

# 10 Anforderungsspezifikation mit Objekt- oder Klassenmodellen

## 10.1 Grundidee

- Diejenigen **Elemente des Anwendungsbereichs** identifizieren, von denen das zu spezifizierende System Informationen kennen und verarbeiten muss
- Diese Elemente auf **Objekte**, deren **Eigenschaften** und **Beziehungen** abbilden
- Die Anforderungen anhand des resultierenden **Strukturmodells** gliedern und darstellen
  
- + Gut geeignet zur Beschreibung der **Systemstruktur**
- + Unterstützt **Lokalität** von Daten und **Einkapselung** von Eigenschaften
- + Erlaubt **strukturähnliche** Implementierungen
- + Bei Objektmodellen ist **Systemdekomposition** möglich
- Funktionalität aus **Benutzersicht** schlecht modellierbar
- **Dekomposition** bei Klassenmodellen schwierig und meistens nicht unterstützt

## 10.2 Objekte, Klassen und abstrakte Objekte

**Objekt** (object) – Ein individuell erkennbares, von anderen Objekten eindeutig unterscheidbares Element der „Realität“, das heißt des betrachteten Problem- oder Lösungsbereichs.

Beispiel: Die konkrete Person Eva Müller, 36 Jahre alt, Dr. oec. publ., Leiterin Fertigung in der Firma AGP, verheiratet, ein Kind, ... wird als *Objekt* modelliert

- **Konkrete Objekte eignen sich nicht** zur Modellierung von Anforderungen
  - zu speziell
  - oft zum Zeitpunkt der Spezifikation gar nicht bekannt
- Abhilfe
  - Modellierung mit **Klassen** → Klassenmodelle
  - Modellierung mit **abstrakten Objekten** → Objektmodelle

**Klasse** (class) – eine eindeutig benannte Einheit, welche eine Menge gleichartiger Objekte beschreibt.

Beispiele: Student, Professor

**Abstraktes Objekt** (abstract object) – eine abstrakte Repräsentation eines einzelnen konkreten Objekts (Singularobjekt) oder einer Menge von konkreten Objekten (Objektmenge).

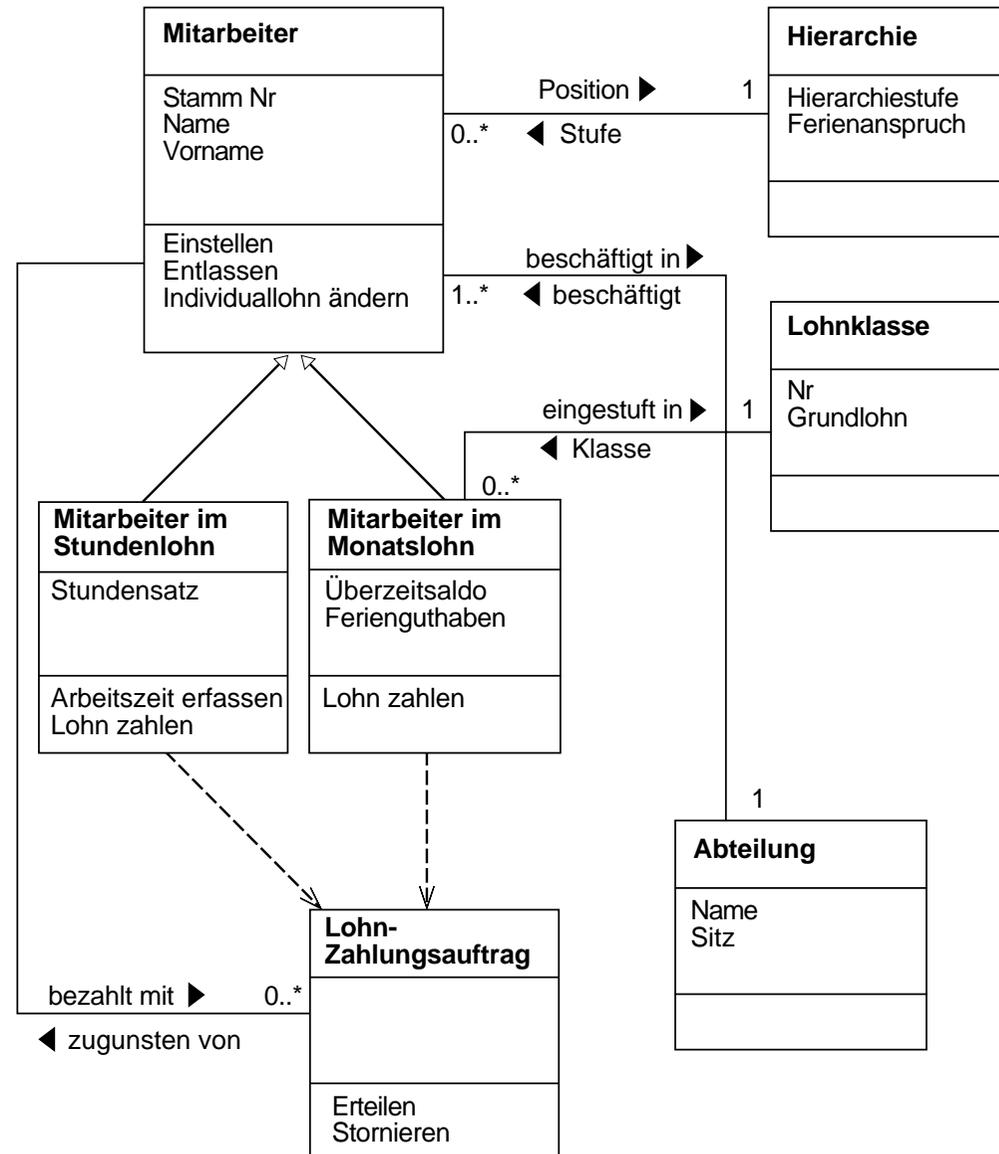
Beispiele: Dekan (Singularobjekt), Tutor (Objektmenge)

In UML werden abstrakte Objekte als Classifier Roles bezeichnet

Zu jeder Klasse (in Klassenmodellen) bzw. zu jedem abstrakten Objekt (in Objektmodellen) werden modelliert:

- lokale Eigenschaften als **Attribute**
- von den Objekten der Klasse bzw. vom abstrakten Objekt angebotene Dienstleistungen als **Operationen**
- Zusammenhänge zwischen Objekten durch **Beziehungen**

# Beispiel eines Klassenmodells



## 10.3 Objektanalyse

- ☆ Ein Verfahren zur **Gewinnung** von Objekt- und Klassenmodellen
- ☆ Hier für **Klassenmodelle** gezeigt
- ☆ Voraussetzung: **Text vorhanden** (schriftlich oder mündlich)
- ☆ Idee: **Analyse der grammatischen Struktur** der Sätze
- ☆ Problem: Liefert große Menge schwach strukturierter Kandidaten für Modellelemente;

# Durchführung der Objektanalyse

- Text analysieren:
  - grammatisches *Subjekt*, grammatische *Objekte* → Kandidaten für *Objekte*, *Klassen*, *Attributwerte*, *Attribute* oder *Wertebereiche*
  - *Verben* beschreiben Zusammenhänge oder Handlungen:
    - Zusammenhänge → *Beziehungen*, *Attribute*
    - Handlungen → *Funktionalität*, *Verhalten*
  - *Adjektive* präzisieren Aussagen oder *schränken* sie *ein*
- Fragmente *klassifizieren*, *ordnen*, *vervollständigen*
- *Abgrenzung* von Klassen/Objekten gegen Attribute/ Werte:
  - Jedes Objekt muss eine Identität haben
  - Attributwerte sind Daten ohne eigene Identität
  - Attribute von Attributen werden in der Regel vermieden: In solchen Situationen Klassen und Beziehungen modellieren

# Beispiel: Modellierung einer Bibliothek

## Problembeschreibung

Die Bibliothek bewirtschaftet eine Menge von Büchern. Ein Buch wird beschafft und von einer Bibliothekarin klassifiziert. Es erhält eine Signatur und wird einem oder mehreren Sachgebieten zugeordnet. Jedes Sachgebiet hat eine Nummer und einen Namen.

Gewisse Bücher dürfen nur im Lesesaal eingesehen werden; die meisten aber sind ausleihbar. Jedes ausleihbare Buch kann durch einen Bibliotheksbenutzer ausgeliehen werden. Ein ausgeliehenes Buch muss spätestens nach drei Wochen zurückgegeben werden. Die Ausleihfrist kann verlängert werden, wenn keine andere Benutzerin sich in der Zwischenzeit hat vormerken lassen. Bei Rückgabe wird die vorgemerkte Benutzerin benachrichtigt. Wer die Ausleihfrist überschreitet, wird gemahnt.

Nicht mehr benötigte oder abhanden gekommene Bücher werden aus dem Bestand entfernt. Nur die in einer Benutzerdatei mit Name und Anschrift registrierten Benutzerinnen und Benutzer dürfen Bücher ausleihen.

Die Bibliothek führt einen Katalog. Jedes Buch hat einen Katalogeintrag mit Autor(en), Titel, Buchsignatur und bis zu 5 Stichworten. Der Katalog kann nach Autoren, Titeln oder Stichworten durchsucht werden.

# Durchführen der Objektanalyse

## **Objekt/Klassenkandidaten:**

Bibliothek, Buch, BibliothekarIn, Signatur, Sachgebiet, Sachgebietsnummer, Sachgebietsname, Lesesaal, ausleihbares Buch, (Bibliotheks-)BenutzerIn, Ausleihfrist, Bestand, Benutzerdatei, Name, Anschrift, Katalog, Katalogeintrag, Autor, Titel, Buchsignatur, Stichwort

## Durch Ordnen und Klassifizieren entsteht:

<b>KLASSE</b>	<b>ATTRIBUT</b>	<b>BEMERKUNGEN</b>
Bibliothek	Ausleihfrist	Aggregation von Buch, BibliothekarIn, Katalog
Buch	Signatur	
AusleihbaresBuch		Unterklasse von Buch
LesesaalBuch		Unterklasse von Buch
BibliothekarIn		
Sachgebiet	Sachgebietsnummer, Name	
BenutzerIn	Name, Anschrift	
Katalog		Aggregation von Katalogeintrag
Katalogeintrag	Autor, Titel, Buchsignatur, Stichwort	

**Entfallen sind:** Bestand (als Facette von Bibliothek aufgefasst), Benutzerdatei (physische Repräsentation von BenutzerIn), Lesesaal (als reines Klassifikationskriterium für Bücher aufgefasst)

## Beziehungen bestehen:

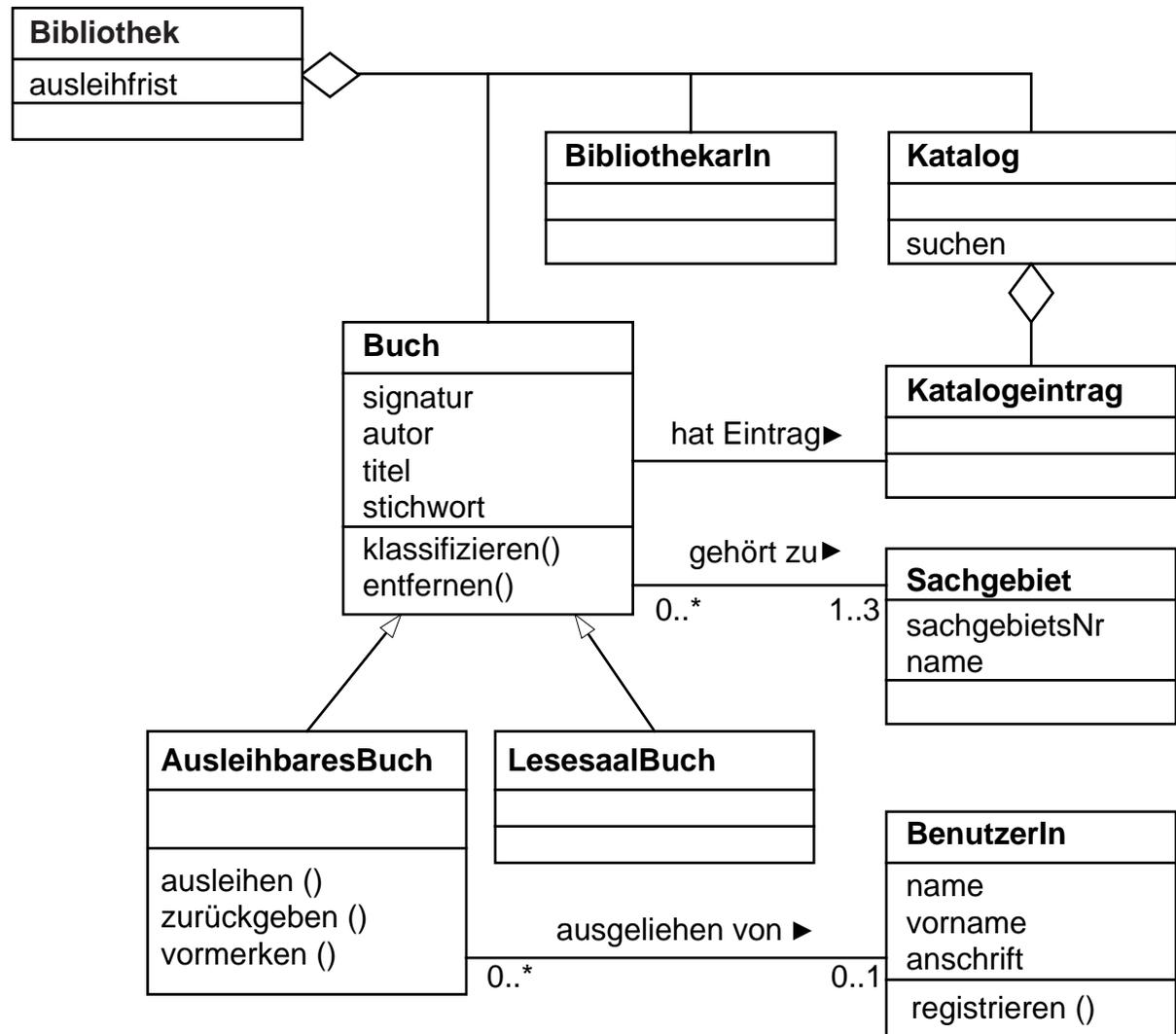
- zwischen Buch und BenutzerIn («kann ... ausgeliehen werden»)
- zwischen Buch und Sachgebiet («wird ... zugeordnet»)
- zwischen Katalogeintrag und Buch («hat»)

## Ferner können folgende Operationen erkannt werden:

<b>Klasse</b>	<b>Operation</b>
Buch	Klassifizieren Entfernen
AusleihbaresBuch	Ausleihen Zurückgeben Vormerken
BenutzerIn	Registrieren
Katalog	Suchen

# Erster Modellentwurf (UML-Klassenmodell)

- Alle Buchattribute sind in der Klasse Buch konzentriert.  
Die Klassen Katalog und Katalogeintrag werden als Zugriffsklassen aufgefasst.
- Noch offen ist, ob die Klasse BibliothekarIn als Zugriffsklasse im Modell verbleibt oder ob sie schließlich als irrelevant betrachtet und entfernt wird.



## 10.4 Ereignis-Reaktions-Analyse

- Alle **Ereignisse**, die eine Reaktion des Systems erfordern, auflisten
- Für jedes Ereignis die erforderlichen **Reaktionen** bestimmen
- **Verhalten** und **Operationen** bestimmen durch
  - Feststellen, welche **Operationen** auf Objekten welcher Klassen erforderlich sind, um die geforderten Reaktionen zu erzeugen
  - Beschreiben der Operationen durch Angabe ihrer **Voraussetzungen** und Resultate (**Ergebniszusicherung**)
  - Bestimmen der **Zustände**, in denen Operationen erlaubt/zulässig sind und der **Zustandsübergänge**, die mit der Operationsausführung verbunden sind
  - Beschreiben des Verhaltens von Objekten z. B. mit **Zustandsdiagrammen**

- **Interaktionen** bestimmen durch
  - Gruppierung von Ereignissen, die vom gleichen externen Akteur stammen, zu logischen Sequenzen von Ereignissen und Systemreaktionen
  - Beschreiben solcher Sequenzen in **Szenarien** → Kapitel 11
  
- **Klassen/Attribute/Beziehungen** bestimmen durch
  - Feststellen, welche Daten
    - (a) für die Erzeugung der Reaktion notwendig sind, aber nicht mit dem Ereignis mitgeliefert werden
    - (b) mit dem Ereignis mitgeliefert werden, aber erst später für eine Reaktion benötigt werden
  - Diese Daten als **Attribute** oder **Beziehungen** in geeigneten **Klassen** modellieren

# Beispiel: Ereignis-Reaktions-Analyse für ausleihbare Bücher (Auszug)

## A. Übersicht

<b>Nr Ereignis</b>	<b>Reaktion</b>
1 neues Buch trifft ein	Buch klassifizieren
2 BenutzerIn will Buch ausleihen	Buch nicht ausgeliehen und nicht vorgemerkt: ausleihen sonst: vormerken
3 BenutzerIn will verlängern	Buch vorgemerkt: verweigern, sonst: verlängern
4 BenutzerIn gibt Buch zurück	zurückgeben, falls vorgemerkt: VormerkerIn benachrichtigen
5 Ausleihfrist überschritten	AusleiherIn mahnen

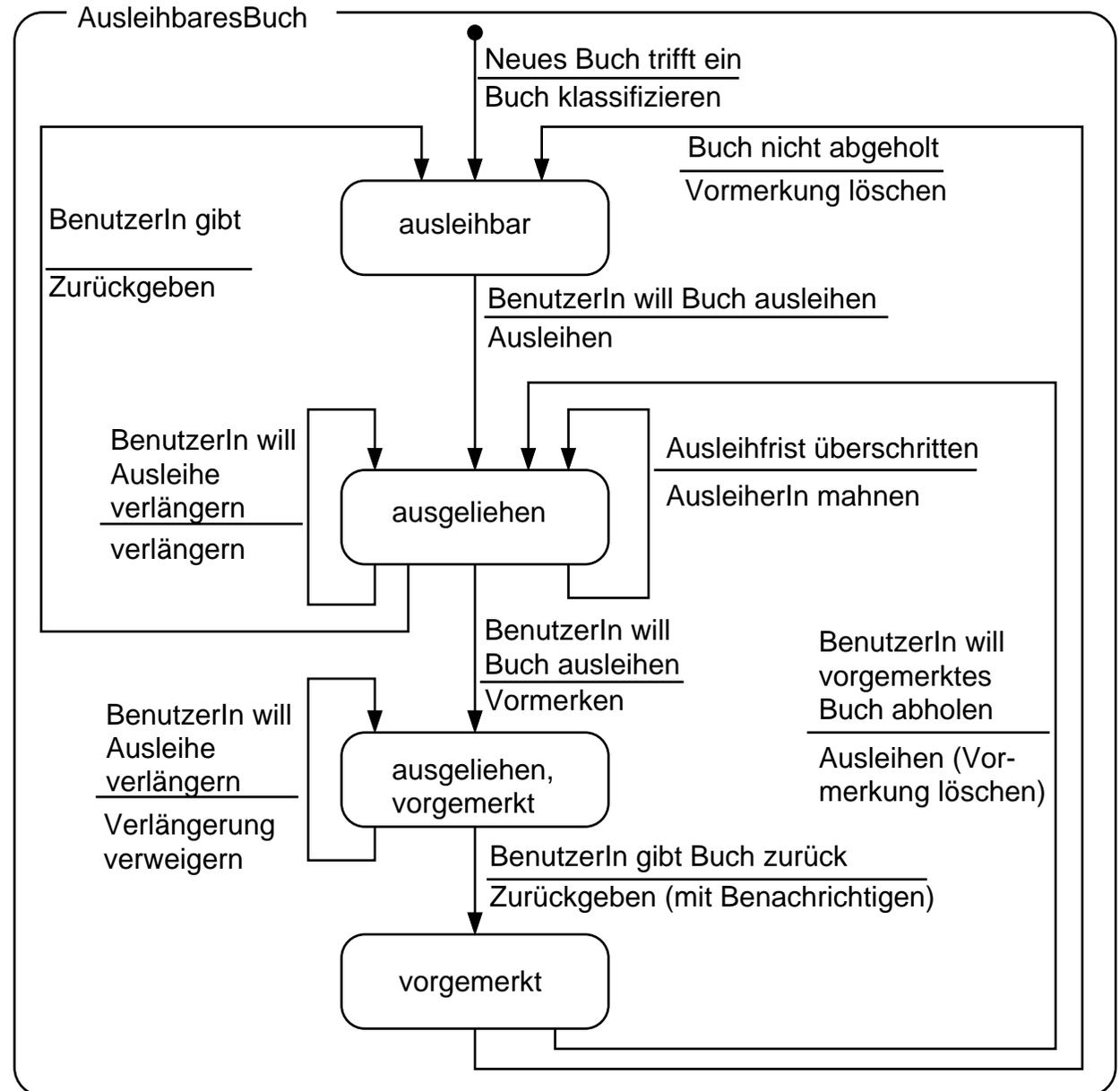
## B. Details von Schritt 2

- |                                     |  |
|-------------------------------------|--|
| 2.1 BenutzerIn liest ihre Karte ein | Karte prüfen und validieren  |
| 2.2 BenutzerIn wählt „Ausleihen“    | Ausleihe aktivieren  |
| 2.3 Benutzerin liest Buchcode ein   | Buch identifizieren, Ausleihe registrieren,<br>Diebstahletikett deaktivieren |
| 2.4 Benutzerin beendet Ausleihe     | Leihschein drucken   |

# Dynamisches Verhalten von Objekten der Klasse AusleihbaresBuch

(modelliert mit einem Zustandsdiagramm):

Hinweis: In diesem Modell gibt es Probleme. Analysieren Sie das Modell und identifizieren Sie die Problemstellen.



# Beschreibung der Operation Zurückgeben

AusleihbaresBuch.zurückgeben ()

## **Voraussetzung**

Zustand des Buchs ist ausgeliehen oder ausgeliehenVorgemerkt

## **Ergebniszusicherung**

Ausleihbeziehung zu BenutzerIn ist gelöscht,  
neuer Zustand ist ausleihbar oder vorgemerkt,  
vorgemerkte Benutzerin ist benachrichtigt.