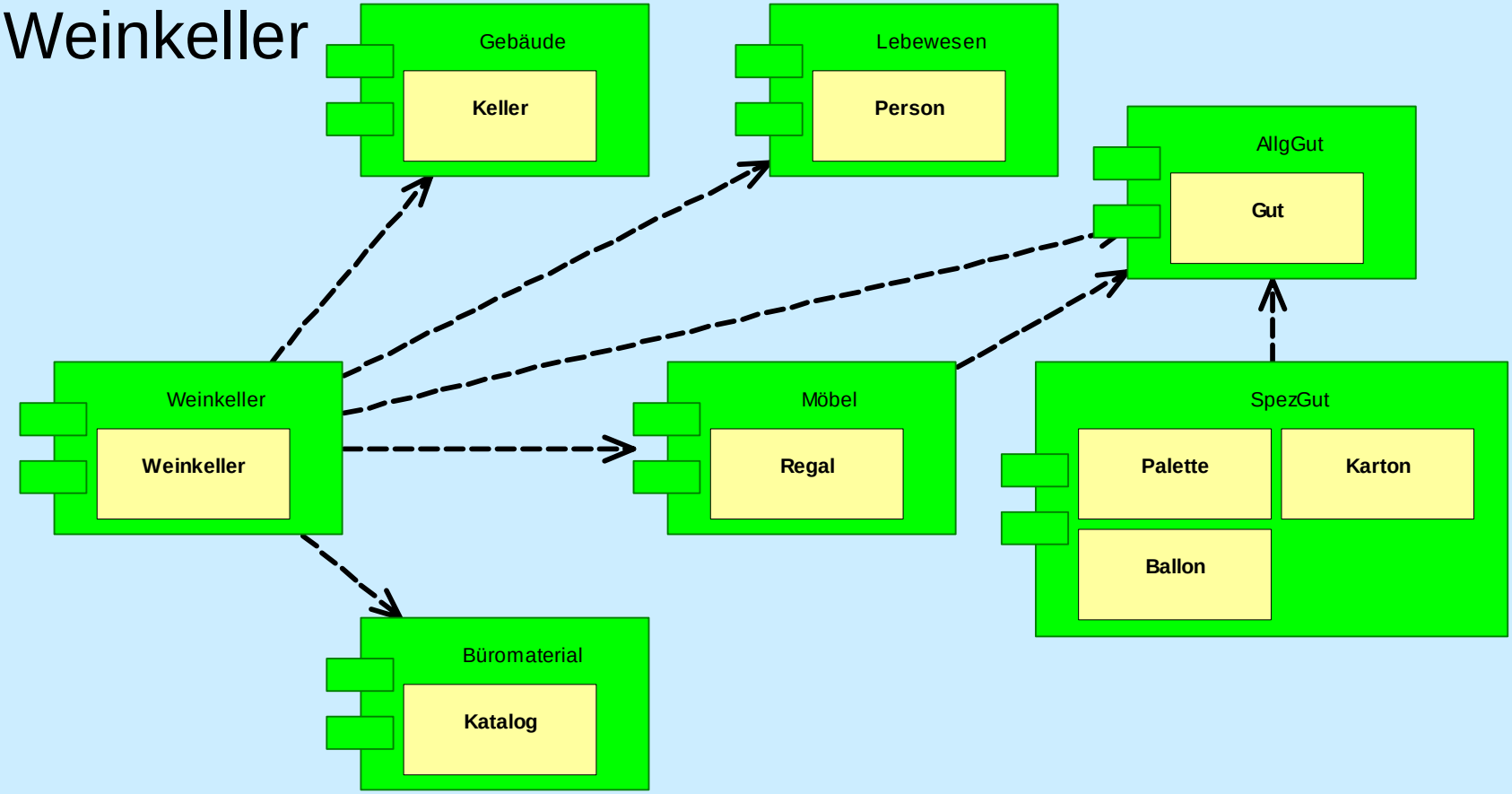
  - Modul - Modul



Beispiel



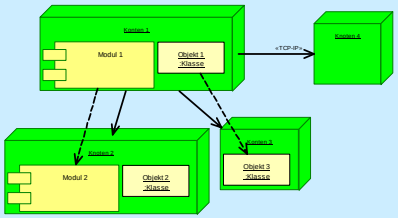
- Weinkeller



Inhalt

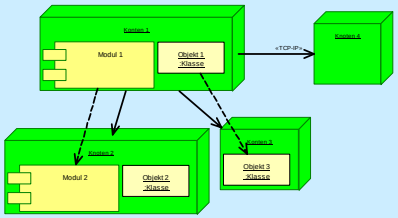
- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
    - Komponentendiagramm
    - Verteilungsdiagramm
- Modellverwertung

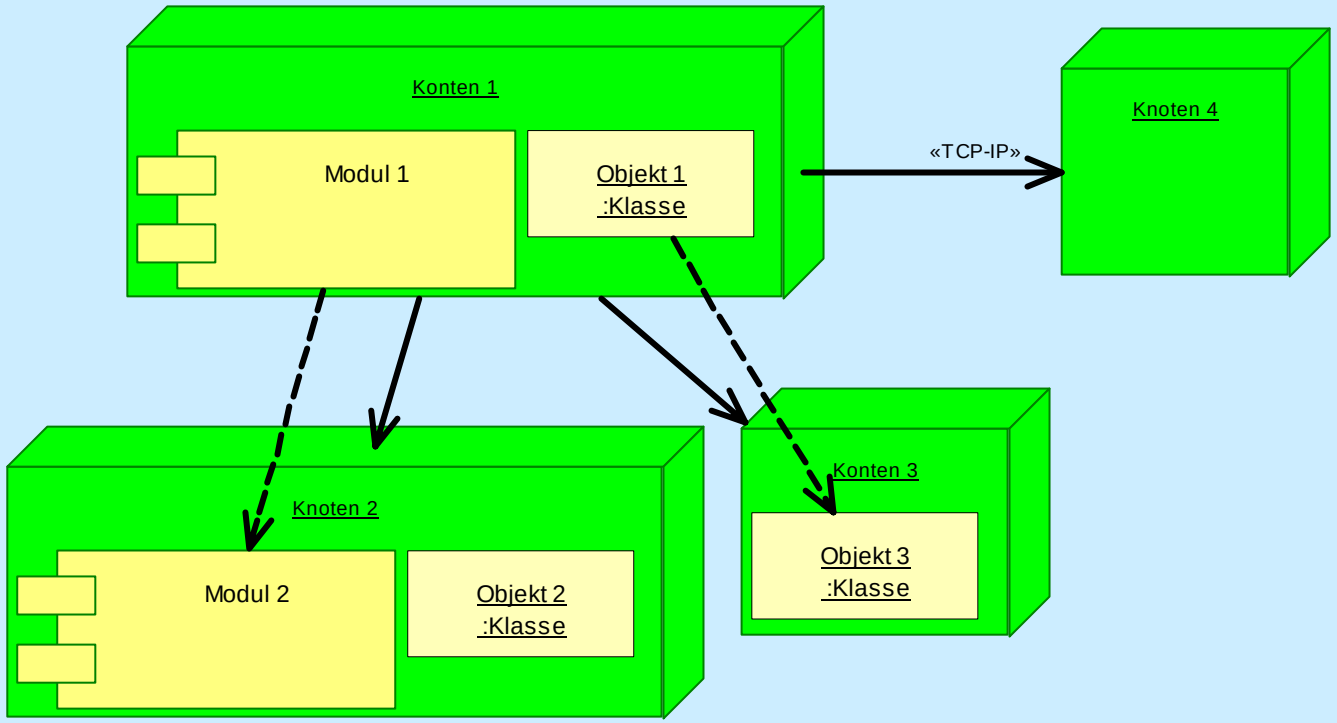
Verteilungsdiagramm



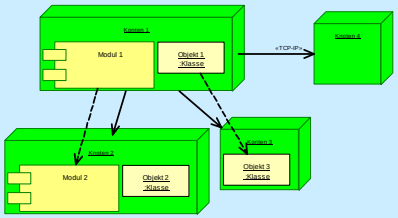
- Physische Architektur
  - Laufzeitumgebung (Software, Hardware)
  - Verteilung
- Laufzeitelemente
  - Softwarekomponenten
  - Prozesse
  - Objekte
- Konfiguration, Abhängigkeiten
- Instanzen, keine Typen

Graphische Darstellung

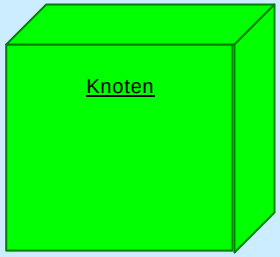




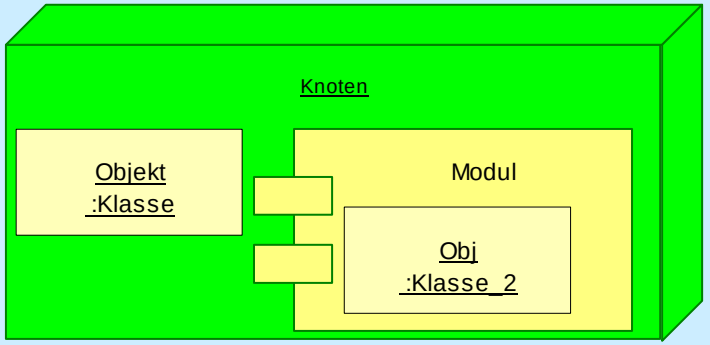
Knoten



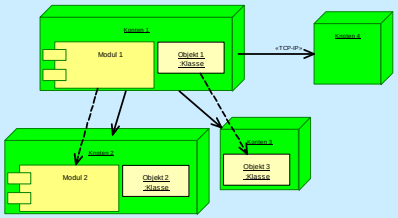
- Physisches Objekt zur Laufzeit
- Verarbeitungseinheit
  - Speicherkapazität
  - Rechenleistung (in der Regel)



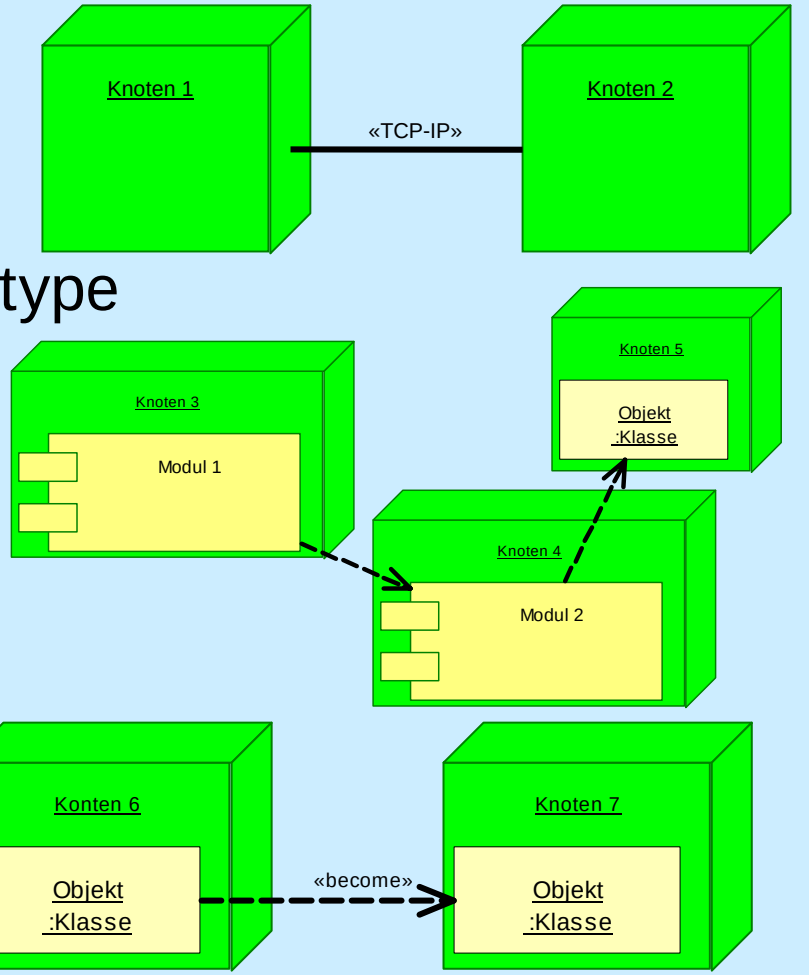
- eingebettete Elemente
  - Objekte
  - Module - Komponenten



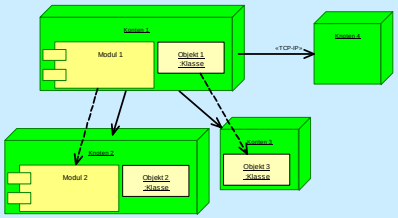
Beziehungen



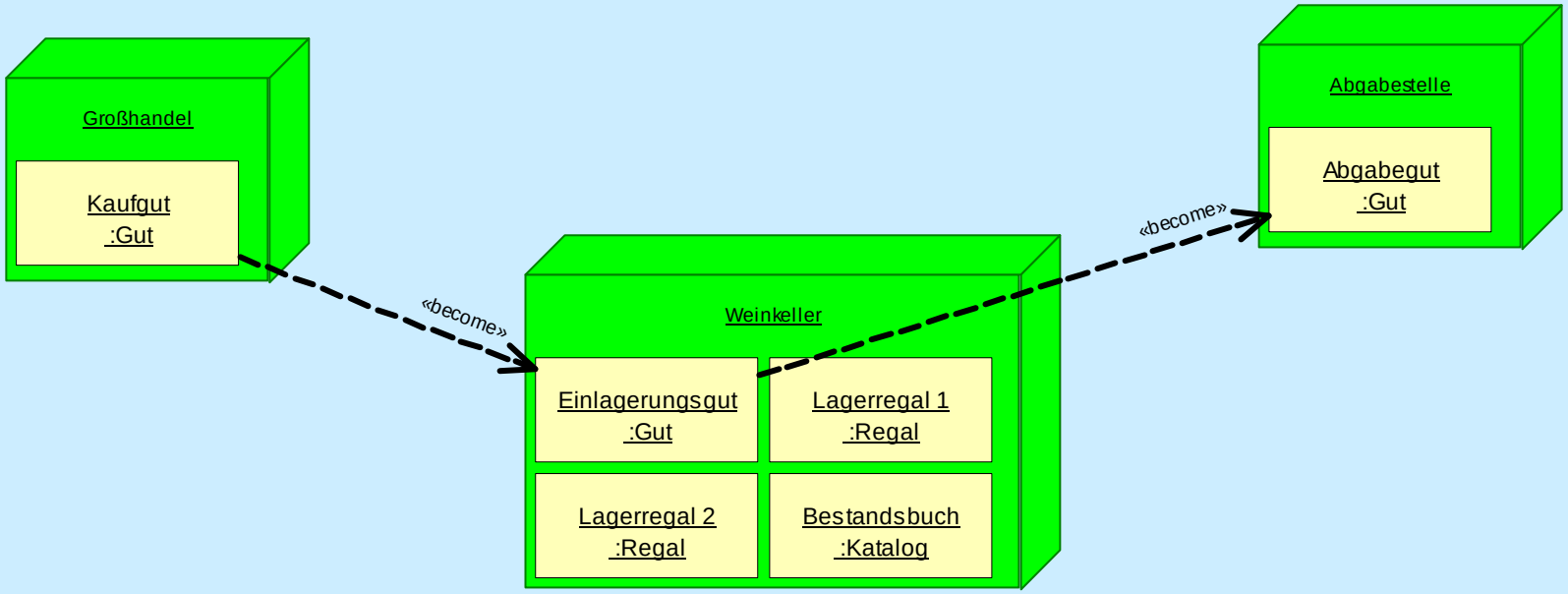
- Assoziationen
  - Kommunikationspfad
  - Charakterisierung: Stereotype
- Nutzung/Zugriff
  - Objekt
  - Modul (über Schnittstelle)
- Verteilung
  - Migration: «become»
  - Replikation: «copy»



Beispiel



- Weinkeller



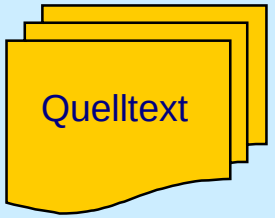
Inhalt

- Einleitung
- Struktur objektorientierter Modelle
- Diagrammtypen der UML
  - Use-Case-Diagramme
  - Verhaltensdiagramme
  - Strukturdiagramme
  - Architekturdiagramme/Implementationsdiagramme
- Modellverwertung

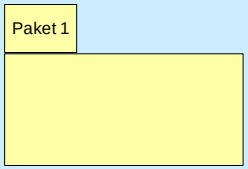
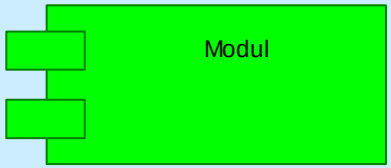
Modellverwertung

- Zweck der Modellierung
  - Dokumentation
  - Entwurfsunterstützung - Quelltextgenerierung
  - Wiederverwendung
  - Modellüberprüfung

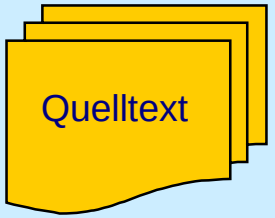
Quelltextgenerierung - 1

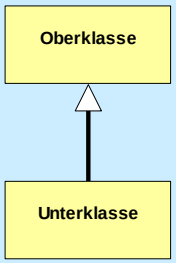
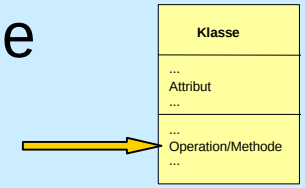
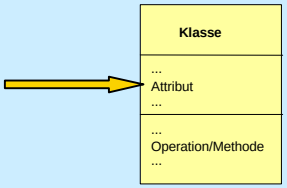


- Interpretationsmöglichkeiten

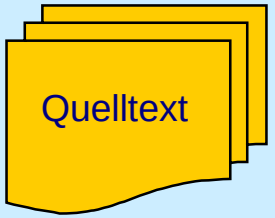
UML-Element	C++-Quelltext	Java-Quelltext
Paket 	Namensraum/ Verzeichnis	Package
Modul 	Deklarations- und Implementierungs-datei (.h- und .cpp)	Java-Datei (.java)
Klasse 	Klasse	Klasse

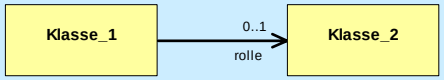
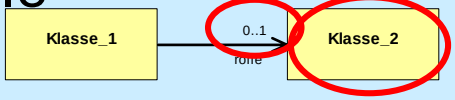
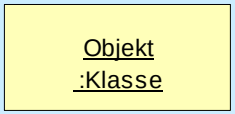
Quelltextgenerierung - 2



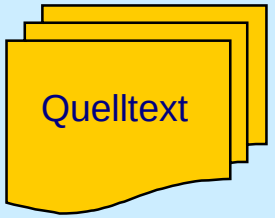
UML-Element	C++-Quelltext	Java-Quelltext
Vererbung 	Vererbung	Vererbung: «extends» oder «implements» (je nach Basisklasse)
Methode 	Methode einer Klasse	Methode einer Klasse
Attribut 	Attribut einer Klasse	Attribut einer Klasse

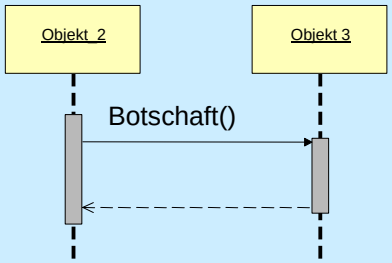
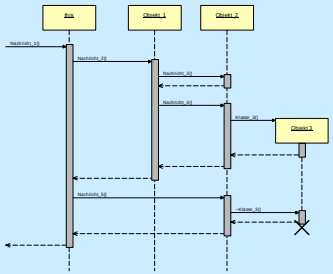
Quelltextgenerierung - 3



UML-Element	C++-Quelltext	Java-Quelltext
Rolle 	Attribut einer Klasse	Attribut einer Klasse
Kardinalität und Rollenklasse der Rolle 	Datentyp des Attributs	Datentyp des Attributs
Objekt 	Globales, lokales, Parameterobjekt	Globales, lokales, Parameterobjekt

Quelltextgenerierung - 4



UML-Element	C++-Quelltext	Java-Quelltext
Botschaft 	Aufruf einer Methode	Aufruf einer Methode
Sequenzdiagramm 	Implementierung einer Methode	Implementierung einer Methode

Zusammenfassung

- Überblick Unified Modeling Language (UML)
- Strukturierung objektorientierter Modelle
- Diagrammtypen
  - Zweck
  - Syntax
  - Einsatz
- Verwertungskriterien
- Beispiel Weinkeller

Ausblick

- Erweiterte Modellierungskonzepte
  - Kollaborationen
  - Muster/Frameworks
  - Object Constraint Language
- Software-Entwicklungsprozeß/Vorgehensmodell
- Werkzeuge
- weitere Verwertungskriterien