

Grundbegriffe der Objektorientierung

- Objekt
- Merkmale
 - Zustand
 - Verhalten
 - Lebenszyklus
- Beziehungen zwischen Objekten
- Kategorisierung von Objekten

Grundbegriffe der Objektorientierung

- Objekt
- Merkmale
 - Zustand – Attribute
 - Verhalten – Methoden
 - Lebenszyklus – Konstruktoren
- Beziehungen zwischen Objekten – Assoziationen
- Kategorisierung von Objekten – Klassen

Objektorientierte Modellierung:

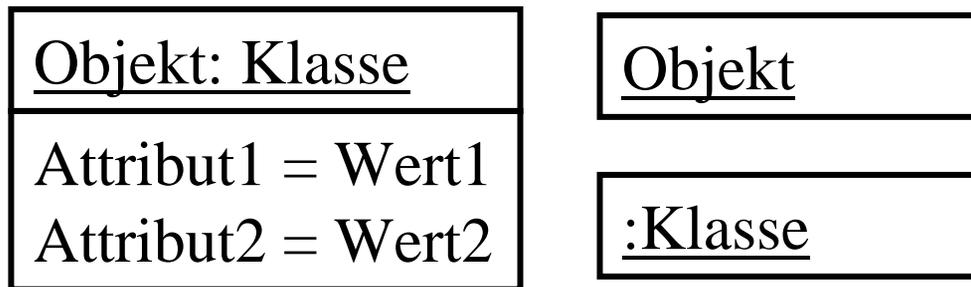
- Strukturelle Aspekte
- Dynamische Aspekte
- Funktionale Aspekte

Objektorientierte Modellierung: **Werkzeuge in UML**

- Strukturelle Aspekte
 - Klassendiagramme
 - Objektdiagramme
- Dynamische Aspekte
 - Sequenzdiagramme/Interaktionsdiagramme
 - Zustandsdiagramme
 - Aktivitätsdiagramme
- Funktionale Aspekte
 - Anwendungsfälle

Objekt (*object*)

- Ein Objekt besitzt einen bestimmten Zustand und reagiert mit einem definierten Verhalten auf seine Umgebung.
- Jedes Objekt besitzt eine Identität.



:Klasse

bei einem anonymen Objekt wird nur der Klassenname angegeben

Objekt:Klasse

wenn das Objekt über einen Namen angesprochen werden soll.

Objekt

wenn der Objektname ausreicht, um das Objekt zu identifizieren und der Name der Klasse aus dem Kontext ersichtlich ist

Zustand (*state*)

Der Zustand eines Objekts umfasst die Attribute bzw. deren aktuellen Werte

Verhalten (*behavior*)

Das Verhalten eines Objekt wird durch eine Menge von Operationen beschrieben.

Objektidentität (*object identity*)

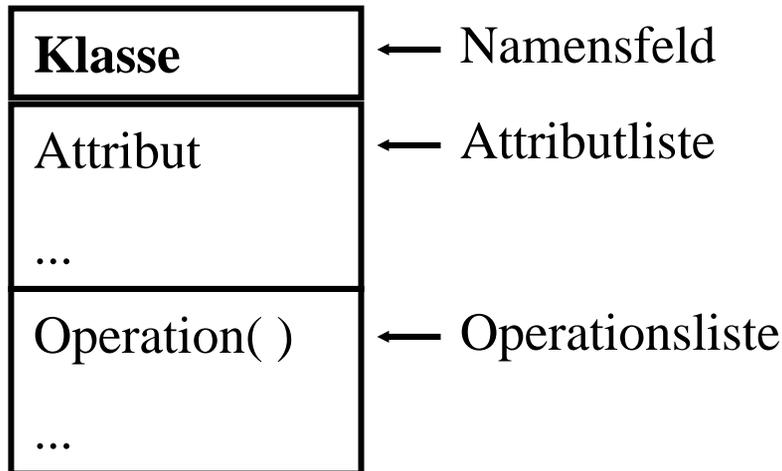
- Ist die Eigenschaft, die ein Objekt von allen anderen Objekten unterscheidet.
- Die Identität eines Objekts kann sich nicht ändern.
- Keine zwei Objekt können dieselbe Identität besitzen.

Objektname

Identifiziert ein Objekt im Objektdiagramm.

Klasse

- Definiert für eine Kollektion von Objekten deren Struktur (Attribute), Verhalten (Operationen) und Beziehungen (Assoziationen und Vererbungsstrukturen).
- Sie besitzen einen Mechanismus, um neue Objekte zu erzeugen (*object factory*).
- Das Verhalten (*behavior*) wird durch Botschaften (Nachrichten) beschrieben.



Klassenname

- Ist stets ein Singular
- Kann durch ein Adjektiv ergänzt werden
- Der Klassennamen muss eindeutig sein

Stereotyp (*stereotype*)

Klassifiziert Elemente des Modells

Merkmal (*property*)

- Beschreibt Eigenschaften des Modells
- Mehrere Merkmale können zusammengefasst werden

Generische Klasse (*parameterized class, template*)

Ist eine Beschreibung einer Klasse mit einem oder mehreren formalen Parametern.

Schnittstelle (*interface*)

Besteht nur aus den Signaturen von Operationen

Attribut

- Beschreibt die Daten, die von den Objekten einer Klasse angenommen werden können
- Alle Objekte einer Klasse besitzen dieselben Attribute, jedoch unterschiedliche Attributwerte
- Attribute werden durch Angabe von Typ, Anfangswert und Merkmalen **spezifiziert**



Attribut: Typ = Anfangswert

{Merkmal1, Merkmal2, ...}

Attributname

Muss im Kontext der Klasse eindeutig sein.

Klassenattribut (*class scope attribute*)

Ein Klassenattribut liegt vor, wenn nur ein Attributwert für alle Objekte einer Klasse existiert. Klassenattribute existieren auch dann, wenn es zu einer Klasse – noch – keine Objekte gibt.

Sichtbarkeit (*visibility*)

class
+ publicAttribute
protectedAttribute
- privateAttribute
~ packageAttribute

Für jedes Attribut wird im Entwurf die Sichtbarkeit angegeben. Die UML unterscheidet folgende Arten:

- Public: sichtbar für alle Klassen,
- Protected: sichtbar für alle Unterklassen und innerhalb der Klassen,
- Private: sichtbar nur innerhalb der Klasse,
- Package: sichtbar innerhalb des Paktes.

Operation

Klasse
Operation ()
<u>Klassenoperation ()</u>
Abstrakte Operation ()

Operation () {Merkmal1, Merkmal2, ...}

Eine Operation ist eine ausführbare Tätigkeit. Alle Objekte einer Klasse verwenden dieselben Operationen. Jede Operation kann auf alle Attribut eines Objekts dieser Klasse direkt zugreifen.

Abstrakte Operation

Eine abstrakte Operation besteht nur aus der Signatur.
In der UML werden abstrakte Operationen kursiv eingetragen oder mittels {abstract} gekennzeichnet.

Signatur (*signature*)

Die Signatur einer Operation besteht aus dem Namen der Operation, den Namen und Typen aller Parameter und dem Ergebnistyp.

Operation (Paramertliste) : Ergebnistyp

Für jeden Parameter der Parameterliste gilt:

[in | out | inout] Name: Typ = Anfangswert

Kardinalität

Die Kardinalität einer Assoziation definiert den zulässigen Wertebereich für die Anzahl der Objekte, die an der Assoziation teilnehmen (können).

Notation:

- Nummer
- Nummer..Nummer
- Nummer..
- k
- *

:=

Beispiel:

1,2, 3,4
0..1 1..2 0..1,4
* 1..* 2..*
*
*

Äquivalent zu 0..*

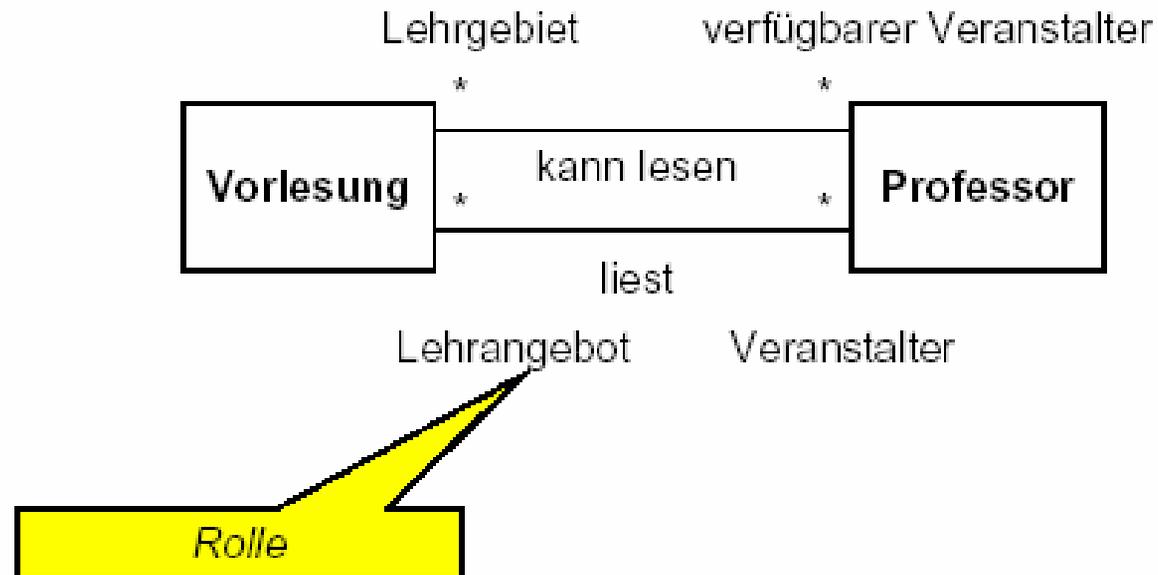
Rolle

Eine **Rolle** in einer Assoziation ist ein Name, der die Beteiligung der

Klasse an der Assoziation näher besch

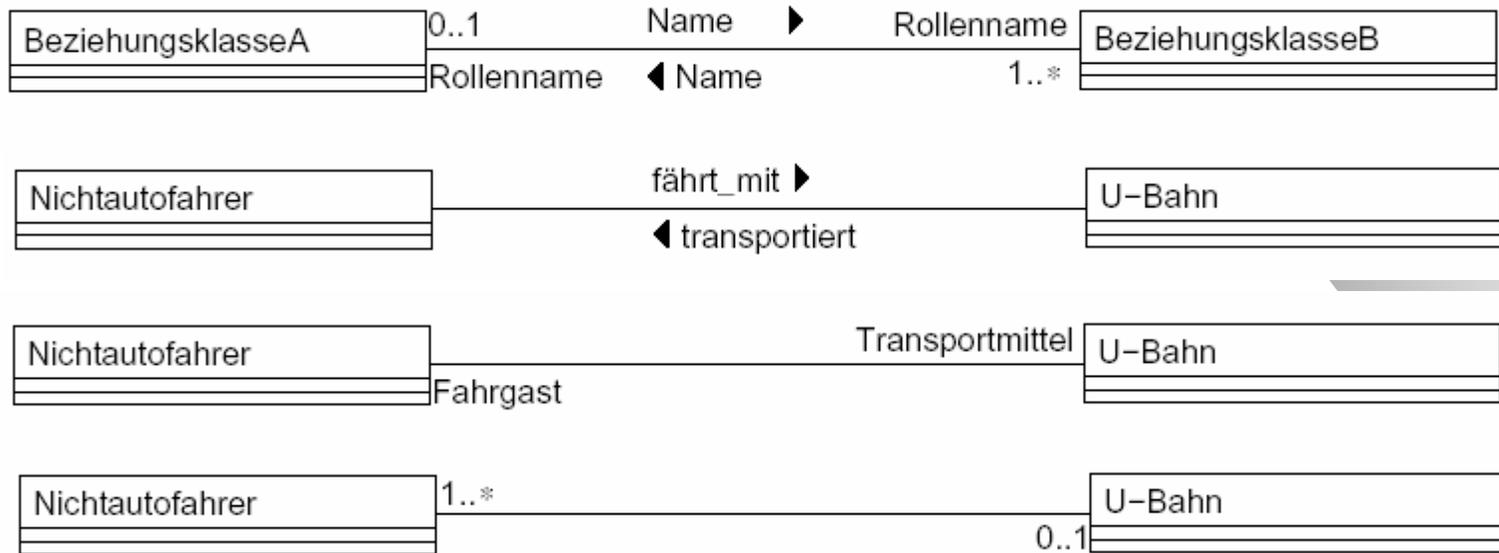
Englischer Begriff: *role*

Notation:



Assoziationen:

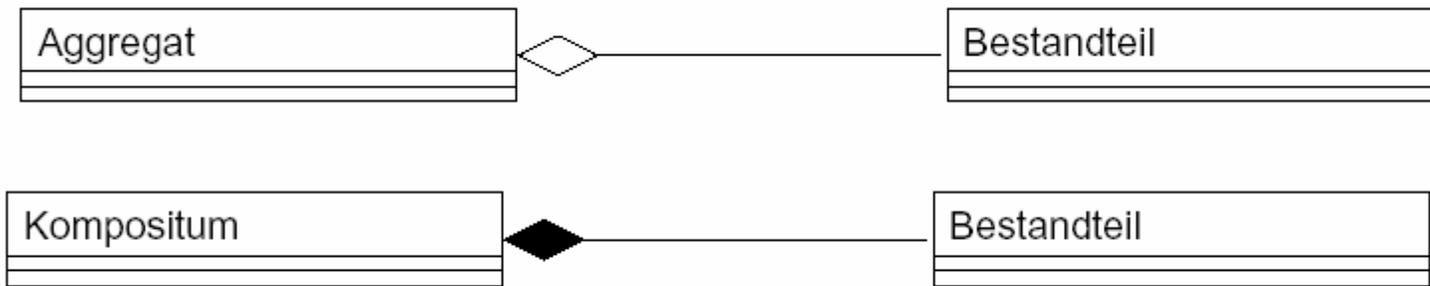
Beziehungen werden durch Assoziationen modelliert:



Die Beziehung besteht zwischen Objekten!

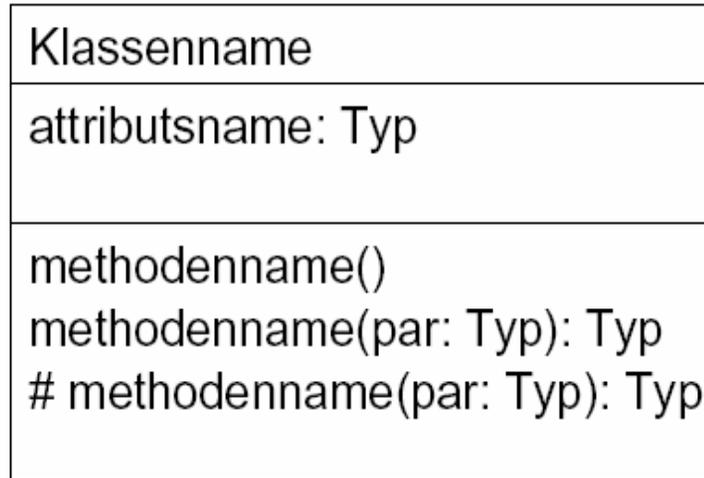
Aggregation und Komposition:

beide Assoziationen drücken die "ist-Bestandteil-von"
bzw. "enthält"-Beziehung aus.



Der Unterschied besteht in der Stärke der Bindung . . .

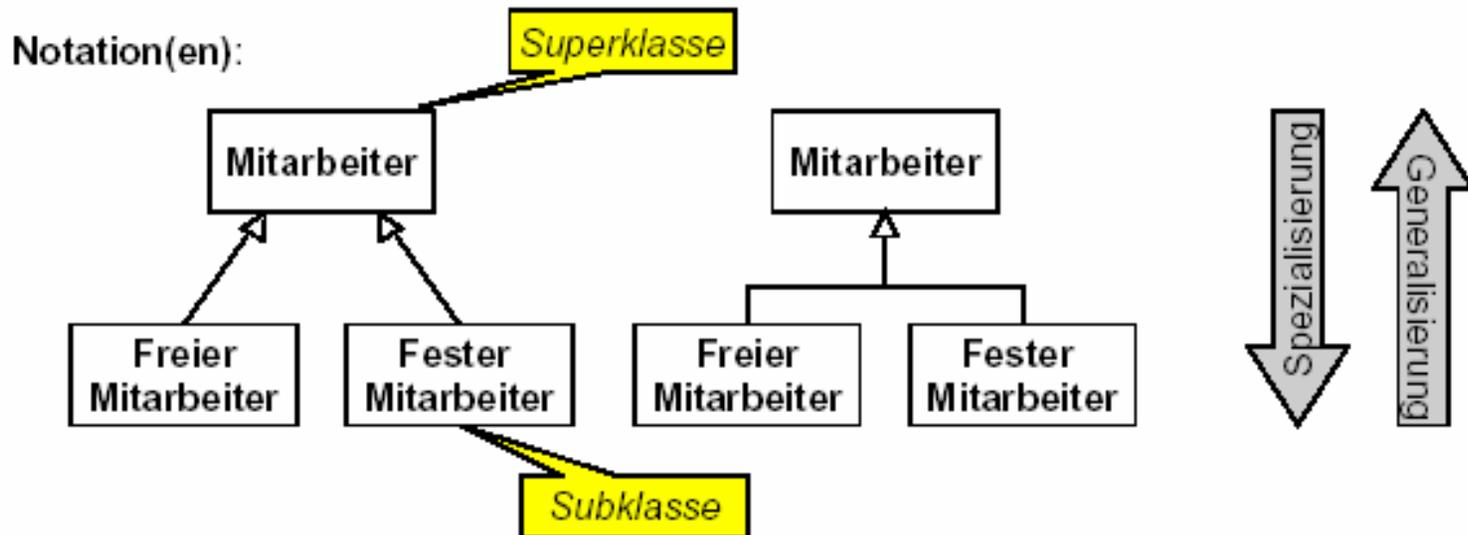
Klassendiagramme in UML: Klassen, Attribute, Methoden



Je nach Modell–Ebene unterschiedlich detaillierte Angaben

Generalisierung

Die **Generalisierung** ist eine semantische Beziehung zwischen einem allgemeineren Konzept A (**Superklasse**) und einem spezielleren Konzept B (**Subklasse**). „A generalisiert B“, „jedes B ist ein A“

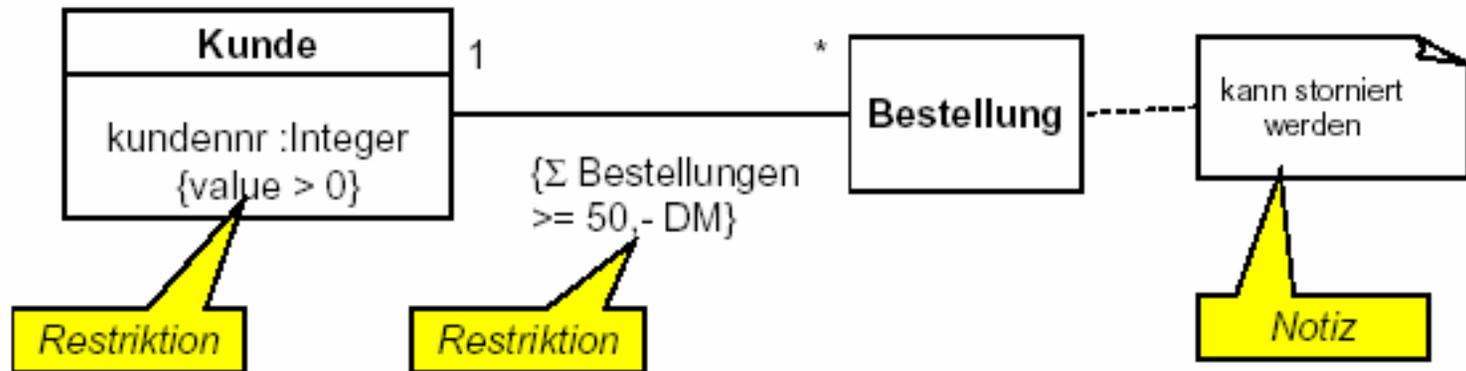


Restriktion und Notiz

Restriktionen und **Notizen** können u.a. an Assoziationen, an Attributen und an Klassen annotiert werden. Restriktionen sind semantische Einschränkungen, die als Ausdruck

Englischer Begriff: *constraint*

Englischer Begriff: *note*

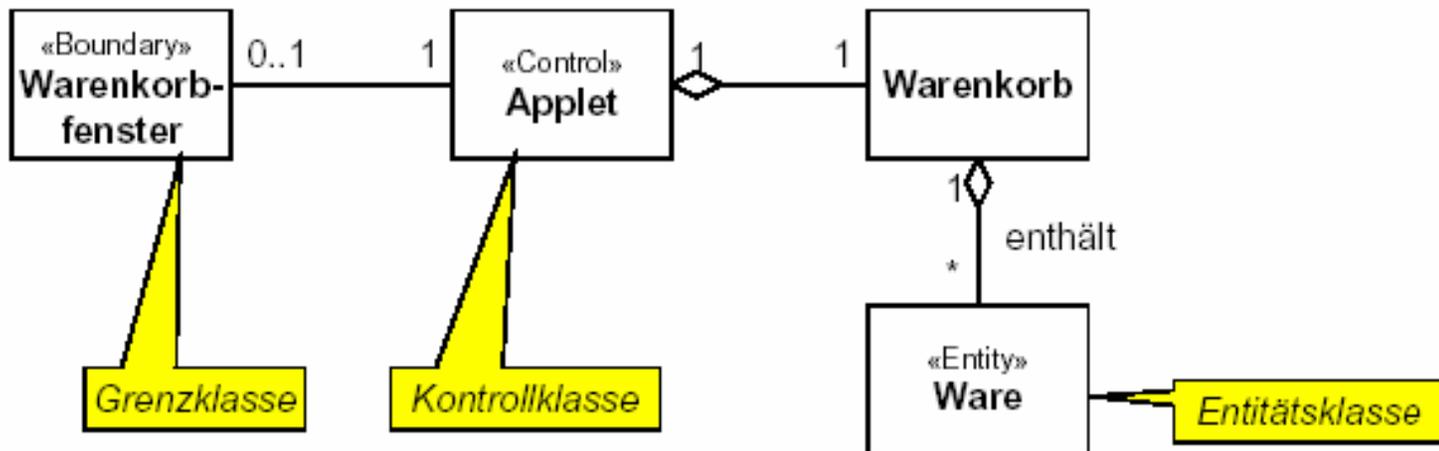


Auch Notizen können Restriktionen enthalten.

Entitäts-, Grenz- und Kontrollklasse

Unterscheidung von verschiedenen Arten von Klassen:

- **Entitätsklasse**: modelliert langlebige, passive Objekte und Daten und deren Verhalten
- **Grenzklasse**: modelliert Schnittstellen zu Aktoren und deren Verhalten
- **Kontrollklasse**: modelliert einen (anwendungsfallbezogenen) Manager für die Interaktion zwischen Grenzklassen und dem System



Objektdiagramm

Ein **Objektdiagramm** zeigt Objekte (Instanzen von Klassen), evtl. Ihren Zustand und ihre Beziehungen zu eine bestimmten Zeitpunkt.

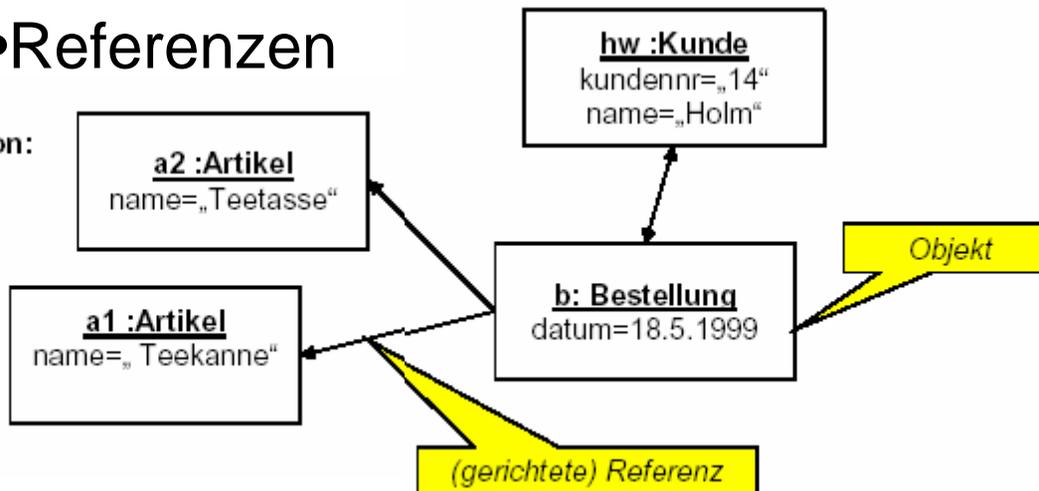
Bestandteile

:

- Objekte

- Referenzen

Notation:



Methodisches Vorgehen

Methode Erstelle konzeptuelles Klassendiagramm

Eingabe: Pflichtenheft.Anwendungsfall-Modell,
Pflichtenheft.Glossar

Ausgabe: Pflichtenheft.KonzeptuellesKlassendiagramm

Änderung: Pflichtenheft

Pflichtenheft.KonzeptuellesKlassendiagramm:= ∅

repeat

Finde neue **Klassen** und Attribute

Finde neue **Assoziationen**

Dokumentiere Klassen und Beziehungen

Bespreche das Ergebnis mit dem Auftraggeber

until Auftraggeber und Auftragnehmer zufrieden

Methodisches Vorgehen: Finde Klassen

1. Identifiziere Klassen über die Substantivmethode

- Relevante Dokumente: Lastenheft, Anwendungsfalldokumentation, Glossar, Formulare, technische Systembeschreibungen, Gesprächsnotizen, ...
- identifiziere **Substantive** (Konzepte), *die für das Problemverständnis relevant sind*
 - Technik: Unterstreichen im Text, lieber zunächst zu viele Substantive als zu wenige
- Identifiziere **Synonyme**, **Akronyme** und Fachbegriffe.
 - Zwei verschiedene Namen sind Synonyme, falls sie das gleiche Konzept bezeichnen.
 - Ein Akronym (Initialwort) ist ein Name, der aus den Anfangsbuchstaben eines Namens gebildet wird.
- Identifiziere **Homonyme**.
 - Ein Homonym ist ein *Name*, der zwei (verschiedene) Konzepte benennt.

Methodisches Vorgehen: Finde Klassen

2. Kategorisiere die Substantive

- Konkrete Objekte bzw. Dinge, z.B. Pkw
- Personen und deren Rollen, z.B. Kunde, Mitarbeiter, Dozent
- Informationen über Aktionen, z.B. Banküberweisung
- Orte, z.B. Wartezimmer
- Organisationen, z.B. Filiale
- Schnittstellen des Systems, z.B. Liste, Auswahlliste
- Beziehungen zwischen Klassen, z.B. Mietvertrag zwischen Kunde und Mietobjekt
- Allgemeine und fachbezogene Informationen, z.B. Artikel, Seminartyp

Methodisches Vorgehen: Finde Klassen

3. Streiche Substantive und Konzepte, die keine eigenständigen konzeptuellen Klassen bezeichnen

- Das Ziel ist **nicht**, möglichst viele Klassen oder Klassen möglichst geringerer Komplexität zu identifizieren.
- Gestrichene Substantive können evtl. in einer **To-Do-Liste** gesammelt werden, da sie später bei der Identifikation relevanter Attribute, Operationen und Beziehungen hilfreich sind.
- Streiche Namen für **Rollen**, die eine Klasse in einer Beziehung zu einer anderen Klasse spielt.
- **Indizien** für zu streichende Substantive:
 - Es lassen sich weder Attribute noch Operationen identifizieren.
 - Das Substantiv modelliert Implementierungsdetails.
 - Die Klasse enthält nur Operationen, die sich anderen Klassen zuordnen lassen.

Methodisches Vorgehen: Finde Klassen

4. Wähle aussagefähige und verbindliche Klassennamen

Jeder Klassenname soll

- ein Substantiv im Singular sein,
- so konkret wie möglich gewählt werden,
- kein Homonym sein,
- die Gesamtheit der Attribute und Operationen darstellen.

5. Dokumentiere jede Klasse knapp

- Ein definierender Satz, z.B. Eine *Seminarbuchung* ist ein Dokument, das die Anmeldung eines Kunden zu einer Seminarveranstaltung dokumentiert.
- Liste der Synonyme, eventuell Abgrenzung zu anderen Klassen

Synonyme: Anmeldung, Reservierung

Methodisches Vorgehen: Finde Attribute

1. Identifiziere Attribute im Anschluß an die Substantivmethode

- Identifiziere für das Pflichtenheft relevante Attribute jeder Klasse.
- Erfasse Schlüsselattribute nur dann, wenn sie **fachlich** notwendig sind.

2. Überprüfe den Attributnamen

- Jeder Attributname soll
 - ein Substantiv im Singular oder Plural sein,
 - so konkret wie möglich gewählt werden,
 - kein Homonym sein.

3. Definiere den Typ der Attribute

- Im konzeptuellen Klassendiagramm kann der Typ unspezifiziert oder nur vage
- spezifiziert bleiben. Komplex strukturierte Typen sollten durch eigenständige Klassen ersetzt werden.

Methodisches Vorgehen: Finde Assoziationen

1. Identifiziere mögliche Assoziationen zwischen Objekten

- Suche in den relevanten Dokumenten nach **Verben** und nach **Substantiven**, die Aktionen oder Prozesse in der Problembeschreibung identifizieren.
- Identifiziere für jede Assoziation die beteiligten Klassen.
- Bevorzuge binäre Assoziationen (d.h. mit zwei beteiligten Klassen).

2. Kategorisiere diese Assoziationen

- räumliche Nähe (in der Nähe von)
- Aktionen (fährt, bucht)
- Kommunikation (redet mit)
- Besitz (hat)
- allgemeine Beziehungen (ist abhängig von, ist verheiratet mit)

Methodisches Vorgehen: Finde Assoziationen

3. Streiche Assoziationen, die keine Assoziationen im konzeptuellen Klassendiagramm sind

- Streiche nicht-permanente Beziehungen
- Streiche für das Pflichtenheft irrelevante Assoziationen
- Streiche implementierungsbezogene Assoziationen

4. Falls erforderlich, definiere Assoziations- und Rollennamen

- Assoziationen sind in der Mehrzahl der Fälle durch die partizipierenden Klassen eindeutig bestimmt. In diesem Fall sind Assoziations- und Rollennamen nicht erforderlich.
- Ansonsten (z.B. bei rekursiven Assoziationen) definiere
 - einen **Namen** (ein Verb oder eine Verbalphrase) für die Assoziation
 - einen **Rollennamen** für jede an der Assoziation beteiligte Klasse
 - einen **Satz**, der die Semantik der Assoziation beschreibt.

Methodisches Vorgehen: Finde Assoziationen

5. Bestimme die Kardinalität jeder Rolle jeder Assoziation

- Liegt eine Muß-Beziehung vor ? ⇒ Kardinalität 1..
 - Sobald das Objekt erzeugt ist, muß auch die Beziehung zu dem anderen Objekt aufgebaut werden.
- Liegt eine Kann-Beziehung vor ? ⇒ Kardinalität 0..
 - Die Beziehung kann zu einem beliebigen Zeitpunkt nach dem Erzeugen des Objekts erzeugt werden.
- Ist die Obergrenze fest oder variabel ? ⇒ Kardinalität ..k
 - Ist eine Obergrenze vom Problem her **zwingend** vorgegeben?
 - Im Zweifelsfall immer mit **variablen** Obergrenzen arbeiten!
- Gelten besondere Bedingungen?
 - Gerade Anzahl, Untergrenze, ...

Methodisches Vorgehen: Finde Assoziationen

5. Unterscheide Assoziation und Aggregation anhand der folgenden Kann-Kriterien:

- Läßt sich die Beziehung durch „besteht aus“ oder „ist Teil von“ beschreiben? (Kollektion, Behälter, Ganzes & Teile, Gruppe & Mitglied)
- Ist die Kardinalität auf einer Seite der Assoziation 1 oder 0..1 ?
- Ist die Assoziation transitiv und asymmetrisch?
- Gehören die Komponenten in ein Subsystem?
- Erfolgt der Zugriff auf die Teilobjekte ausschließlich über das Aggregat-Objekt?
- Ist die Lebensdauer der Komponente durch die Lebensdauer des Aggregats beschränkt?

Bemerkung:

Attribute sind auch durch ihre Klasse aggregiert, sie haben aber keine eigenständige Existenz und können daher nicht unabhängig von ihrer Klasse an Beziehungen teilnehmen.