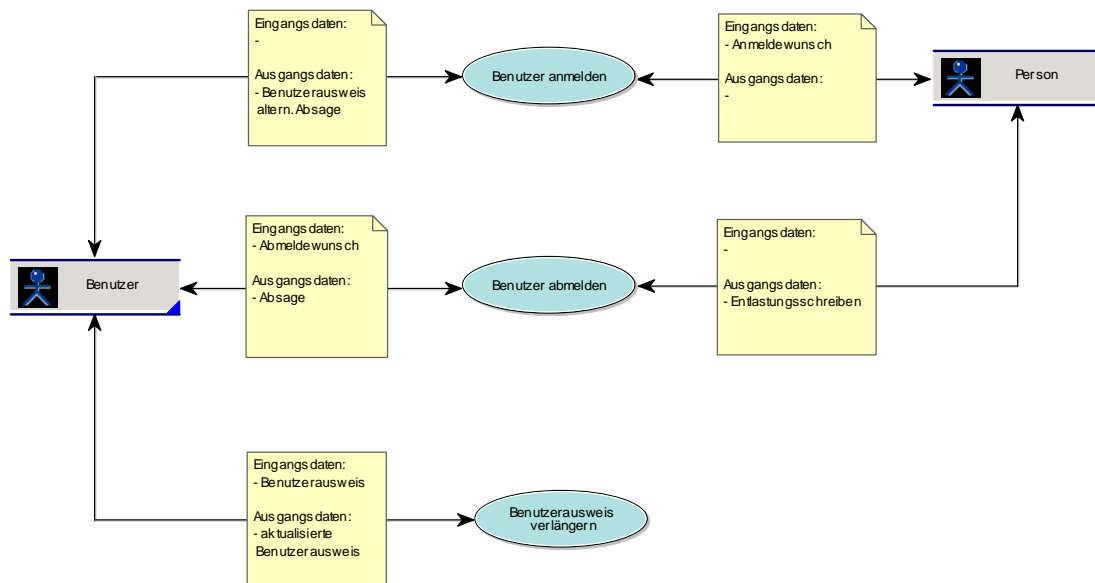


## Das Anwendungsfalldiagramm – 1/3 –

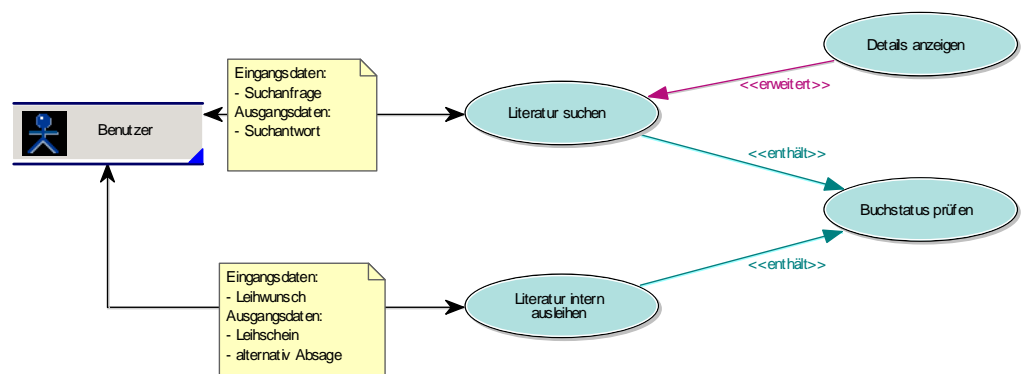
Anwendungsfalldiagramm	
<b>Semantik</b>	Das Anwendungsfalldiagramm stellt die funktionalen Anforderungen aus der Sicht des Anwenders dar. Die funktionalen Anforderungen werden mit dem(den) Beteiligten aus dem Kontext in Beziehung gesetzt, für den (die) der Anwendungsfall einen Nutzen erbringt. Diese Beteiligten sind die sogenannten Akteure. Die Beziehung zwischen den Akteuren und den zugeordneten Anwendungsfällen ist eine Kommunikationsbeziehung.
<b>Topologie</b>	netzförmig
<b>Knoten</b>	<ul style="list-style-type: none"> <li>- Anwendungsfall</li> <li>- Akteur</li> </ul>
<b>Kanten</b>	Beziehung zwischen <ul style="list-style-type: none"> <li>- Anwendungsfall - Akteur: Kommunikationsbeziehung</li> <li>- Anwendungsfall - Anwendungsfall: - enthält (include) - erweitert (extend)</li> <li>- Akteur – Akteur: Generalisierung/Spezialisierung</li> </ul>

Beispiele:



### Hinweis zur Richtungsangabe:

Der AWF „Details anzeigen“ erweitert den AWF „Literatur suchen“.



Der AWF „Literatur suchen“ enthält den AWF „Buchstatus prüfen“.

## Das Anwendungsfalldiagramm - 2/3 -

Begriffserklärung:

**Anwendungsfall** = abstrakte Abbildung einer vom SW-System angebotenen Funktionalität

---

Ein Anwendungsfall kapselt eine Folge von Aktionen, die sequentiell und/oder bedingungsabhängig und/oder zyklisch ausgeführt wird, um die Funktionalität zu realisieren.

Ein Anwendungsfall wird durch ein Ereignis ausgelöst.

Dieses Ereignis kann sein :

entweder

- ein „datengetriebenes Ereignis“  
(Ein Akteur stellt die erforderliche Information - Eingabedaten - für den Start des Anwendungsfalles zur Verfügung. Der Akteur kann ein Anwender oder ein technisches System sein.)
- oder
- ein zeitliches Ereignis (relativ oder absolut)  
(Wenn eine zeitliche Bedingung erfüllt ist, dann startet der Anwendungsfall.)

Ein Anwendungsfall führt i.R. zu einem von außen sichtbaren Ergebnis für den Anwender.

(Ergebnis = von außen für den Anwender sichtbare Reaktion → Ausgabedaten)

Ein Anwendungsfall wird näher beschrieben durch zum Beispiel

- eine textliche Beschreibung (frei, nach einer Textschablone oder nach einer Vorlage)
- ein Aktivitätsdiagramm  
(Anwendungsfall = Aktivität, die aus einzelnen Aktionen besteht)
- ein Zustandsdiagramm  
(Der gekapselte Prozess befindet sich zu einem bestimmten Zeitpunkt in einem definierten Zustand.)

Analoge Begriffe: use case / Nutzungsfall / Szenario / Skript

**Akteur** = abstrakte Abbildung einer externen Instanz, die mit dem SW-System kommuniziert

---

Ein Akteur beschreibt ein außerhalb des SW-Systems existierendes Objekt, das definiert mit dem System interagiert.

Ein Akteur ist

- ➔ rollenorientiert  
D.h. ein und dasselbe Objekt aus dem Kontext kann unterschiedliche Rollen im Kontext des SW-Systems spielen.
- ➔ aktiv oder passiv
  - aktiver Akteur:
    - Anwender
    - externes technisches System (z.B. Kontrolleinrichtung)
    - zeitliches Ereignis
  - passiver Akteur:  
modelliert ein zur Realisierung des Anwendungsfalles benötigtes Objekt

Analoge Begriffe: Begrenzer / Beteiligter / Boundary

## Das Anwendungsfalldiagramm – 3/3 -

### Enthält-Beziehung

---

Eine Folge von Aktionen wird in (z.B. zwei) Teile aufgeteilt; die Teile werden jeweils einem Anwendungsfall zugeordnet. Damit kann z.B. sichtbar gemacht werden, dass ein komplexer Anwendungsfall einen anderen Anwendungsfall enthält. Das ist insbesondere dann von Vorteil, wenn dieser ausgelagerte Teil auch noch in einem oder mehreren anderen Anwendungsfällen enthalten ist.

Die Enthält-Beziehung dient also der Vermeidung von Redundanz in der weiteren Beschreibung von Anwendungsfällen, indem identische Teilfunktionen, die in verschiedenen Anwendungsfällen enthalten sind, ausgelagert und in einem gesonderten Anwendungsfall gekapselt werden.

### Erweitert-Beziehung

---

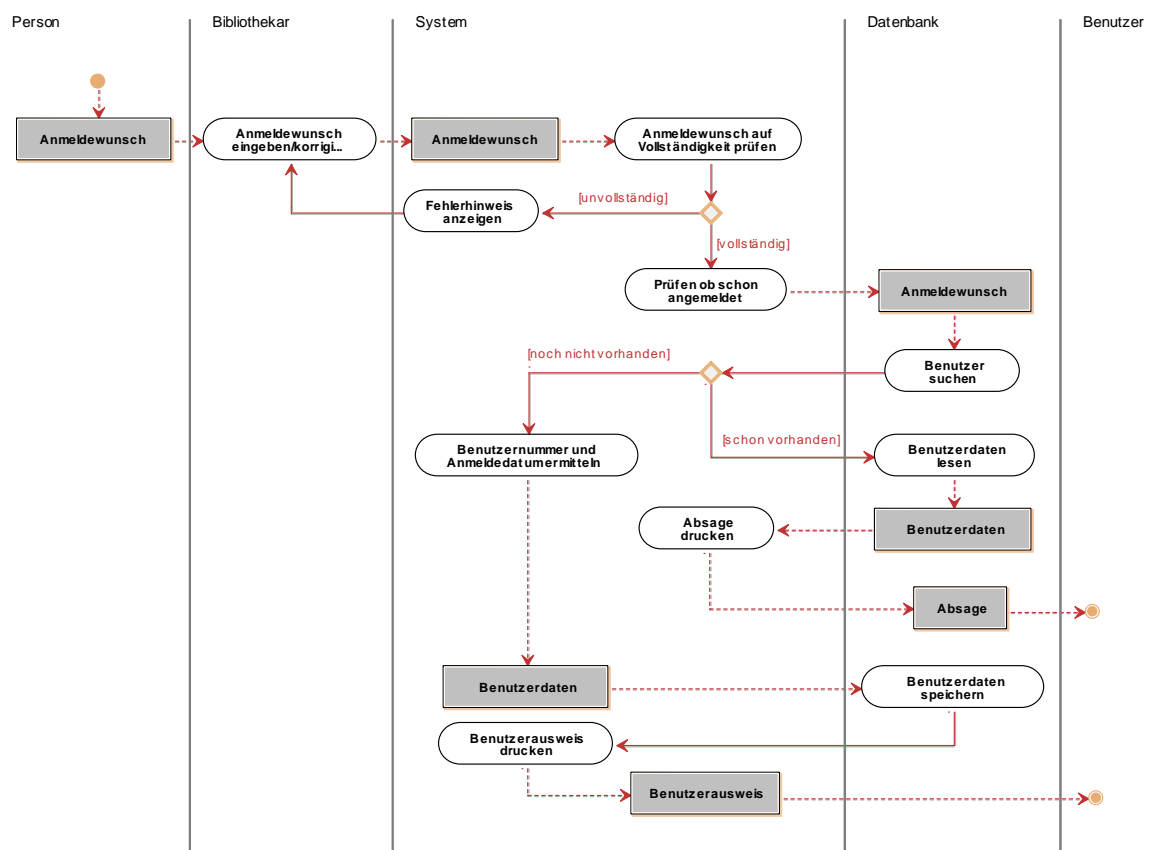
Ein Anwendungsfall ist quasi beendet, könnte aber nach Wunsch noch speziell weiter geführt werden. Diese meist optionale Erweiterung wird in einem gesonderten Anwendungsfall gekapselt, ausgelagert und damit sichtbar gemacht.

### Hinweis:

Die zu frühzeitige Verwendung von Enthält- und Erweitert-Beziehungen kann dazu führen, dass statt der gekapselten Funktionalitäten der innere Ablauf (d.h. der Algorithmus) festgelegt wird. Dies ist aber besser in einem Aktivitätsdiagramm dargestellt.

## Das Aktivitätsdiagramm - 1/2 -

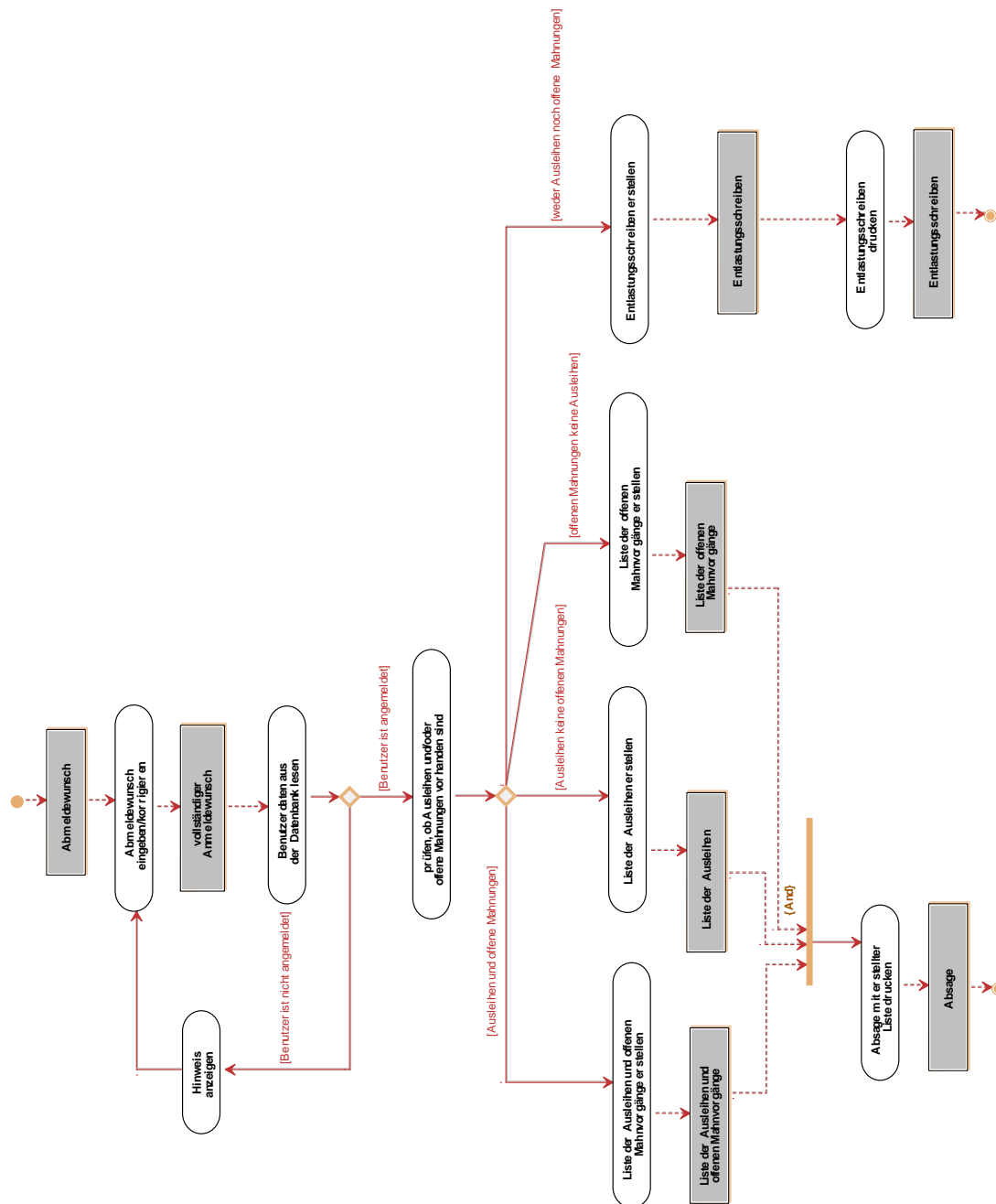
Aktivitätsdiagramm	
<b>Semantik</b>	<p>Das Aktivitätsdiagramm stellt dar, welche Aktionen (Teilfunktionen) in einer Aktivität (Funktionalität) gekapselt sind. D.h. ein Aktivitätsdiagramm kapselt einen Algorithmus. Die einzelnen Aktionen können sequenziell, bedingungsabhängig oder zyklisch ablaufen. Dabei können einzelne Aktionen jeweiligen Verantwortungsbereichen zugeordnet werden.</p> <p>Das Aktivitätsdiagramm nutzt Darstellungsmittel aus Datenflussdiagrammen, Petri.Netzen, Struktogrammen und Programmablaufplänen.</p>
<b>Topologie</b>	netzförmig
<b>Knoten</b>	<ul style="list-style-type: none"> <li>- Aktion</li> <li>- Start- / Endzustand</li> <li>- Entscheidungsknoten</li> <li>- Synchronisationsknoten (zum Parallelisieren und wieder Synchronisieren)</li> </ul> <p>Mit diesen Knoten kann ein Ablauf gut und leicht verständlich dargestellt werden.</p> <p>Eine Vielzahl weiterer spezielle Knoten<sup>1</sup> zur Abbildung aller gängigen Ablaufkonzepte einer höheren Programmiersprache lässt eine spezialisierte algorithmische Darstellung zu. Die Anwendung dieser Modellelemente erfordern eine genaue Kenntnis der Modellierungssprache.</p>
<b>Kanten</b>	<ul style="list-style-type: none"> <li>- Transition (gerichteter Übergang von einer Aktion zu einer anderen)</li> <li>- Objektfluss (Der gerichtete Übergang von einer Aktion zu einer anderen ist begleitet von einem Objekt. → „fließendes Objekt“ zwischen zwei Aktionen)</li> </ul>



<sup>1</sup> Rupp, C.; Queins, S.; Zengler, B. UML2 Glasklar, Hanser, 4. Auflage 2012

## Das Aktivitätsdiagramm – 2/2 -

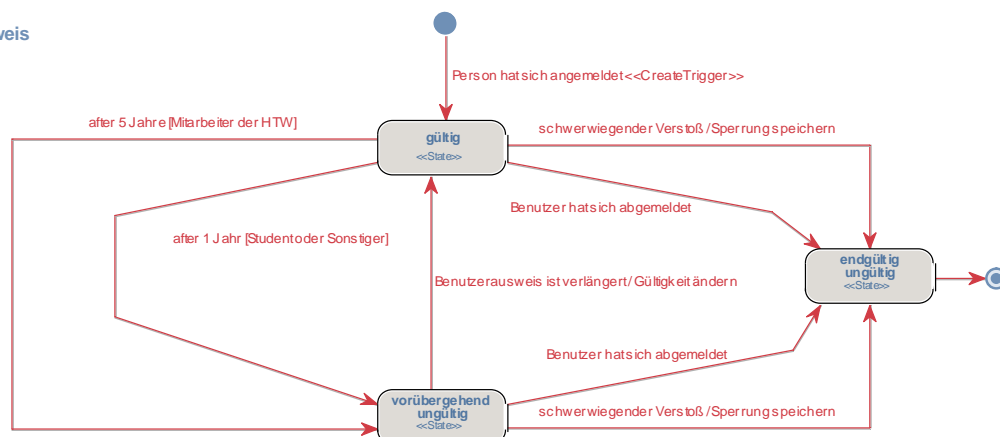
Benutzer abmelden



## Das Zustandsdiagramm – 1/2 -

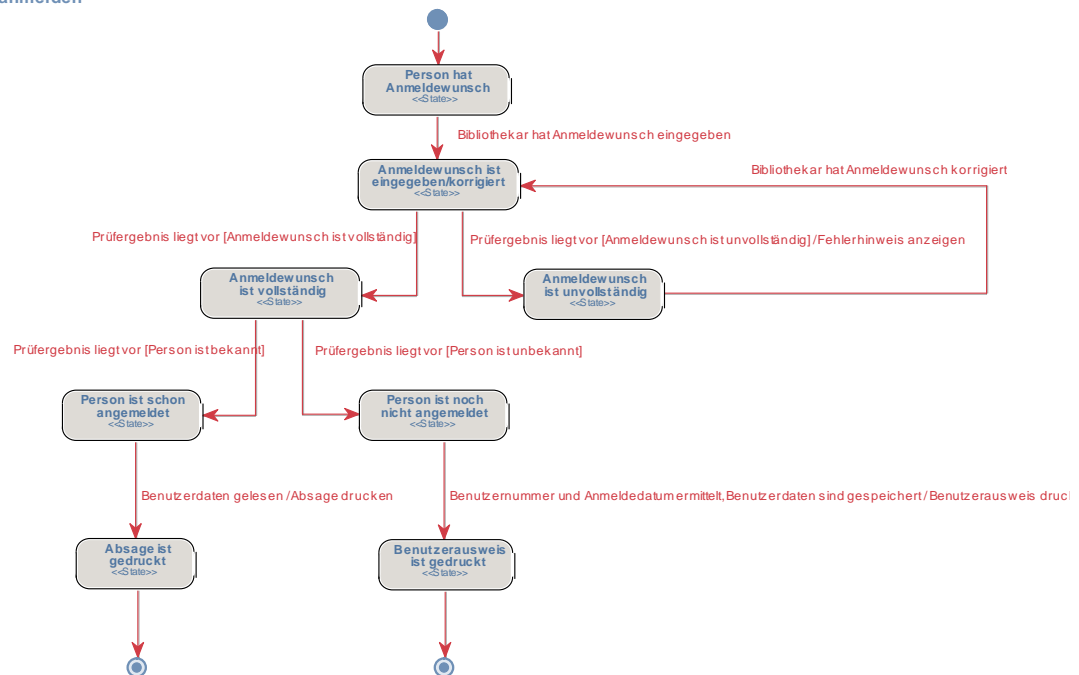
Zustandsdiagramm	
Semantik	<p>Das Zustandsdiagramm stellt die einzelnen Zustände dar, die eine Betrachtungseinheit (ein System, ein Teilsystem oder eine Objekt ..) haben kann. Gleichzeitig werden die Zustandsübergänge beschrieben. Das heißt es wird modelliert, wann ein Zustand in einen anderen wechselt (nach welchem Ereignis) und welche nach außen sichtbare Reaktion dabei erfolgt. Eventuell muss für diesen Zustandswechsel auch eine definierte Bedingung erfüllt sein. Eine Betrachtungseinheit befindet sich dabei zu einem bestimmten Zeitpunkt in einem definierten (Elementar-)Zustand und nur in diesem.</p> <p>Für die Modellierung gilt als Ausgangspunkt ein ausgewählter Zustand, als Endpunkt können mehrere Zustände als Varianten möglich sein. Dieser Endzustand kann i.R. nicht mehr verlassen werden.</p> <p>Das Zustandsdiagramm modelliert das Zustandsverhalten eines Prozesses oder eines Objektes oder eines (Teil-)Systems.</p> <p>Es ist insbesondere geeignet, Steuerverhalten abzubilden (Lift-, Ampel-, CD-Steuerung usw.).</p>
Topologie	netzförmig
Knoten	<p>grundlegende Knoten (auch unabhängig von der UML):</p> <ul style="list-style-type: none"> <li>• Startzustand</li> <li>• Endzustand</li> <li>• Zwischenzustand → Zustand</li> </ul> <p>weitere Knoten ( UML 2.x)</p> <ul style="list-style-type: none"> <li>• zusammengesetzter Zustand mit Eintritts- und Austrittspunkt</li> <li>• Entscheidung</li> <li>• Gabelung/Vereinigung</li> <li>• Kreuzung</li> <li>• Terminator</li> <li>• flach und Tiefe Historie</li> </ul>
Kanten	<p>Transitionen</p> <p>Ereignis [Bedingung ] / Aktion</p>
<p>Regionen dienen der Modellierung von Zuständen einer Betrachtungseinheit, die parallel möglich sind. Daher können Zustände aus der einen Region nicht in Zustände aus einer anderen Region wechseln. Regionen werden über Start- bzw. Endzustand oder über Eintritts- bzw. Austrittspunkt erreicht bzw. verlassen.</p>	

## Benutzeranweis

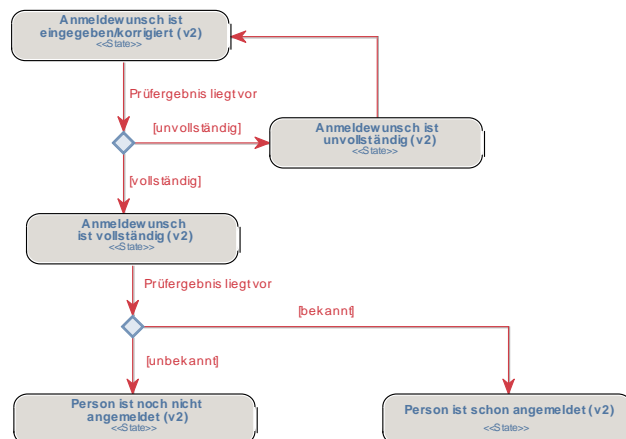


## Das Zustandsdiagramm – 2/2 -

### Benutzer anmelden



alternative Darstellung



**Das Klassendiagramm – 1/3 –**

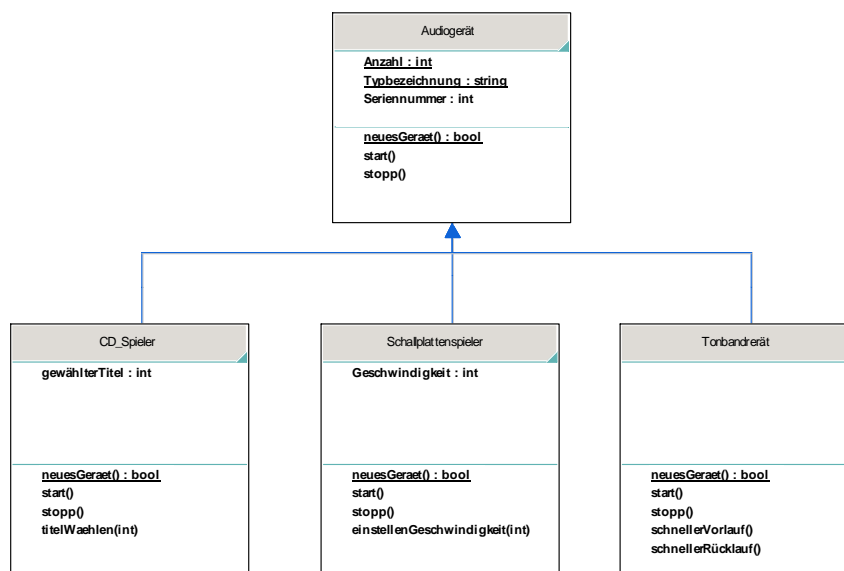
Das Klassendiagramm kann aus unterschiedlichen Perspektiven betrachtet werden:

- **Perspektive der Analyse** zur Modellierung des fachlichen Konzeptes (Analyseklassendiagramm)
- **Perspektive des Entwurfes** zur Modellierung des Lösungskonzeptes insbesondere zur Darstellung von Schnittstellen
- **Perspektive der Implementierung** zur Modellierung der Implementationsdetails

Hinweis: Die folgende Tabelle bezieht sich auf das Analyse-Klassendiagramm.

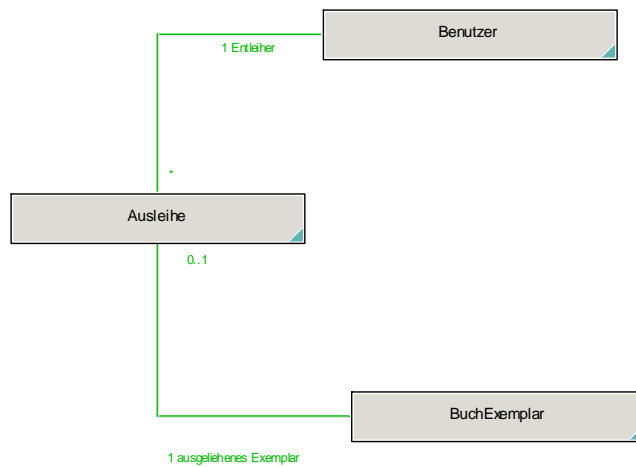
<b>Analyse-Klassendiagramm</b>	
<b>Semantik</b>	Das Klassendiagramm stellt die Klassen und deren statische Beziehungen dar.
<b>Topologie</b>	netzförmig
<b>Knoten</b>	Klasse <ul style="list-style-type: none"> <li>- „normale“ Klasse, d.h. Klassen, von den eine Instanz erzeugt werden kann</li> <li>- abstrakte Klasse (dient der Einordnung von Typen)</li> </ul>
<b>Kanten</b>	Beziehung zwischen <ul style="list-style-type: none"> <li>- Assoziation: abstrakte der Kommunikationsbeziehung zwischen Objekten</li> <li>- Aggregation: Teile-Ganze-Beziehung d.h.. die Aggregation modelliert eine spezielle Assoziation, die zwischen dem Ganzen und den Teilen</li> <li>- strenge Teile-Ganze-Beziehung → Komposition</li> <li>- Generalisierung („Vererbung“) zur Einordnung von Typen</li> </ul>

Beispiele:

**Generalisierung/Spezialisierung**



## Assoziation:



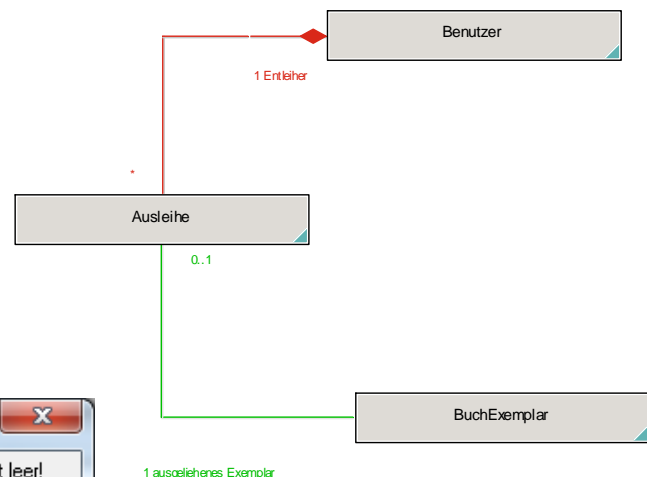
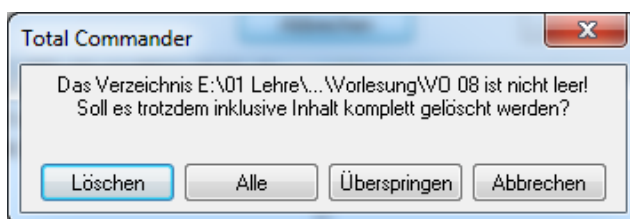
Die Instanzen einer Klasse sind „rollenbehaftet“. Eine Instanz der Klasse Benutzer spielt in der Beziehung zu einem Objekt der Klasse Ausleihe die Rolle des Entleihers; eine Instanz der Klasse BuchExemplar spielt in der Beziehung zu einem Objekt Ausleihe die Rolle des ausgeliehenen Buchexemplares.

Die Multiplizität gibt an, wieviele Objekte an einer Beziehung beteiligt sind. (1, 0..1, 1..\*, \*, 2..6)

## Komposition (= strenge Aggregation):

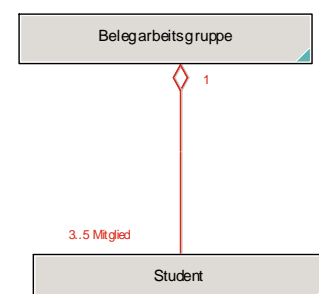
Wenn sich der Benutzer abmeldet, dann existieren auch keine ihm zugeordnete Ausleihen mehr.

Wenn ein Verzeichnis gelöscht wird, dann existieren auch keine Dateien oder Unterverzeichnisse mehr. Daraus resultieren u. U. der Hinweis und die Frage beim Löschen, ob das Verzeichnis inklusive des Inhaltes gelöscht werden soll.



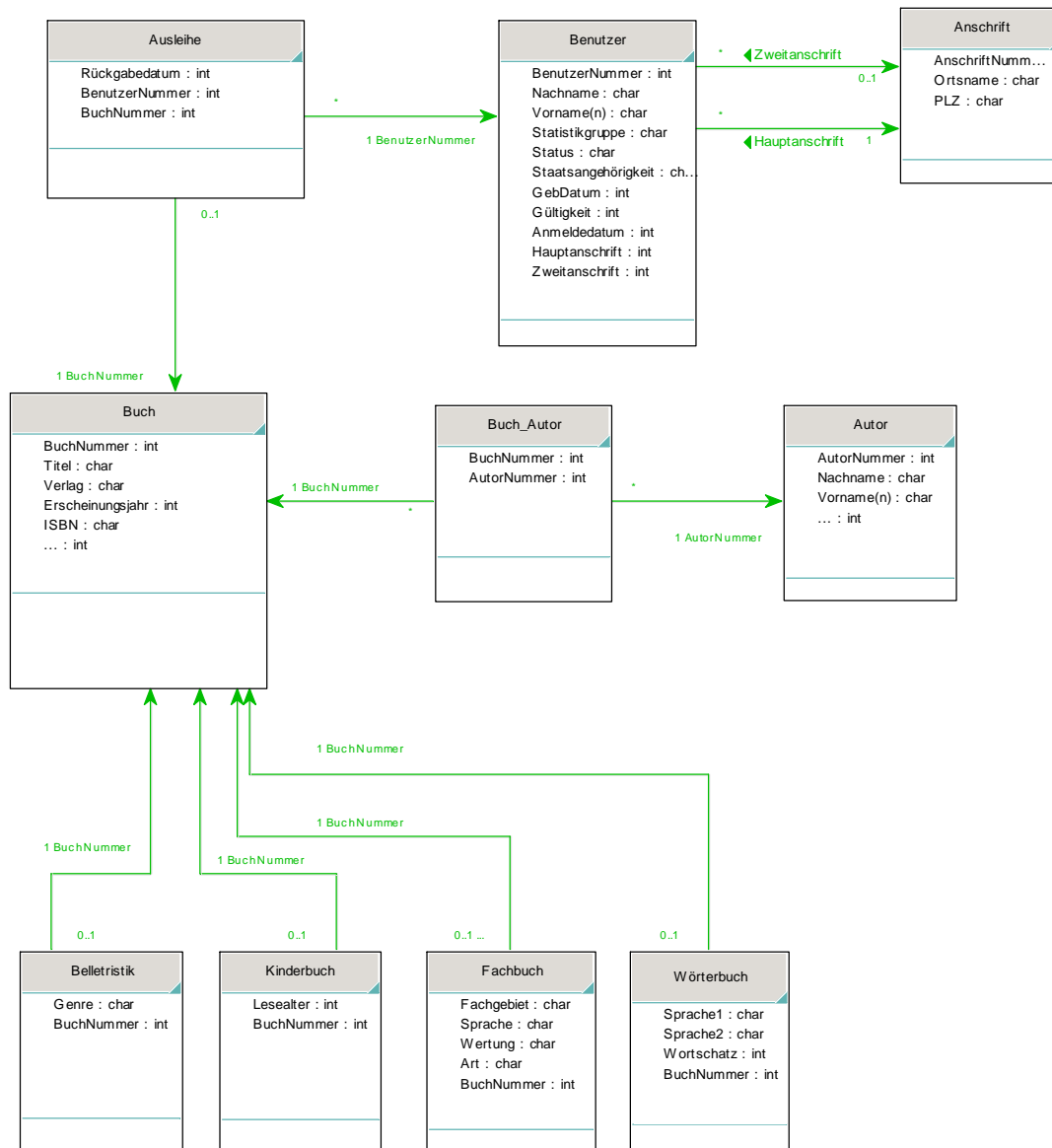
## Aggregation:

Wenn die Belegarbeitsgruppe nach dem dritten Semester nicht mehr existiert, existieren aber sehr wohl die Studenten noch. Sie studieren dann im vierten Semester.



## Das Klassendiagramm 3/3

## Klassendiagramm aus dem Relationenmodell erzeugt (case/4/0 → Datenmodell Bibliothek-Demo)



### Ein Dateisystem organisiert nach dem Entwurfsmuster Kompositum (Composite) nach Erich Gamma

Vgl. Entwurfsmuster - Elemente wiederverwendbarer objektorientierter Software von Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison-Wesley Deutschland, © 1996, ISBN 3-89319-950-0, S. 215.

