Inhaltsverzeichnis

[Klassendiagramm 1](#_Toc536001090)

[Attribute und Operationen 1](#_Toc536001091)

[Assoziation 2](#_Toc536001092)

[Aggregation 5](#_Toc536001093)

[Generalisierung 6](#_Toc536001094)

[Abstrakte Klasse 6](#_Toc536001095)

[Mehrfachvererbung 7](#_Toc536001096)

[Objektdiagramm 8](#_Toc536001097)

[Deployment Diagram (Verteilungsdiagramm) 9](#_Toc536001098)

[Komponenten-Diagramm 10](#_Toc536001099)

[PaketDiagramm 11](#_Toc536001100)

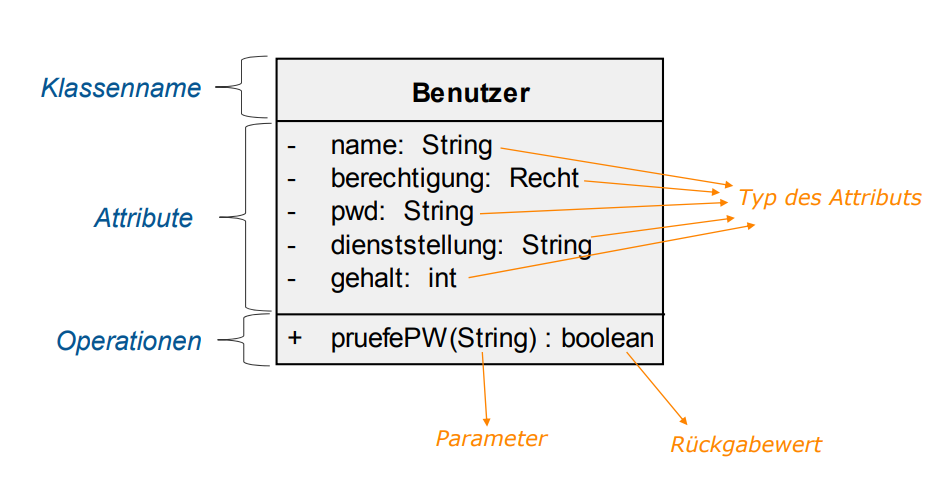
[Analysemuster 12](#_Toc536001101)

## Klassendiagramm

Klassendiagramm: Beschreibt den strukturellen Aspekt eines Systems auf Typebene in Form von Klassen, Interfaces und Beziehungen

Klasse in UML: Schablone, Typ

Objekt: Ausprägung einer Klasse



### Attribute und Operationen

Sichtbarkeiten von Attributen und Operationen:

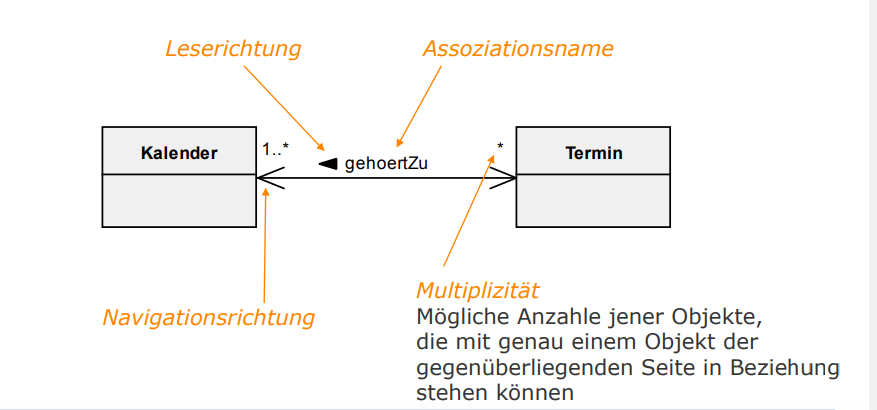
* + ... public
* - ... private
* # ... protected
* ~ ... package (vgl. Java)

Eigenschaften von Attributen:

* „ /“ attributname: abgeleitetes Attribut
* Bsp.: /alter:int
* {optional}: Nullwerte sind erlaubt
* [n..m]: Multiplizität

### Assoziation

Assoziationen zwischen Klassen modellieren mögliche Objektbeziehungen zwischen den Instanzen der Klassen



Rolle ( Teilnehmer, Vortragender => bei Referent)

Assoziation: Navigationsrichtung

Eine gerichtete Kante gibt an, in welche Richtung die Navigation von einem Objekt zu seinem Partnerobjekt erfolgen kann

Ein nicht-navigierbares Assoziationsende wird durch ein "X" am Assoziationsende angezeigt

Ungerichtete Kanten bedeuten "keine Angabe über Navigationsmöglichkeiten"

Die Angabe von Navigationsrichtungen stellt einen Hinweis für die spätere Entwicklung dar

Assoziation als Attribut

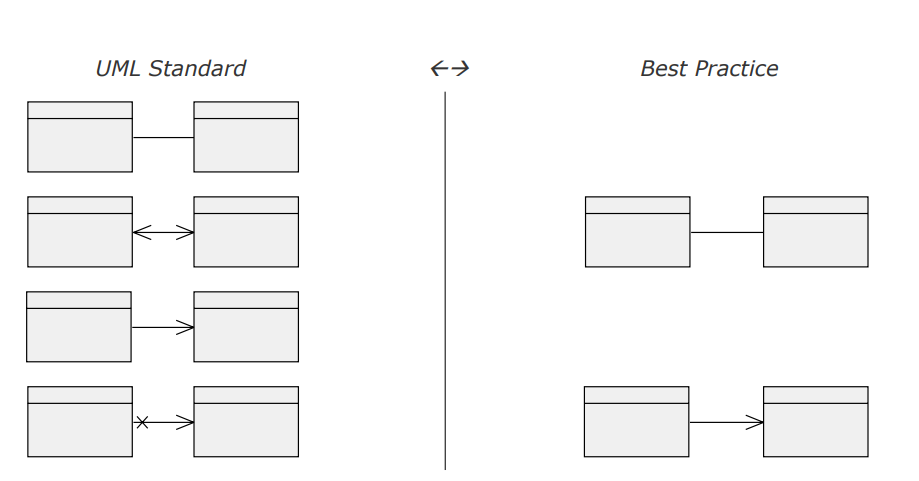
Ein navigierbares Assoziationsende hat die gleiche Semantik wie ein Attribut der Klasse am gegenüberliegenden Assoziationsende

Ein navigierbares Assoziationsende kann daher anstatt mit einer gerichteten Kante auch als Attribut modelliert werden

Die mit dem Assoziationsende verbundene Klasse muss dem Typ des Attributs entsprechen

Die Multiplizitäten müssen gleich sein

Für ein navigierbares Assoziationsende sind somit alle Eigenschaften und Notationen von Attributen anwendbar



Assoziation: Multiplizität

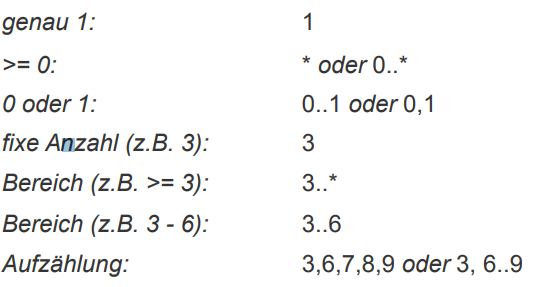
Bereich: "min .. max"

Beliebige Anzahl: "\*" (= 0.. \*)

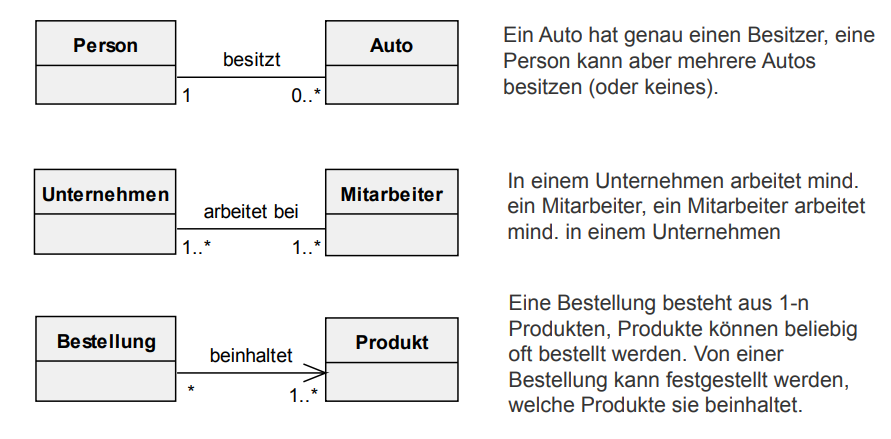
Aufzählung möglicher Kardinalitäten (durch Kommas getrennt)

Defaultwert: 1

Beispiele für Multiplizitäten:

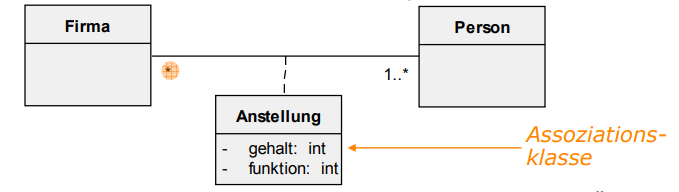


Beispiele für Assoziationen

  
Assoziationsklasse

Kann Attribute der Assoziation enthalten

Bei m:n-Assoziationen mit Attributen notwendig



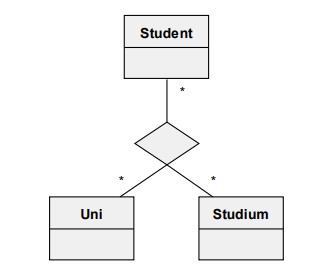
n-äre Assoziation

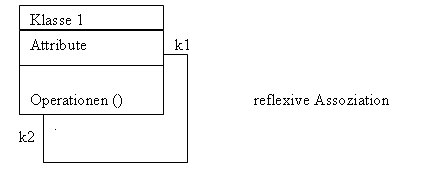
Beziehung zwischen mehr als zwei Klassen

Navigationsrichtung kann nicht angegeben werden

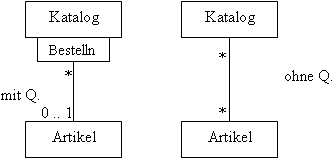
Multiplizitäten geben an, wie viele Objekte einer Rolle/Klasse einem festen (n-1)-Tupel von Objekten der anderen Rollen/Klassen zugeordnet sein können

Multiplizitäten implizieren Einschränkungen, in einem bestimmten Fall funktionale Abhängigkeiten



**Reflexive Assoziation**

**Qualifizierte Assoziation**

Ist ein spezielles Attribut der Assoziation, dessen Wert ein oder mehrere Objekte auf der anderen Seite der Assoziation selektiert, d.h. die Qualifikationsangabe zerlegt die Menge der Objekte am anderen Ende der Assoziation in Teilmengen. Der *qualifier* kann auch aus mehreren Attributen bestehen.

### Aggregation

Aggregation ist eine spezielle Form der Assoziation mit folgenden Eigenschaften:

Transitivität: C ist Teil von B u. B ist Teil von A ◊ C ist Teil von A

Bsp.: Kühlung = Teil von Motor & Motor = Teil von Auto ◊ Kühlung ist (indirekter) Teil von Auto

Anti-Symmetrie: B ist Teil von A ◊ A ist nicht Teil von B

Bsp.: Motor ist Teil von Auto, Auto ist nicht Teil von Motor

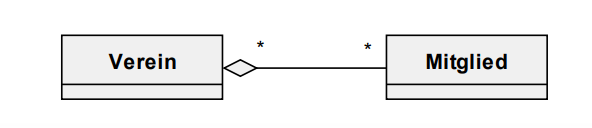
UML unterscheidet zwei Arten von Aggregationen:

Schwache Aggregation (shared aggregation)

Schwache Zugehörigkeit der Teile, d.h. Teile sind unabhängig von ihrem Ganzen

Die Multiplizität des aggregierenden Endes der Beziehung (Raute) kann > 1 sein

Es gilt nur eingeschränkte Propagierungssemantik

Die zusammengesetzten Objekte bilden einen gerichteten, azyklischen Graphen

Starke Aggregation – Komposition (composite aggregation)

Ein bestimmter Teil darf zu einem bestimmten Zeitpunkt in maximal einem zusammengesetzten Objekt enthalten sein

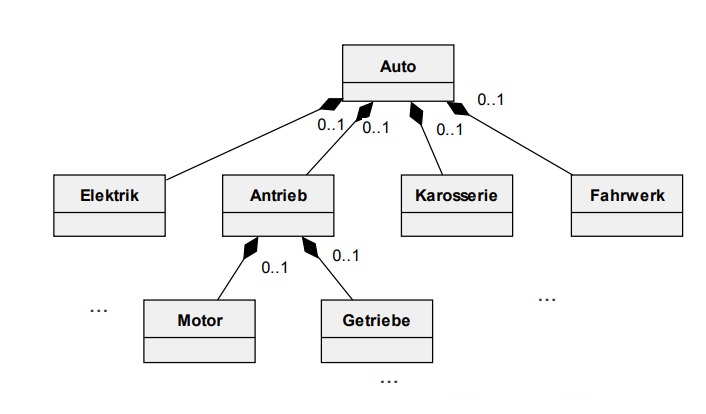
Die Multiplizität des aggregierenden Endes der Beziehung kann (maximal) 1 sein

Abhängigkeit der Teile vom zusammengesetzten Objekt

Propagierungssemantik

Die zusammengesetzten Objekte bilden einen Baum

Mittels starker Aggregation kann eine Hierarchie von „Teil-von“- Beziehungen dargestellt werden (Transitivität!)



### Generalisierung / Spezialisierung

Leserichtung Superklasse zu Subklasse => Spezialisierung

Leserichtung Subklasse zu Superklasse => Generalisierung

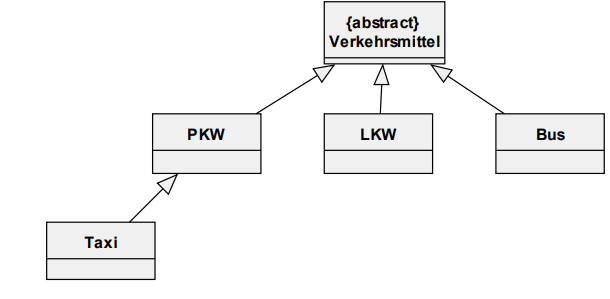
Taxonomische Beziehung zwischen einer spezialisierten Klasse und einer allgemeineren Klasse

Die spezialisierte Klasse erbt die Eigenschaften der allgemeineren Klasse

Kann weitere Eigenschaften hinzufügen

Eine Instanz der Unterklasse kann überall dort verwendet werden, wo eine Instanz der Oberklasse erlaubt ist (zumindest syntaktisch)

Mittels Generalisierung wird eine Hierarchie von „ist-ein“- Beziehungen dargestellt (Transitivität!)



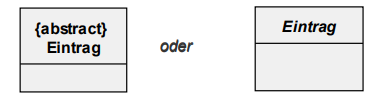
### Abstrakte Klasse

Klasse, die nicht instanziert werden kann

Nur in Generalisierungshierarchien sinnvoll

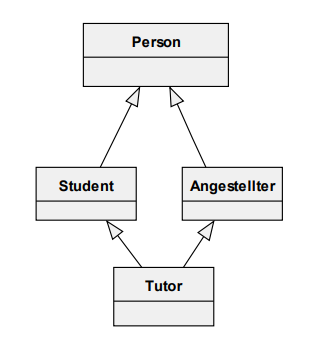
Dient zum "Herausheben" gemeinsamer Merkmale einer Reihe von Unterklassen

Notation: Schlüsselwort {abstract} oder Klassenname in kursiver Schrift



### Mehrfachvererbung

Klassen müssen nicht nur eine Oberklasse haben, sondern können auch von mehreren Klassen erben



## Objektdiagramm

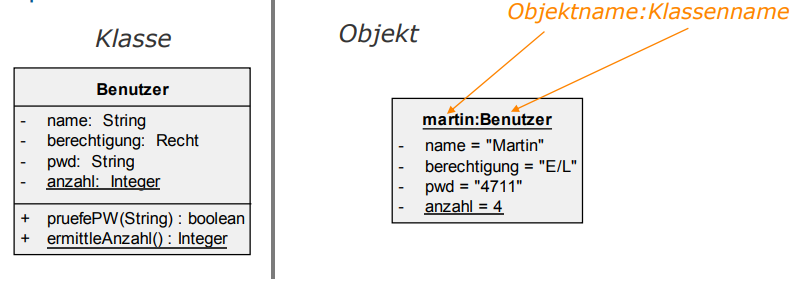
Beschreibt den strukturellen Aspekt eines Systems auf Instanzebene in Form von Objekten und Links

Momentaufnahme (snapshot) des Systems – konkretes Szenario

Ausprägung zu einem Klassendiagramm

Eigentlich eine »Instanzspezifikation

Prinzipiell kann jede Diagrammart auf Instanzebene modelliert werden



Objektdiagramm: Basiskonzepte

Basiskonzepte des Objektdiagramms

Instanz einer Klasse: Objekt

Instanz einer Assoziation: Link

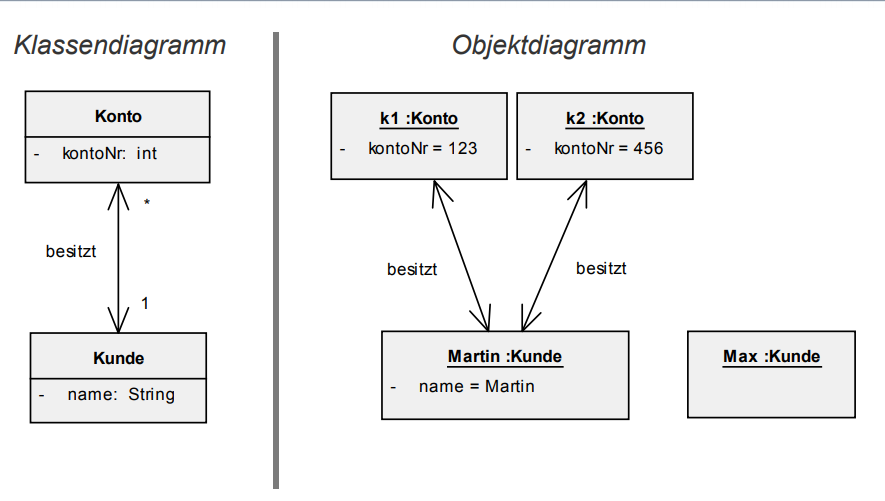
Instanz eines Datentyps: Wert

Einheitliche Notationskonventionen

Gleiches Notationselement wie auf Typebene benutzen

Unterstreichen (bei Links optional)

Objektdiagramm muss nicht vollständig sein

Z.B. können Werte benötigter Attribute fehlen, aber auch Instanzspezifikation abstrakter Klassen modelliert werden

## Deployment Diagram (Verteilungsdiagramm)

Es wird verwendet, um die Darstellung der Verteilung von Komponenten auf Rechenknoten zu zeigen.

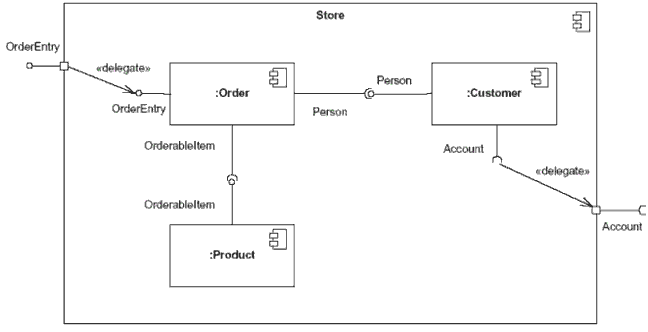
Die Darstellung umfasst dabei typischerweise Rechnerknoten, Komponenten, Artefakte, Ausprägungsspezifikationen, Verbindungen und Verteilungsbeziehungen.

|  |
| --- |
|  |
|  |
|  |

## 

## Komponenten-Diagramm

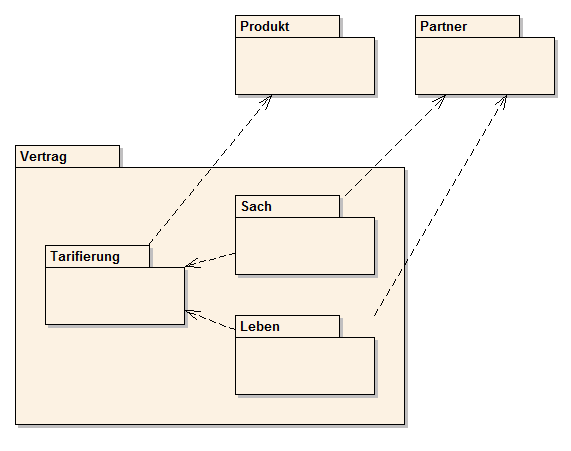
Es ist ein Strukturdiagramm. Die Darstellung umfasst dabei typischerweise Komponenten mit deren Schnittstellen bzw. Ports. Es zeigt auch, wie Komponenten über Abhängigkeitsbeziehungen und Konnektoren miteinander verbunden sind.



## 

## PaketDiagramm

Ein Paket (Package) ist eine logische Ansammlung von Modellelementen beliebigen Typs, mit denen das Gesamtmodell in kleinere überschaubare Einheiten gegliedert wird.



## Analysemuster

Ein Analysemuster ist eine Vorlage, um eine organisationale, soziale oder wirtschaftliche Problemstellung in einer Fachdomäne zu lösen.

**Beispiel**

Eine Rechnung enthält eine oder mehrere Rechnungspositionen.



Eigenschaften

* Beziehungstyp: Komposition
* Ein Ganzes besteht aus gleichartigen Teilen (es gibt nur eine Teil-Klasse).
* Teil-Objekte sind einem Aggregat-Objekt fest zugeordnet, lassen sich aber löschen, ohne das Ganze zu löschen.
* Die Attribut-Werte des Aggregat-Objekts sind auch für die Teil-Objekte gültig (hier: Nummer und Datum der Rechnung)
* Das Aggregat-Objekt enthält in der Regel mindestens ein Teil-Objekt (Kardinalität ist mindestens 1, meistens 1…\*).