

Analysemuster

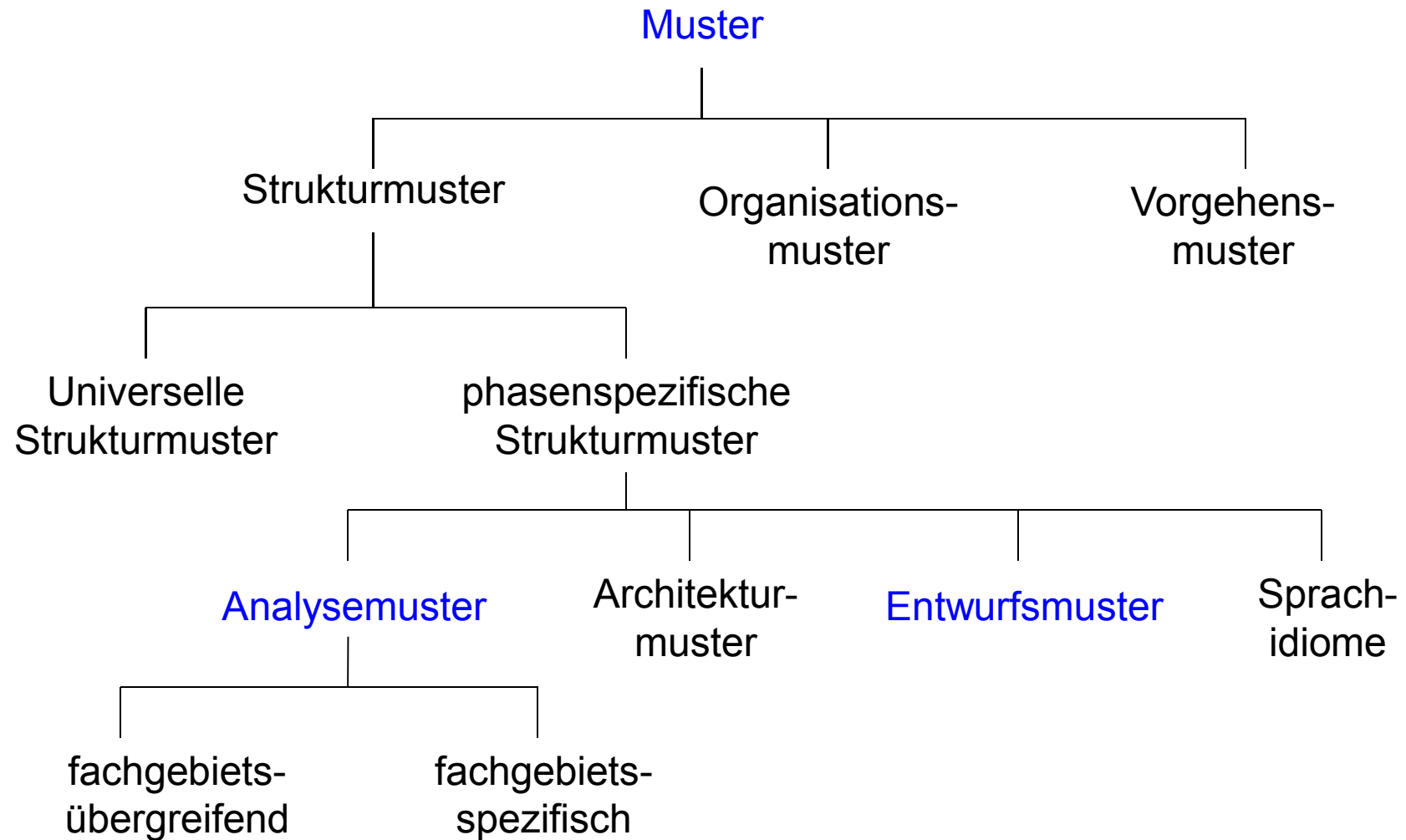
(Analysis Pattern)

(Quelle: H. Balzert u.a.)

Was sind Muster?

- „Ein Muster ist eine Idee, die sich in einem praktischen Kontext als nützlich erwiesen hat und es auch in anderen sein wird.“ (M. Fowler)
- „Ein Analysemuster ist eine Gruppe von Klassen mit feststehenden Verantwortlichkeiten und Interaktionen.“ (P. Coad)
- Es kann eine Gruppe von Klassen sein, die durch Beziehungen verknüpft ist, oder eine Gruppe von kommunizierenden Objekten.
- ➔ schematische Lösung für eine Klasse verwandter Probleme
- ➔ Muster zur Modellierung häufig vorkommender Strukturen

Klassifikation von Mustern



Welche Muster werden hier behandelt?

1. Liste
2. Exemplartyp
3. Baugruppe
4. Stückliste
5. Koordinator
6. Rollen
7. Wechselnde Rollen
8. Historie
9. Gruppe

Muster 1: Liste

Beispiel: Bestellung

Bestellnummer:

167334

Datum:

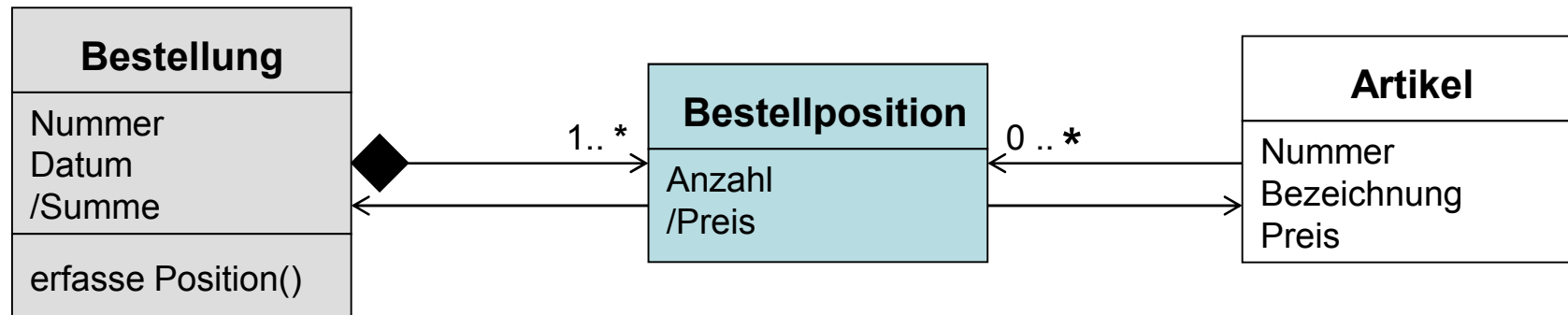
01.06.2004

Nr.	Bezeichnung	Einzelpreis	Anzahl	Preis
47	Kugelschreiber	1,20	100	120,00
11	Folienstift	2,50	5	12,50
13	...			

Summe:

132,50

Muster 1: Liste

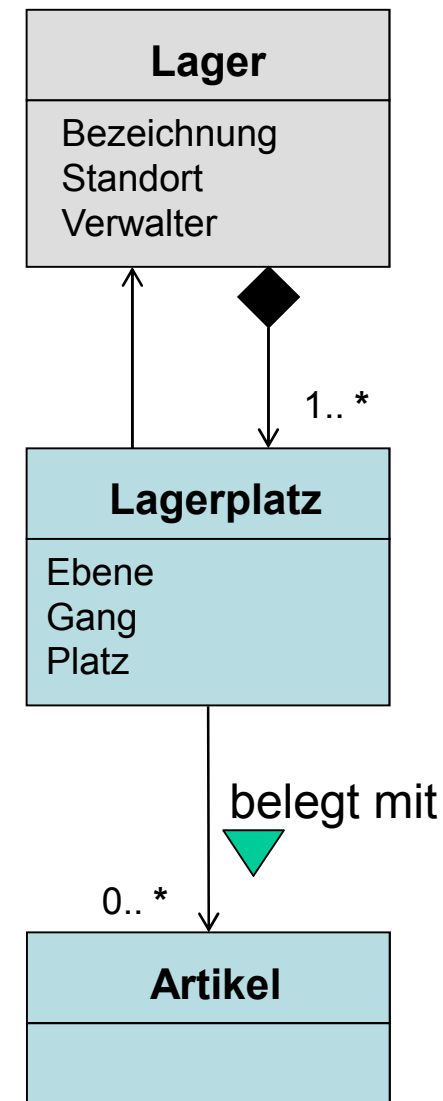
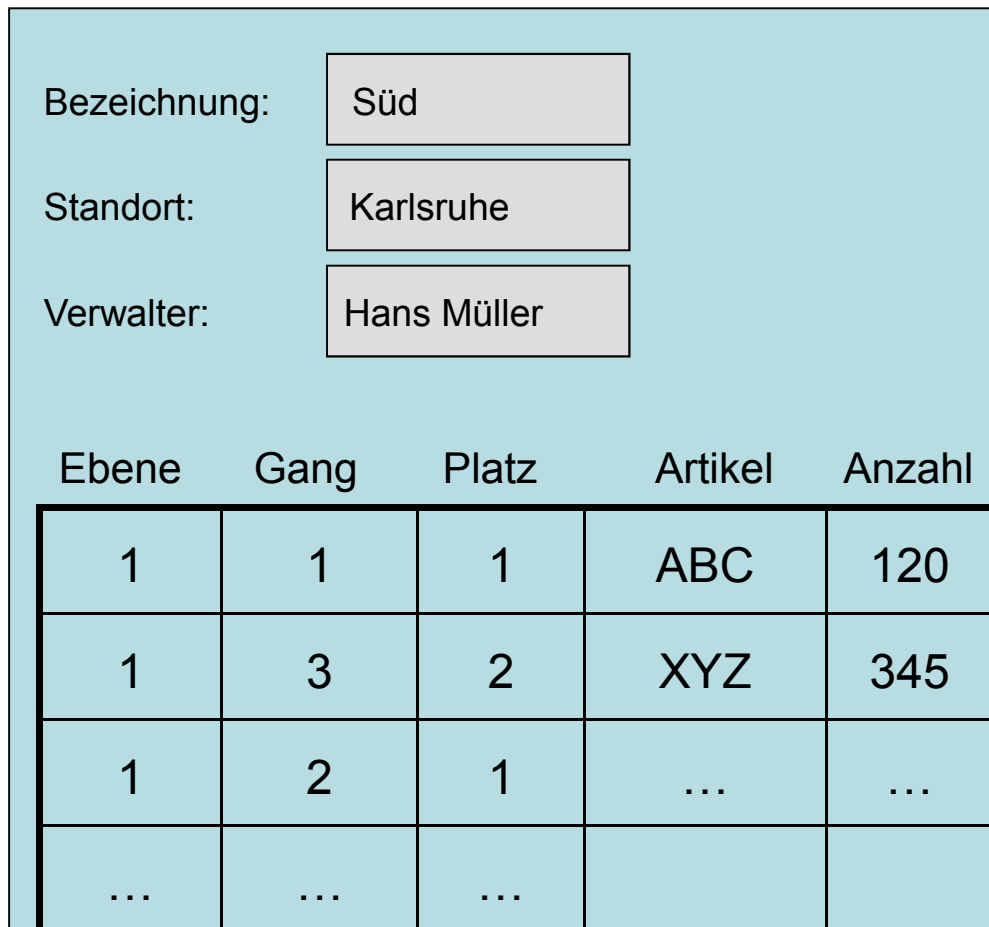


Eigenschaften:

- ➔ **Komposition** (Listenelemente in genau einer Kollektion, nicht allein existent)
- Ein Ganzes besteht aus **gleichartigen** Teilen, d.h. es gibt nur **eine** Teil-Klasse
- Teil-Objekte bleiben einem Aggregat-Objekt fest zugeordnet (können jedoch einzeln gelöscht werden)
- Attributwerte des Aggregat-Objekts gelten auch für die Teile (z.B. *Nummer*)
- Das Aggregat-Objekt enthält im allgemeinen mindestens ein Teil-Objekt, d.h. die Multiplizität ist meist **1..***

Muster 1: Liste

Beispiel 2: Lager



Muster 2: Exemplartyp

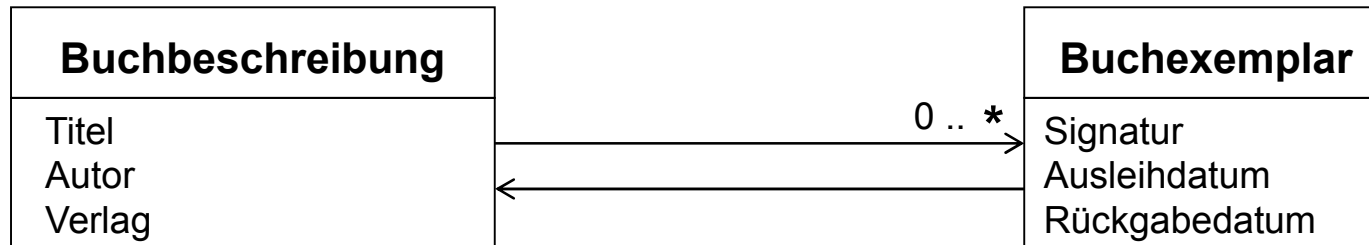
Problembeschreibung:

- Es gibt Gruppen von Objekten, deren Eigenschaften im wesentlichen identisch sind, sodass man sie als Kopien voneinander oder als Exemplare des selben Typs ansehen kann.
- Eigenschaften des Typs (Werte seiner Attribute) sind auch Eigenschaften der Exemplare dieses Typs.
- Die Exemplare haben eine Seriennummer oder Inventarnummer, durch die sie unterscheidbar sind (werden i.d.R. nie mehr verändert), ansonsten nur sehr wenige weitere Eigenschaften, die nicht schon durch den Typ gegeben sind.

Beispiel: Von einem Buch sind mehrere Exemplare zu verwalten

- Bei Modellierung einer einzigen Klasse **Buch**:
identische Attributwerte bei mehreren Objekte bei *Titel*, *Autor* und *Verlag*
- **Bessere Modellierung:**
Zusammenfassung der gemeinsamen Attributwerte mehrerer Buchexemplare in einer neuen Klasse **Buchbeschreibung**

Muster 2: Exemplartyp



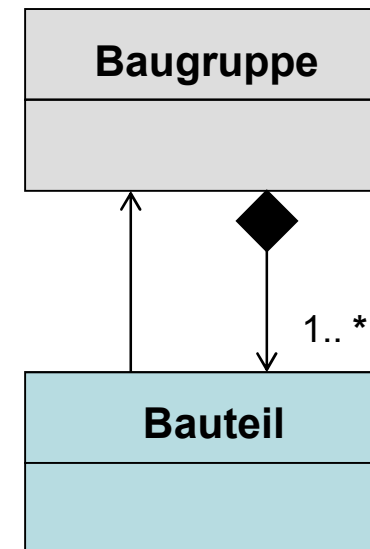
Eigenschaften:

- ➔ **einfache Assoziation** (keine sog. *whole-part*-Beziehung)
- Erstellte Objektverbindungen werden nicht verändert. Sie werden nur gelöscht, wenn das betreffende Exemplar gelöscht wird
- Der Name der neuen Klasse enthält oft Begriffe wie *Typ*, *Gruppe*, *Beschreibung*, *Spezifikation*
- Eine Beschreibung kann unabhängig von konkreten Exemplaren existieren. Daher ist die Multiplizität meist **0..***
- Würde auf die neue Klasse verzichtet, so würde als Nachteil lediglich die redundante „Speicherung“ von Attributwerten auftreten

Muster 3: Baugruppe

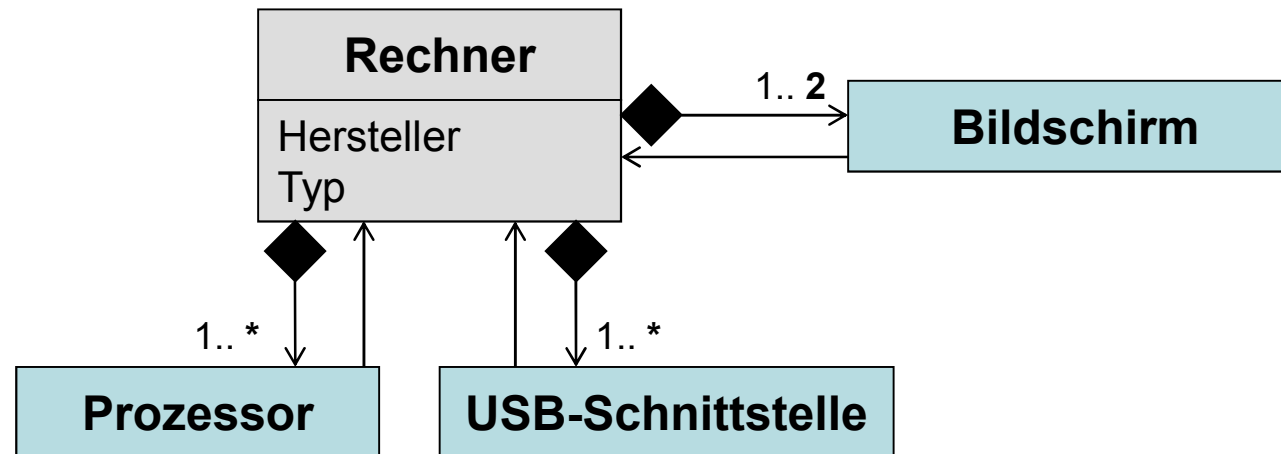
Problembeschreibung:

- Gliederung des Ganzen in viele Teile
- Existenzabhängige Zugehörigkeit der Teile
- „Zugang“ zu Teilen nur über das Ganze
- Teile können **unterschiedlichen** Typs sein
- Objektverbindungen bestehen meist über längeren Zeitraum
- Trennen der Objekte möglich und an andere Baugruppen anschließbar



Muster 3: Baugruppe

Beispiel:



Weitere Beispiele:

- Schaltkreis – Bauteile
- Festplatte – Bauteile
- Auto – Räder (Handelt es sich dabei um eine Baugruppe?)
- Wie sieht es bei einem Laptop aus?

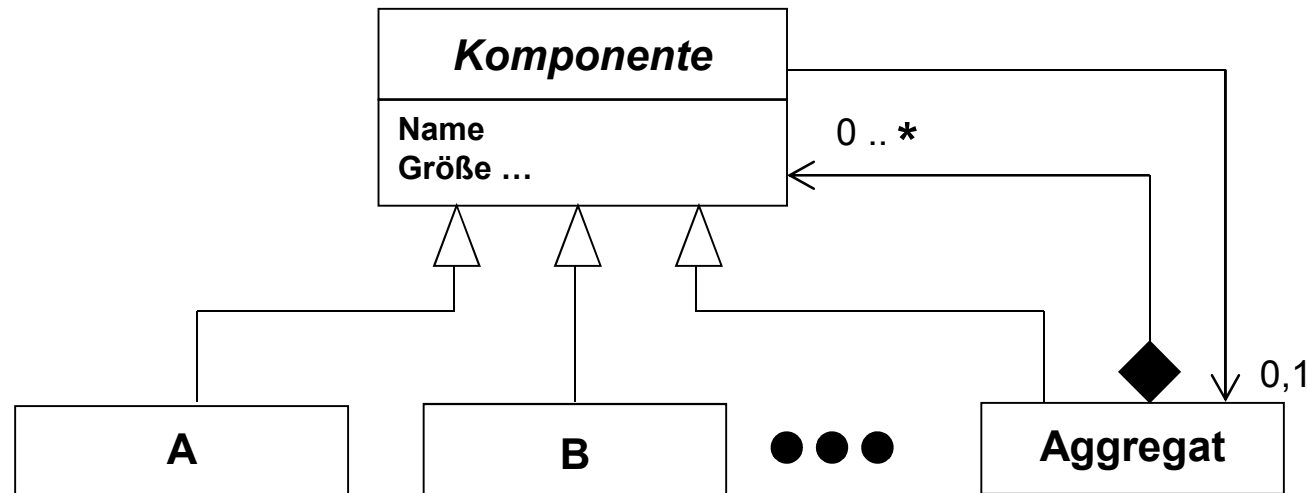
Muster 4: Stückliste (Kompositum)

Problembeschreibung:

- Modellierung einer Teil/Ganzes-Hierarchie
- unbekannte / ungleichmäßige / zeitlich variierende Schachtelungstiefe
- die (Teil-von-)Struktur ist ein Baum, die atomaren Bestandteile bilden die Blätter, die unterschiedlichen Typs sein können
- alle Knoten (Blätter) des Baums weisen einheitliche Merkmale bzw. Verhaltensweisen auf
- die zusammengesetzten Teile müssen oft genauso wie atomare Bestandteile behandelt werden (lokale Namen, werden komplett gelöscht oder kopiert)
- Teil-Objekte können einem Aggregat-Objekt zugeordnet werden

Muster 4: Stückliste

Lösung:



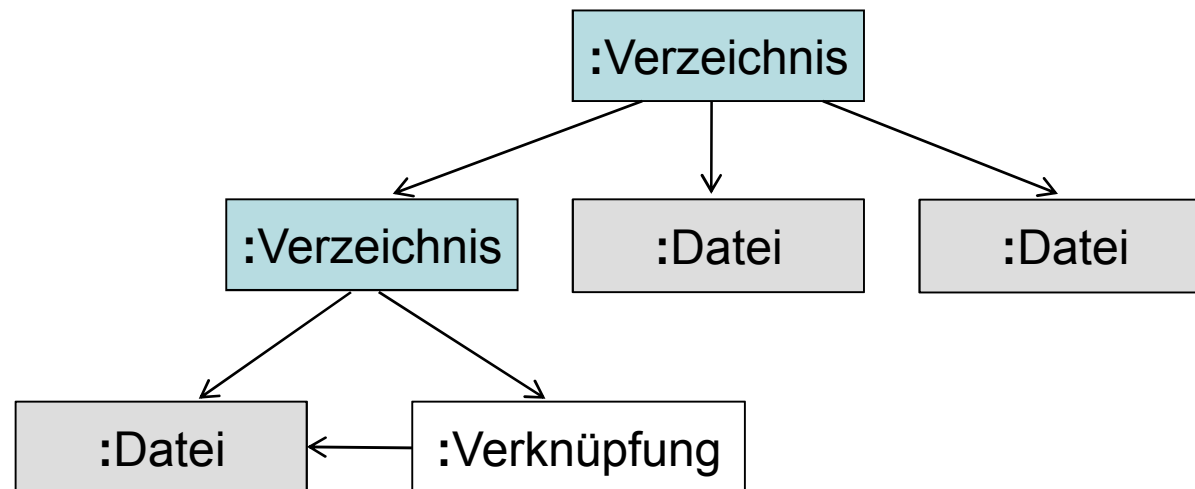
- ➔ **Komposition** (Multiplizität bei der Aggregatklasse ist **0,1**)
- eine abstrakte Oberklasse, die die gemeinsamen Eigenschaften (und Operationen) von atomaren und zusammengesetzten Komponenten modelliert
- je eine Klasse für jeden konkreten Typ von atomaren Bestandteilen (Blätter der Teil-von-Struktur)
- Genau eine Klasse für die zusammengesetzten Bestandteile (d.h. die inneren Knoten der Teil-von-Struktur)

Muster 4: Stückliste

Beispiel: Verzeichnisbaum

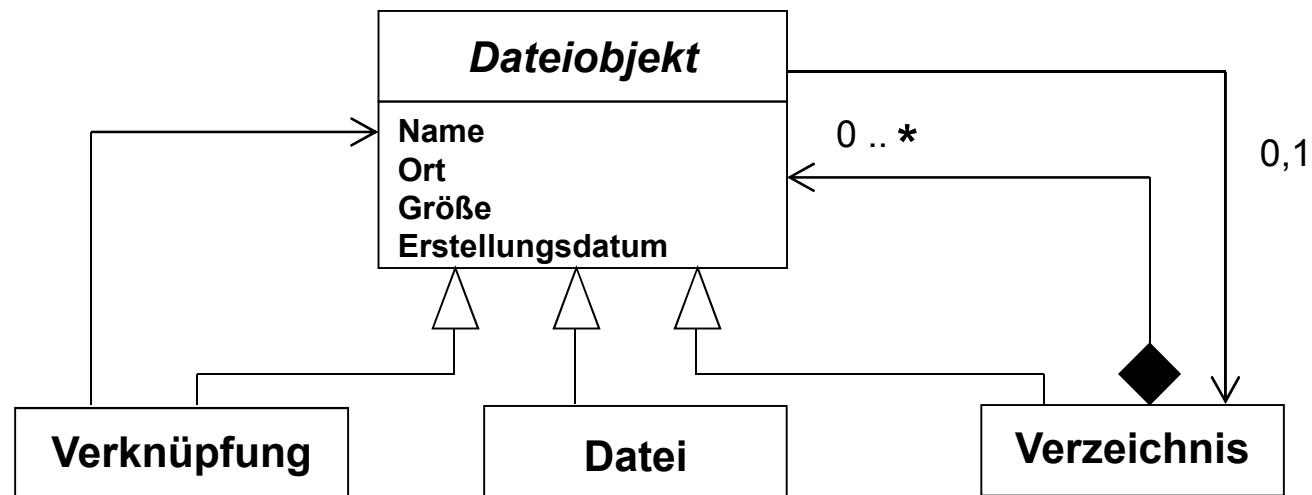
- Ein Verzeichnis soll Verknüpfungen, Dateien und weitere Verzeichnisse enthalten können
- Sowohl das Verzeichnis und alle darin enthaltenen Objekte als Einheit als auch jedes dieser Objekte sollen einzeln behandelt werden können (z.B. für rekursives Kopieren und Löschen)

Hierarchiemöglichkeit (Instanzendiagramm):



Muster 4: Stückliste

Verzeichnisbaum, Lösung:



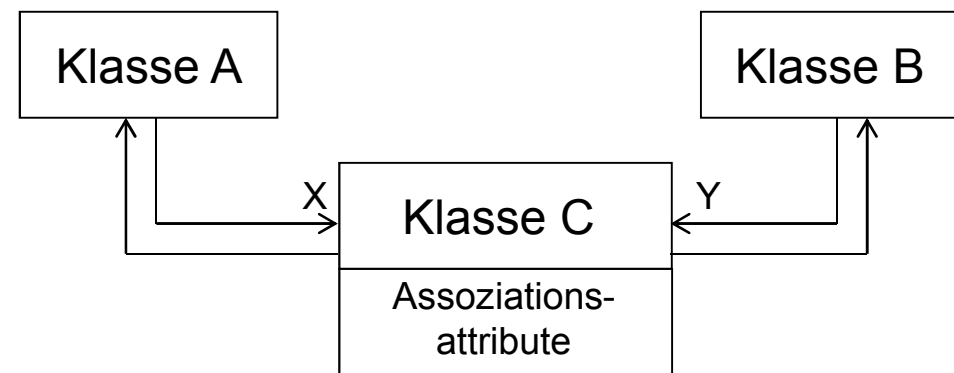
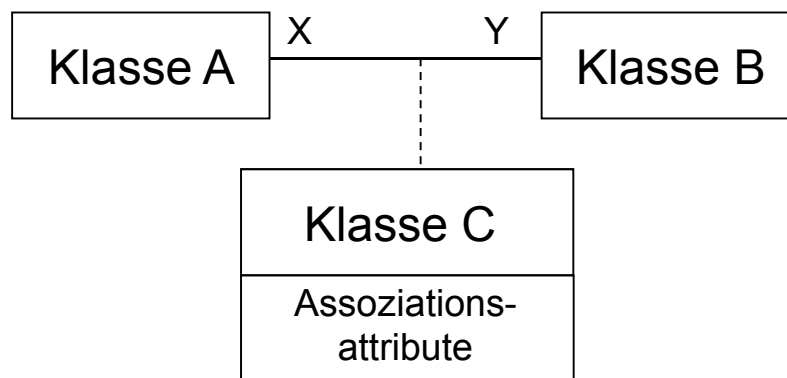
Muster 5: Koordinator

Problembeschreibung:

- Für eine **Assoziation** sind **zusätzliche Attribute** zu speichern, die zu keiner der beteiligten Klassen gehören.

Lösungsmöglichkeit:

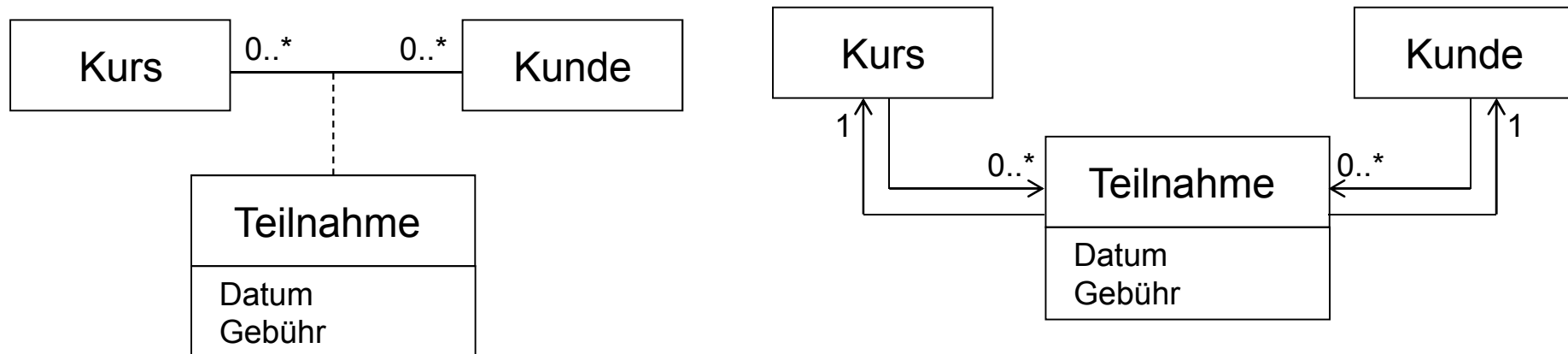
- Verwendung der so genannten "Assoziationsklassen" von UML
- Einführung einer eigenen Klasse für die Assoziation



Muster 5: Koordinator

Beispiel:

- Ein Kunde nimmt an mehreren Kursen teil. Die Kurse selbst werden von mehreren Kunden gebucht..



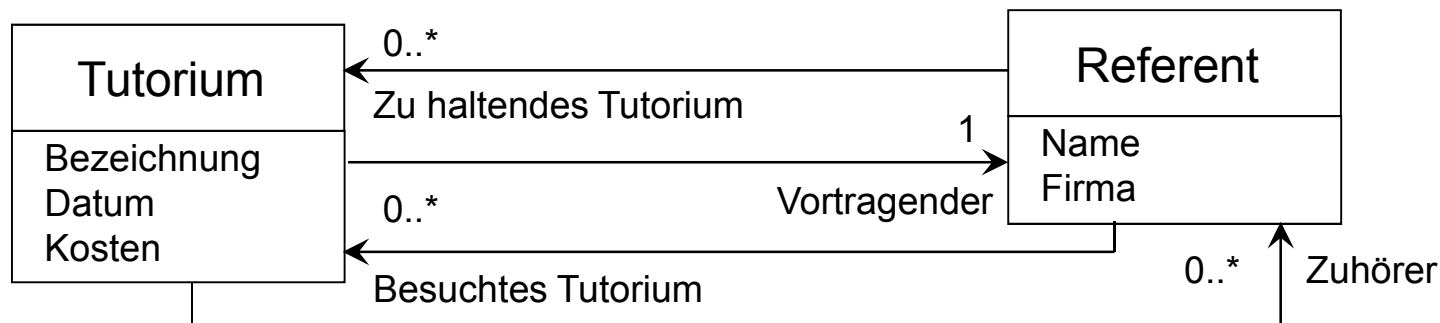
- Es liegen einfache Assoziationen vor
- Die Koordinator-Klasse besitzt kaum Attribute/Operationen, sondern mehrere Assoziationen zu anderen Klassen, i. a. zu genau einem Objekt jeder Klasse
- Beim Koordinator sind **Beziehungen** wichtig, nicht Attribute

Muster 6: Rollen

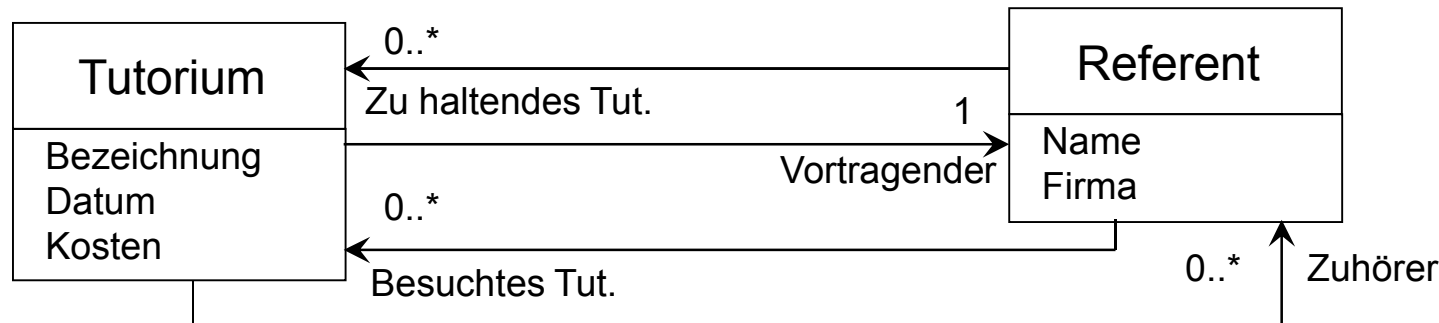
Rollen werden verwendet, wenn eine Klasse in Bezug auf eine andere Klasse unterschiedliche Rollen spielen kann.

Beispiel:

Zu einem Tutorium sind Vortragende und Zuhörer zu verwalten. Dabei kann ein Referent sowohl Vortragender als auch Zuhörer von Tutorien sein.



Muster 6: Rollen



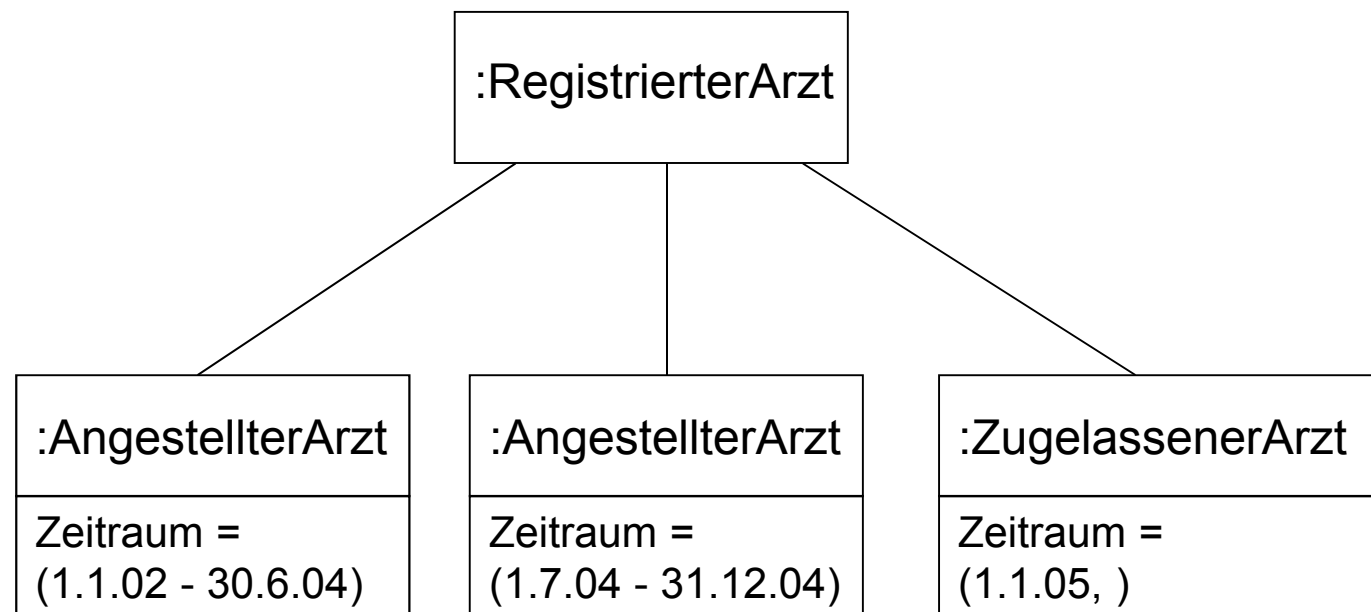
Eigenschaften:

- Zwischen zwei Klassen existieren zwei oder mehrere einfache Assoziationen.
- Ein Objekt kann – zu einem Zeitpunkt – in Bezug auf die Objekte (Instanzen) der anderen Klasse verschiedene Rollen spielen.
- Objekte, die verschiedene Rollen spielen können, besitzen unabhängig von der jeweiligen Rolle die gleichen Eigenschaften und gleiche Operationen.

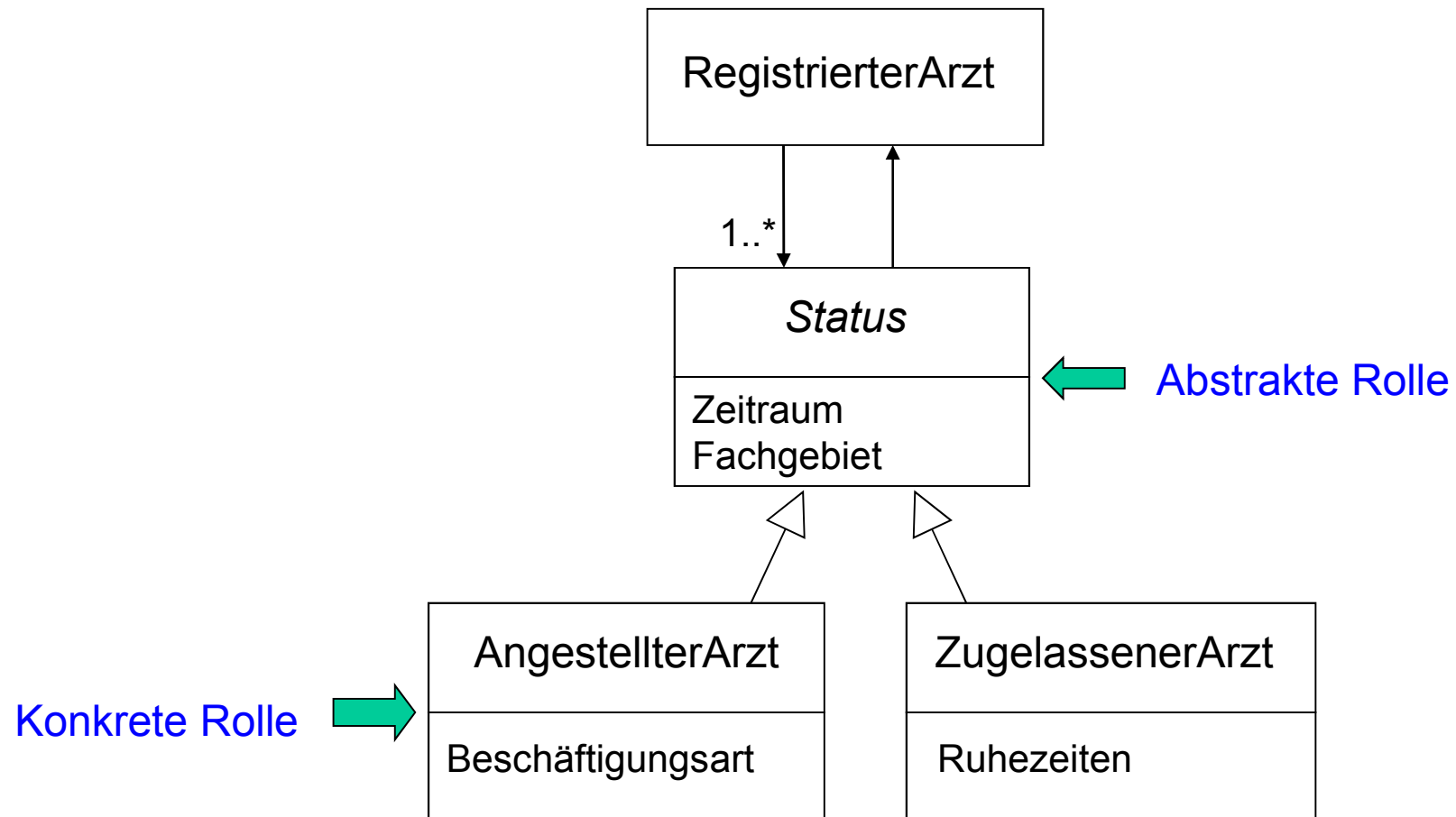
Muster 7: Wechselnde Rollen

Beispiel:

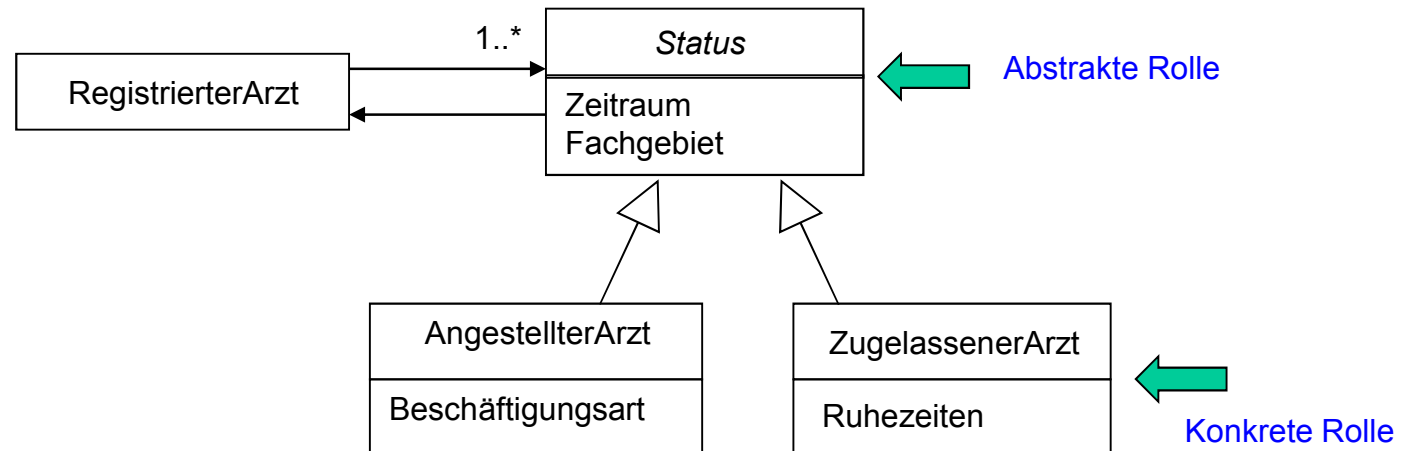
Ein kassenärztlich registrierter Arzt übt zu verschiedenen Zeiten und Praxen eine Tätigkeit als angestellter Arzt aus, bevor er seine Zulassung erhält.



Muster 7: Wechselnde Rollen



Muster 7: Wechselnde Rollen

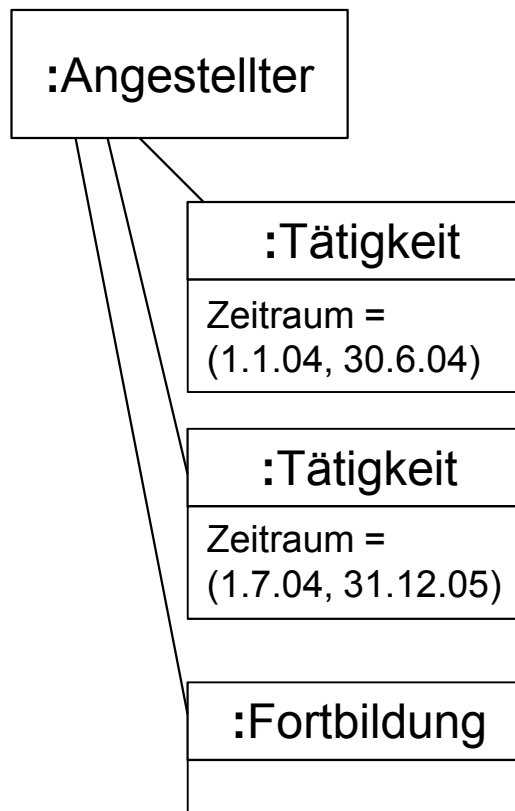


Eigenschaften:

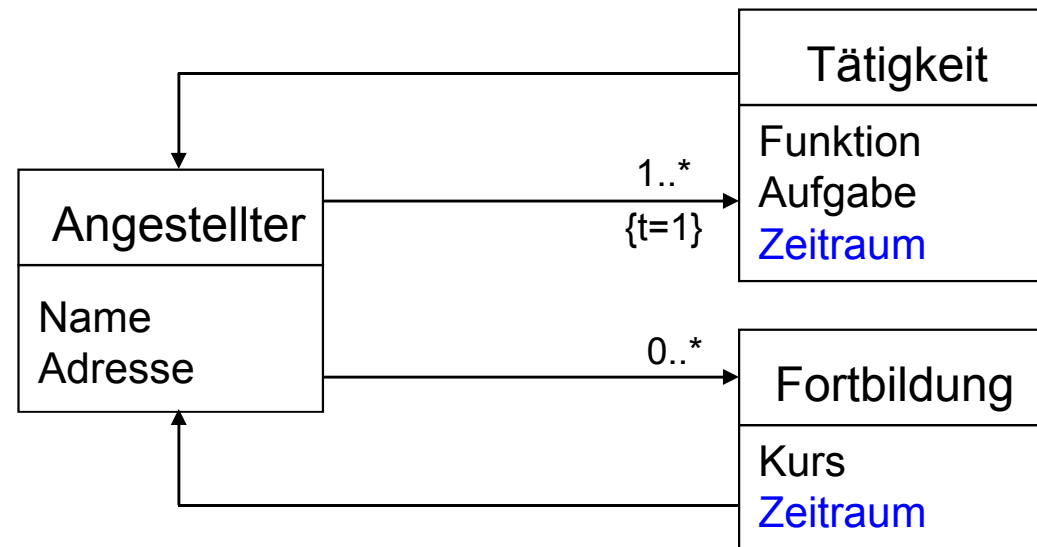
- Ein Objekt der realen Welt kann zu verschiedenen Zeiten verschiedene Rollen spielen. In jeder Rolle kann es unterschiedliche Eigenschaften und Operationen besitzen.
- Die unterschiedlichen Rollen werden mittels Vererbung modelliert.
- Objektverbindungen zwischen dem Objekt und seine Rollen werden nur erweitert, d.h. weder gelöscht noch zu anderen Objekten aufgebaut.

Muster 8: Historie

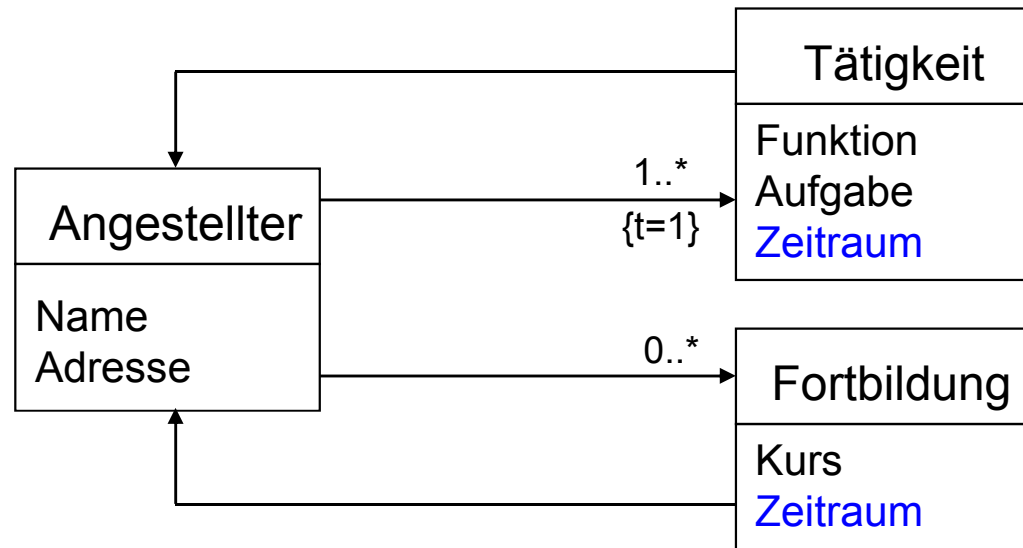
Instanzendiagramm:



Klassendiagramm (Muster):



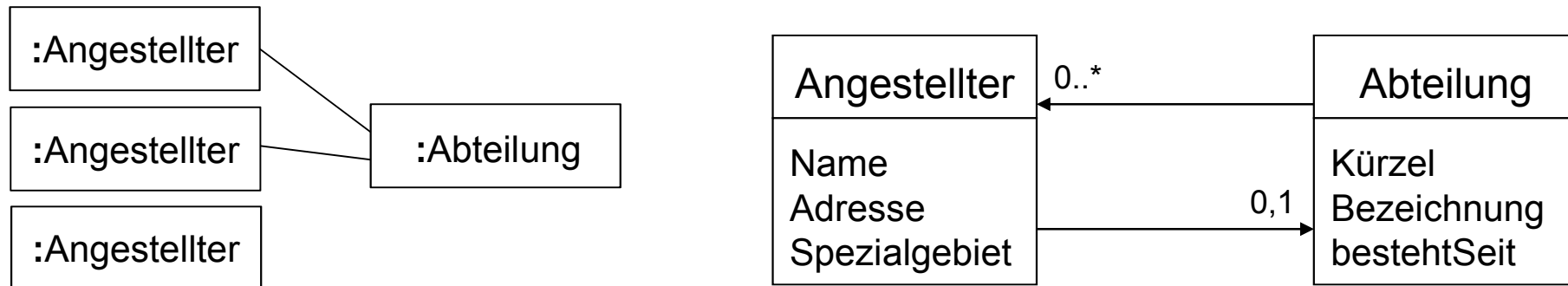
Muster 8: Historie



Eigenschaften:

- Einfache Assoziation
- Die zeitliche Restriktion $\{t = k\}$ (k = gültige Kardinalität bzw. Multiplizität, hier $k=1$) sagt aus, was zu einem Zeitpunkt gelten muss
- Wenn alle Tätigkeiten u. Fortbildungen gespeichert sein sollen:
Die Verbindungen zwischen Angestellter und Tätigkeiten bzw. Fortbildungen bleiben solange bestehen, bis der Mitarbeiter die Firma verlässt und seine Daten gelöscht werden.

Muster 9: Gruppe



Eigenschaften:

- Mehrere Einzelobjekte gehören - **zu einem Zeitpunkt** - zum selben Gruppenobjekt. Zeitweise kann die Zahl der Einzelobjekte auf Null sinken.
- Jedes Einzelobjekt kann *höchstens in einer Gruppe* Mitglied sein. Ein Einzelobjekt ist ggf. keiner Gruppe zugeordnet.
- Es ist jeweils zu prüfen, ob die Gruppe - zeitweise - ohne Einzelobjekte existieren kann oder ob sie immer eine Mindestanzahl von Einzelobjekten enthalten muss (**1..*** oder **0..*** bzw. **0,1** oder **1** als Multiplizität).
- Objektverbindungen können auf- und abgebaut werden.
- ➔ **Einfache Assoziation**

Erstellen Sie jeweils ein Klassendiagramm für die folgenden Problemstellungen:

1. Mehrere Personen schließen sich zu einer Fahrgemeinschaft zusammen.
2. Ein Projektplan besteht aus mehreren Planungsschritten.
3. In einem Grafiksystem bilden Kreise und Rechtecke eine Gruppe. Diese Gruppe kann wiederum Teil einer anderen Gruppe sein.
4. Bei mehreren Videokassetten in einer Videothek handelt es sich um den gleichen Film.

Erstellen Sie jeweils ein Klassendiagramm für die folgenden Problemstellungen:

1. Ein Mitarbeiter tritt als Programmierer in ein Unternehmen ein. Nach ein paar Jahren wird er als Manager tätig und steigt später zum Geschäftsführer auf. Für Programmierer, Manager und Geschäftsführer sind unterschiedliche Eigenschaften festzuhalten.
2. In einem Sportverein sind Sportler zu verschiedenen Zeiten in unterschiedlichen Mannschaften aktiv.
3. Zu einem Inventarstück in einem Museum sollen – sofern die Daten vorhanden sind – der derzeitige Eigentümer, der Vorbesitzer, der Finder und/oder der Überbringer festgehalten werden, die jeweils die gleichen Eigenschaften besitzen. Eine Person kann z.B. sowohl Eigentümer als auch Finder sein.
4. Für Personen sollen die Wohnsitze der letzten 10 Jahre ermittelt werden können. Zu einem Zeitpunkt muss jede Person mindestens einen und kann höchstens zwei Wohnsitze besitzen.

Literatur

- Heide Balzert:
Lehrbuch der Objektmodellierung - Analyse und Entwurf;
Spektrum Akademischer Verlag - Heidelberg, Berlin, 1999;
ISBN 3-8274-0285-9
- Fowler, Martin: Analysis Patterns – Reusable Object Models;
Addison Wesley, 1996; ISBN 0-201-89542-0