

6 Klassen- und Objektdiagramme

Inhalt

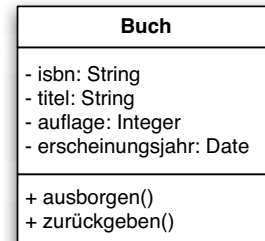
6.1 Klassen- bzw. Objektdiagramme	-2
6.2 Klasse	-3
6.3 Objekt	-8
6.4 Allgemeine Assoziation	-9
6.5 Qualifizierte Assoziation	-11
6.6 Attributierte Assoziation (Assoziationsklasse)	-12
6.7 Mehrgliedrige Assoziation	-14
6.8 Aggregation	-15
6.9 Komposition	-16
6.10 Generalisierung	-17

6.1 Klassen- bzw. Objektdiagramme

- ❑ Aus historischen Gründen hat sich für Klassendiagramme und Objektdiagramme der (unglückliche) gemeinsame Term Klassendiagramme entwickelt.
- ❑ Genauer gesagt sind Klassen- und Objektdiagramme statische Strukturdiagramme.
- ❑ Klassendiagramme beschreiben die abstrakte Modellebene (Schema)
- ❑ Objektdiagramme beschreiben Elemente der Ausprägungsebene (Population)

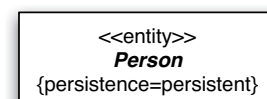
6.2 Klasse

- ❑ Klassen werden als Rechteck dargestellt, das horizontal in einzelne Abschnitte untergliedert ist.
- ❑ Mit Ausnahme des obersten Abschnitts können alle Abschnitte fehlen, wobei das Nichtaufscheinen eines Abschnitts im Diagramm keine Rückschlüsse auf die Existenz der durch ihn verwalteten Charakteristika zulässt.
- ❑ Im obersten Abschnitt sind **Name** und **allgemeine Eigenschaften** der Klasse vermerkt.
- ❑ Der zweite Abschnitt enthält die **Attribute**.
- ❑ Der dritte Abschnitt enthält die **Operationen**.
- ❑ Danach sind prinzipiell noch weitere Abschnitte zulässig, die jedoch nicht von UML normiert wurden.



Erster Abschnitt

- ❑ Im ersten Abschnitt können neben dem Klassennamen noch ein **Stereotyp** und **Eigenschaftsangaben** eingetragen werden.
- ❑ Stereotypen sind Erweiterungsmechanismen, mit denen die Klassenhierarchie des Metamodells erweitert werden kann, indem jedes Modellelement (z.B. Klasse) mit höchstens einem Stereotyp weiter kategorisiert werden kann.
- ❑ Stereotype erscheinen in den "stereotypischen" Anführungszeichen << ... >> oberhalb des Klassennamens.
- ❑ Eigenschaftsangaben erfolgen durch Auflistung in geschwungenen Klammern { ... }. Eine häufig benützte Eigenschaft ist **abstract** zur Kennzeichnung einer abstrakten Klasse. Alternativ kann eine abstrakte Klasse durch einen kursiv geschriebenen Namen dargestellt werden.



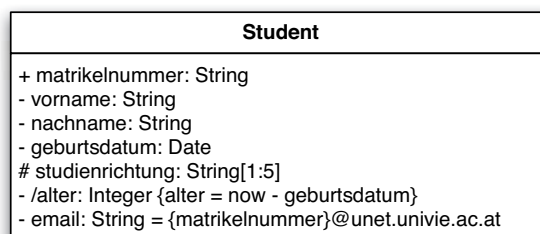
Zweiter Abschnitt

- ❑ Im zweiten Abschnitt werden die **Attribute** der Klasse aufgelistet. Die einfachste Form ist die Angabe des Attributnamens.
- ❑ Die **Sichtbarkeitsangabe** ist C++ nachempfunden:
 - public (+) Attribute stehen für öffentliche bzw. uneingeschränkte Sichtbarkeit.
 - private (-) Attribute sind nur innerhalb der Klasse sichtbar und können nur in Methoden der Klasse verwendet werden.
 - protected (#) Attribute signalisieren Sichtbarkeit für die entsprechende Klasse und deren Unterklassen.
 - Sprachspezifische Erweiterungen sind möglich.
- ❑ Die **Typangabe** ist optional und kann sprachspezifisch erfolgen.
- ❑ Einem Attribut kann ein **Initialwert** zugewiesen werden.
Dabei kann eine beliebige sprachspezifische Syntax gewählt werden.
- ❑ Es ist auch eine **Multiplizitätsangabe** für Attribute möglich.
Die Anzahl der möglichen Wiederholungen wird als Intervall ausgedrückt, z.B. [1..*]

- ❑ Die Tatsache, dass ein **Attribut abgeleitet** ist, d.h., dass seine Ausprägungen aus anderen Informationen berechenbar sind, lässt sich durch einen dem Namen vorangestellten Schrägstrich festhalten.

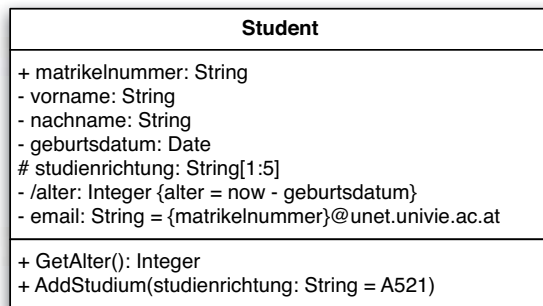
Zusätzlich kann die Berechnungsvorschrift als Einschränkung in geschweiften Klammern angegeben werden.

- ❑ Graphische Darstellung:



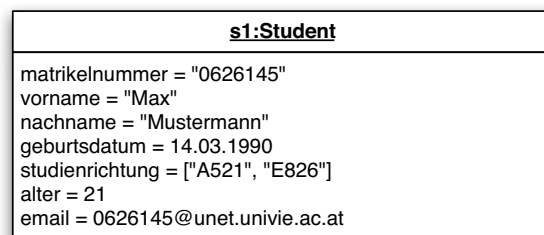
Dritter Abschnitt

- ❑ Im dritten Abschnitt werden die von der Klasse angebotenen **Operationen** angeführt.
- ❑ Für jede Operation wird die Parameterliste angegeben. Wobei für jeden Parameter zumindest der Name angegeben werden muss. Typangabe, Datenflussrichtung und Standardwert sind für die Parameter optional.
- ❑ Nach der Parameterliste kann der Ergebnistyp angegeben werden. Wird er ausgelassen, so bedeutet dies, dass die Operation nichts zurückliefert.
- ❑ Die Angabe der Sichtbarkeit (public, private, protected) ist entsprechend jener für Attribute.
- ❑ Weitere Aspekte einer Operation können wiederum in einer Eigenschaftsliste am Ende der Spezifikation angegeben werden.



6.3 Objekt

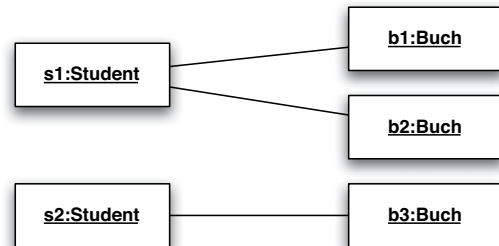
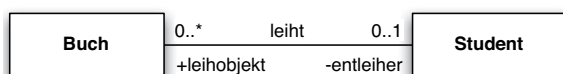
- ❑ UML benützt einen einheitlichen Mechanismus um generelle **Instanzen** von dem dazu gehörigen Typ zu unterscheiden, wie etwa Objekte von Klassen, in dem die Instanz dasselbe graphische Symbol wie der Typ verwendet, jedoch der Name (und falls vorhanden die Typangabe) unterstrichen wird.
- ❑ Objekten, als Instanzen von Klassen, können für ihre Attribute Werte zugeordnet werden.
- ❑ Bei der Angabe des Objektes kann entweder die Angabe des Typs oder des Objekt-namens (anonyme Objekte) entfallen
- ❑ Graphische Notation:



6.4 Allgemeine Assoziation

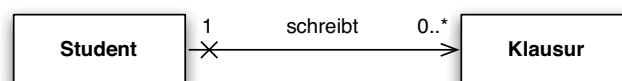
- ❑ Allgemeine Assoziationen beschreiben die gemeinsame Struktur einer Menge von in der Regel statischen Beziehungen zwischen Objekten.
- ❑ Binäre Assoziationen, die Beziehungen zwischen je zwei Klassen beschreiben, werden in UML durch einfache Verbindungslinien (Kanten) zwischen beteiligten Klassen dargestellt.
- ❑ Graphische Darstellung im Objektdiagramm:

- ❑ Graphische Darstellung im Klassendiagramm:



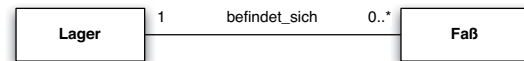
- ❑ Um die Aussagekraft von Assoziationen zu erhöhen können weitere Zusatzaussagen getroffen werden: Assoziationsname, Rollenbezeichnungen, Multiplizitäten.

- ❑ Der Name einer Assoziation wird in der Regel in der Mitte der Assoziationskante notiert. Wie bei abgeleiteten Attributen können abgeleitete Assoziationen durch einen vorangestellten Schrägstrich ausgezeichnet werden.
- ❑ Assoziationsrollen geben an, welche Rollen die beteiligten Objekte im Rahmen der Beziehung spielen. Angaben über die Rollen werden an den Enden der Verbindungslinien notiert, sie werden daher auch manchmal als Assoziationsenden bezeichnet.
- ❑ Ein bestimmtes Objekt kann mit einer gewissen Anzahl von Partnerobjekten der gegenüberliegenden Klasse verbunden sein. Dies lässt sich aus einem geeigneten Objektdiagramm erkennen. Im Klassendiagramm lässt sich natürlich nur die mögliche Anzahl in Form der Multiplizität der Rolle festhalten. Die Multiplizität kann in Form eines Intervalls angegeben werden.
- ❑ Assoziationskanten können auch gerichtet sein, um explizit anzugeben, in welche Richtung die Navigation von einem Objekt zu seinem Partnerobjekt erfolgen kann.

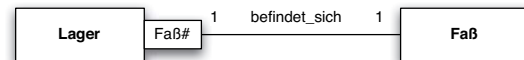


6.5 Qualifizierte Assoziation

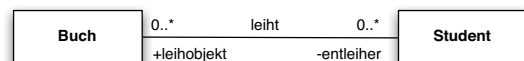
- ❑ In manchen Fällen ist es sinnvoll, bei Assoziationsrollen mit Multiplizität größer als eins die Menge der Objektbeziehungen durch ein oder mehrere sogenannte qualifizierende Attribute in disjunkte Teilmengen zu zerlegen.



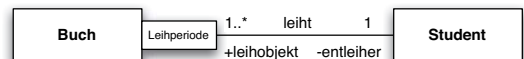
- ❑ Handelt es sich bei den qualifizierenden Attributen um einen Schlüssel der Assoziation so entstehen einelementige Teilmengen, und die Multiplizität der Assoziation wird auf eins reduziert.



- ❑ Qualifizierende Attribute werden in einem kleineren Rechteck an jenem Ende der Assoziation notiert, dessen Rolle den Kontext für die Qualifikation darstellt.



- ❑ Weiteres Beispiel bei historischen Daten

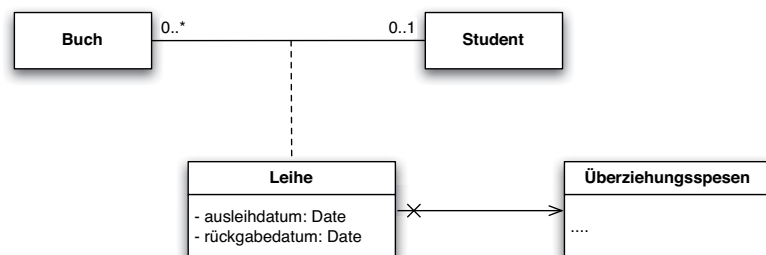


6.6 Attributierte Assoziation (Assoziationsklasse)

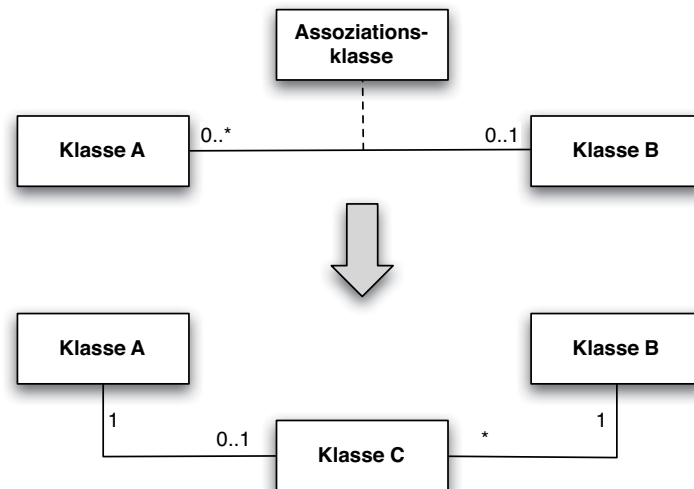
- ❑ Wenn Assoziationen durch eigene (nicht qualifizierende) Attribute näher beschrieben werden sollen oder ihrerseits an anderen Assoziationen teilnehmen sollen, müssen sie zuerst “klassenwertig” gemacht werden.
- ❑ Sogenannte Assoziationsklassen verwenden das Klassensymbol, das durch eine strichlierte Linie mit der Assoziationskante verbunden ist.
- ❑ Semantisch sind Assoziation und Assoziationsklasse identisch, d.h. der Name der Assoziationsklasse entspricht dem Namen der Assoziation.

- ❑ Graphische Notation:

- ❑ Die attributierte Assoziation sollte in der Praxis vermieden werden, da sie nicht objektorientiert ist



- Eine bidirektionale Assoziation ist als Kurzschreibweise für zwei einzelne entgegengesetzt gerichtete Assoziationen zu sehen



6.7 Mehrgliedrige Assoziation

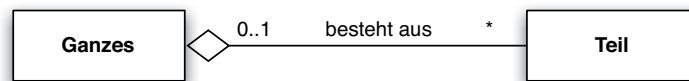
- In manchen Fällen kann eine Beziehung mehr als zwei Partnerobjekte betreffen.
- Sie wird - wie bei ER-Diagrammen - durch eine Raute, die mit den teilnehmenden Klassen verbunden ist, dargestellt.



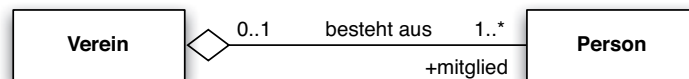
- Die mehrgliedrige Assoziation ist mit verschiedenen semantischen Detailproblemen (Multiplizitätszuordnung, Verantwortlichkeiten) behaftet, ähnlich wie die attributierte Assoziation kein objektorientiertes Konzept und in der Praxis fehleranfällig. Man sollte daher den darzustellenden Sachverhalt mit gewöhnlichen Assoziationen beschreiben

6.8 Aggregation

- ❑ Die Aggregation ist ein Spezialfall der allgemeinen Assoziation und beschreibt wie sich etwas Ganzes aus seinen Teilen logisch zusammensetzt.

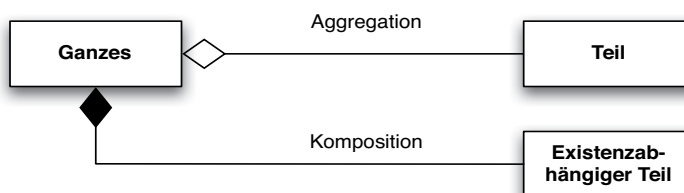


- ❑ Bei der Aggregation handelt es sich um eine asymmetrische Beziehung zwischen nicht gleichwertigen Partnern.
- ❑ Aggregationen werden durch eine Raute dargestellt, die jenes Ende der Assoziationskante markiert, das zur Aggregatklasse, also zum "Ganzen" hinführt.

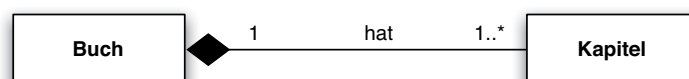


6.9 Komposition

- ❑ Die Komposition ist eine strengere Form der Aggregation, bei der die Teile vom Ganzen existenzabhängig sind. Sie beschreibt, wie sich etwas Ganzes aus Einzelteilen zusammensetzt und diese kapselt.

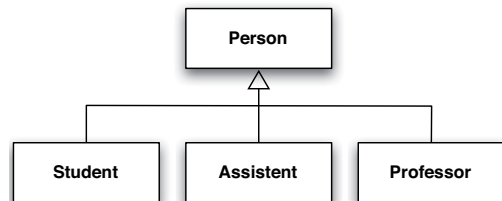
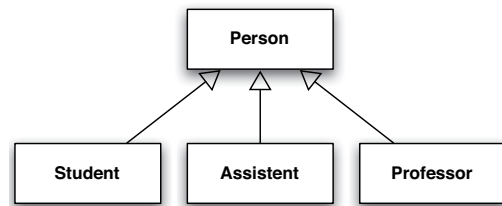


- ❑ Die Kardinalität auf der Seite des Aggregats muss immer 1 sein und jedes Teil darf nur Teil genau eines Kompositionsobjekts sein, sonst wäre die Existenzabhängigkeit widersprüchlich.
- ❑ Die Komposition wird durch eine ausgefüllte Aggregationsraute dargestellt.

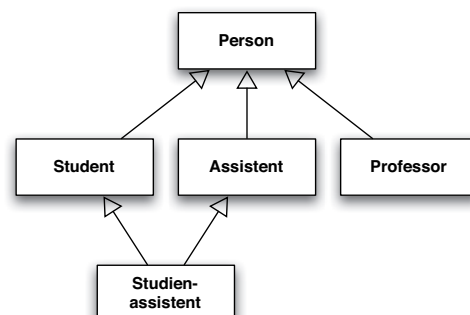
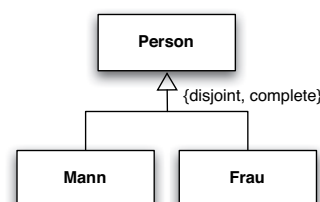


6.10 Generalisierung

- ❑ Die Generalisierung stellt eine taxonomische Beziehung zwischen einer spezialisierten Klasse (Unter- oder Subklasse) und einer allgemeineren Klasse (Ober-, Basis- oder Superklasse) dar.
- ❑ Die Subklasse erbt die Charakteristika der Superklasse und kann weitere Merkmale hinzufügen.
- ❑ Die Generalisierung stellt eine "is-a" Beziehung dar.
- ❑ Die Generalisierung wird durch einen Pfeil mit einem nicht ausgefüllten gleichseitigen Dreieck mit Spitze zur Superklasse notiert. Es können entweder mehrere Generalisierungspfeile eingezeichnet werden, oder diese können zu einem Mehrfachpfeil verschmelzen.



- ❑ Auf die Generalisierungsbeziehung können noch Einschränkungen getroffen werden, die neben das Dreieck in geschwungenen Klammern eingezeichnet werden.
- ❑ Somit kann über vollständig / unvollständig bzw. über überlappend / disjunkt eine Aussage getroffen werden.
- ❑ UML Klassendiagramme selbst unterstützen die Notation von Mehrfachvererbung.





Literatur und Quellen:

- ❑ OMG Unified Modeling Language™ (OMG UML), Version 2.5
Normative Reference: <http://www.omg.org/spec/UML/2.5>
OMG Document Number formal/2015-03-01
- ❑ Kapitel 3.1 in:
Martin Hitz, Gerti Kappel, Elisabeth Kapsammer, Werner Retschitzegger:
UML @ Work - Objektorientierte Modellierung mit UML2.
dpunkt Verlag 2005 / 3. aktualis. u. überarb. Aufl. 2005.
ISBN-13: 9783898642613 ISBN-10: 3898642615
- ❑ Kapitel 2.2 in: Bernd Österreich: UML Kurzreferenz für die Praxis
- ❑ Tool zur Vertiefung: BEE-UP Modelling Tool:
<http://austria.omilab.org/psm/content/bee-up/info>
<http://www.omilab.org/web/guest/omilab-in-education/cmmc>