

## 6.3 Entity-Relationship-Modell

Entity-Relationship-Modell, **ER-Modell** (P. Chen 1976): Kalkül zur Modellierung von **Aufgabenbereichen mit ihren Objekten, Eigenschaften und Beziehungen.**

**Weitergehende Zwecke:**

- **Entwurf von Datenbanken;**  
Beschreibung der Daten, die die DB enthalten soll, „konzeptionelles Schema“
- **Entwurf von Software-Strukturen**  
Entwurfssprache UML basiert auf ER

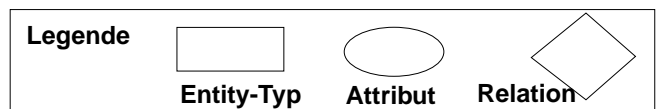
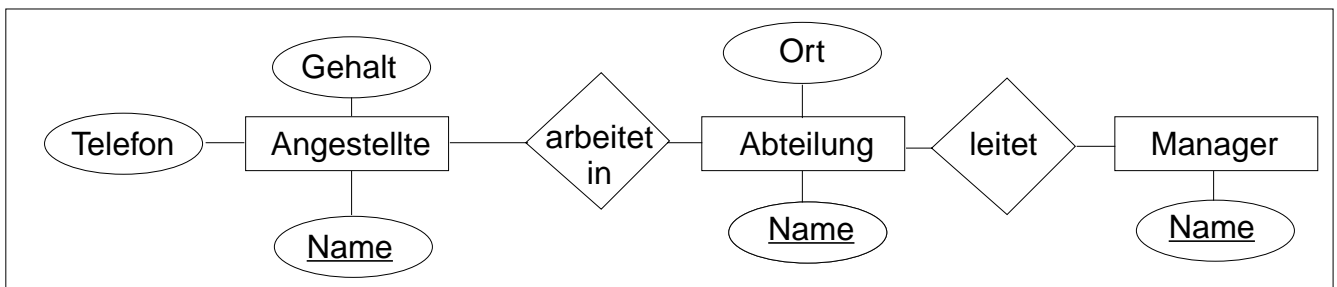
Grundbegriffe

- **Entity** **Objekt** des Aufgabenbereiches
- **Relation** **Beziehung** zwischen Objekten
- **Attribut** Beschreibt ein **Eigenschaft** eines Objektes durch einen **Wert**

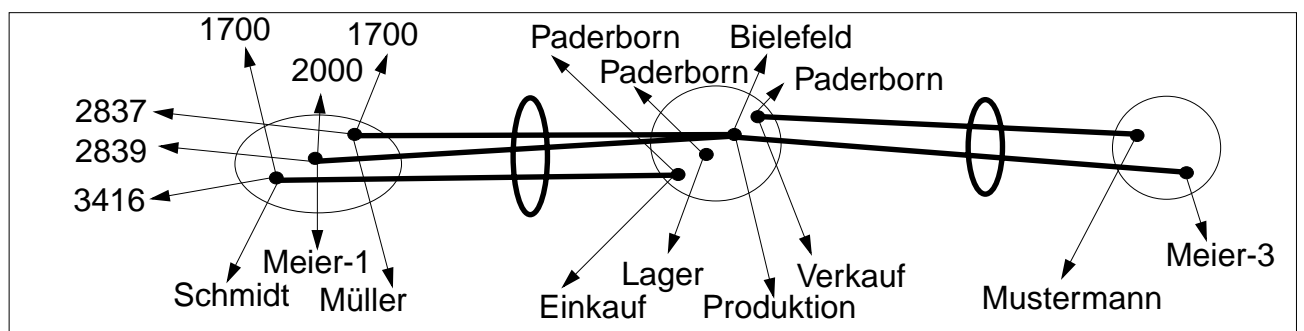
**Graphische** und textuelle **Notationen** für ER-Modellierungen; hier graphische

## Einführendes Beispiel

Ausschnitt aus der **Modellierung** einer Firmenorganisation: [Beispiel nach J. D. Ullman: Principles ...]



Eine **konkrete Ausprägung** zu dem Modell:



## Entities

### Entity:

**Objekt**, Gegenstand aus dem zu modellierenden **Aufgabenbereich**

Jede Entity hat eine **eindeutige Identität**, verschieden von allen anderen

### Entity-Menge (auch Entity-Typ):

**Zusammenfassung von Objekten**, die im Modell als **gleichartig** angesehen werden,

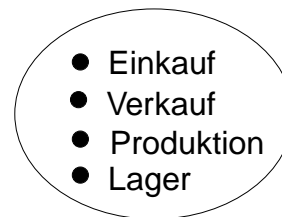
z. B. Angestellte, Abteilung, Manager

Im **Modell steht eine Entity-Menge** für die ggf. nicht-endliche Menge aller infrage kommenden Objekte dieser Art.

Eine **konkrete Ausprägung zu der Entity-Menge** ist eine endliche Teilmenge davon.

Abteilung

steht im Modell für die  
**Menge aller** in  
Unternehmen **möglichen**  
**Abteilungen**



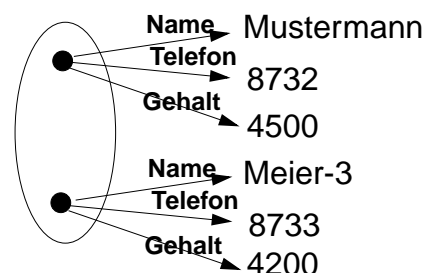
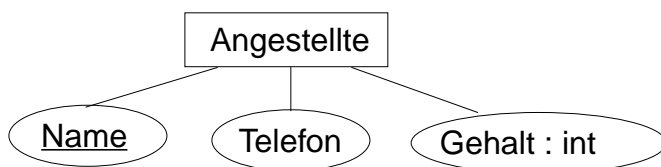
konkrete Ausprägung dazu:  
die **Menge der Abteilungen** eines  
konkreten Unternehmens

## Attribute

**Attribute beschreiben Eigenschaften von Entities.**

Einer Entity-Menge im Modell können Attribute zugeordnet werden, z. B.

eine konkrete Ausprägung:



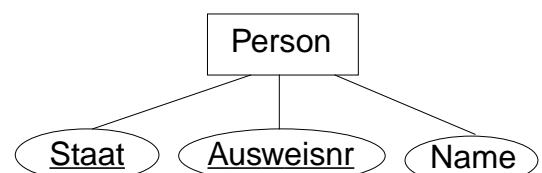
Ein Attribut ordnet jeder Entity aus der konkreten Entity-Menge einen Wert zu.

Der **Wertebereich eines Attributes** kann explizit angegeben sein, z. B. int für Gehalt, oder er wird passend angenommen.

Ein Attribut, dessen **Wert jede Entity eindeutig identifiziert**, heißt **Schlüsselattribut**.

Es wird im Modell unterstrichen.

Auch **mehrere Attribute zusammen** können den Schlüssel bilden:



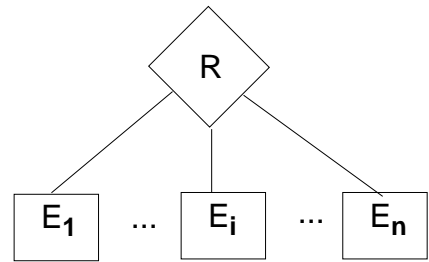
# Relationen

Relationen modellieren Beziehungen zwischen den Entities der Entity-Mengen.

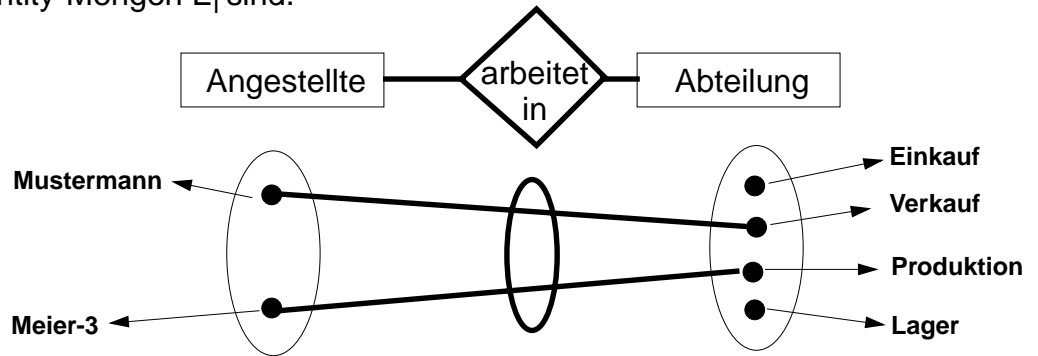
## n-stellige Relation R

über n Entity-Mengen  $E_1, \dots, E_n$ , mit  $n \geq 2$ :

Im Modell wird dadurch der **Typ der Relation** angegeben.

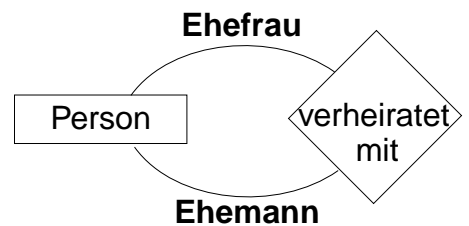


Eine **konkrete Ausprägung von R** ist eine Menge von n-Tupeln  $(e_1, \dots, e_n)$ , wobei die  $e_i$  Entities aus den konkreten Ausprägungen der Entity-Mengen  $E_i$  sind.



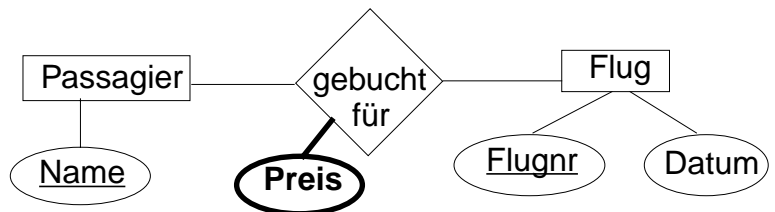
# Rollen und Attribute in Relationen

Für manche Relationen wird aus ihrem Namen und der Graphik nicht klar, welche Bedeutung die Entity-Mengen in der Relation haben. Man kann das durch **Rollenamen an den Kanten** verdeutlichen.

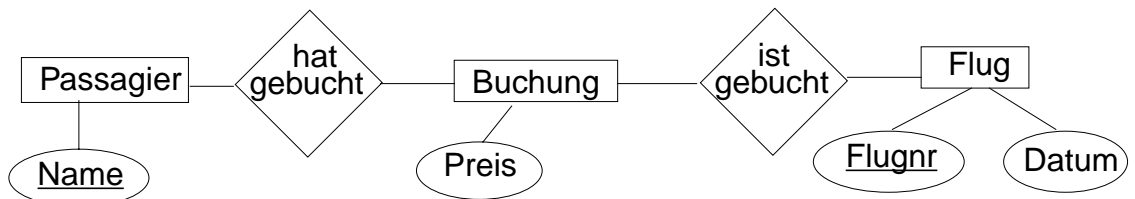


Auch **Relationen können Attribute haben**. Sie beschreiben **Eigenschaften zu jedem Tupel der Relation**.

Der Preis ist eine **Eigenschaft der Buchung** - nicht des Passagieres oder des Fluges.



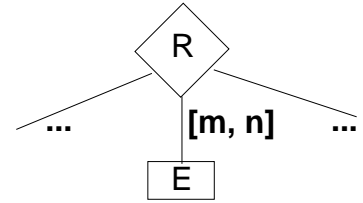
Man könnte natürlich auch **Buchungen als Entities** modellieren:



# Kardinalität von Relationen

In Relationen wird durch Angaben zur **Kardinalität** bestimmt, wie oft eine Entity in den Tupeln der Relation vorkommen kann bzw. vorkommen muss:

**Für jede konkrete Ausprägung der Relation R** muss gelten: **Jede Entity e** aus der konkreten Entity-Menge zu E kommt **in mindestens m und höchstens n Tupeln** vor.



## Spezielle Kardinalitäten:

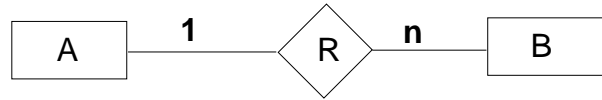
[1, 1] in **genau einem** Tupel: totale Funktion von E auf die übrigen Rollen der Relation

[0, 1] in **höchstens einem** Tupel: partielle Funktion von E auf die übrigen Rollen

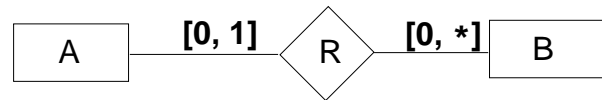
[0, \*] in **beliebig vielen** Tupeln

Ohne Angabe wird [0, \*] angenommen.

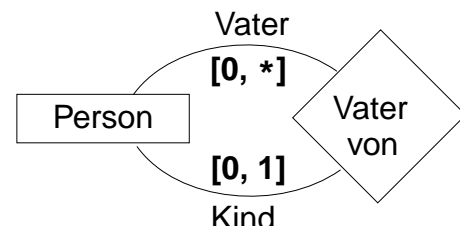
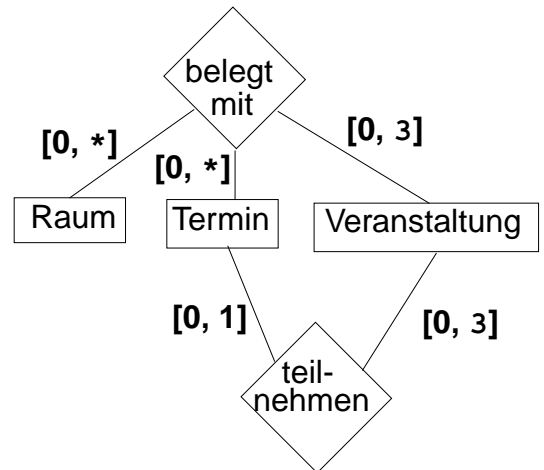
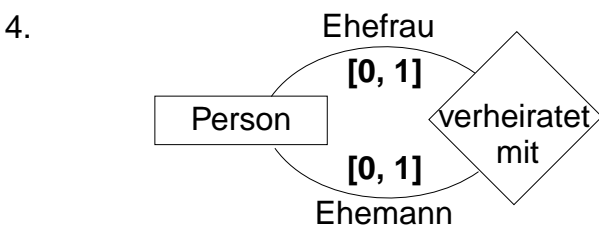
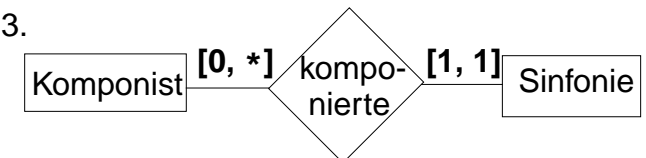
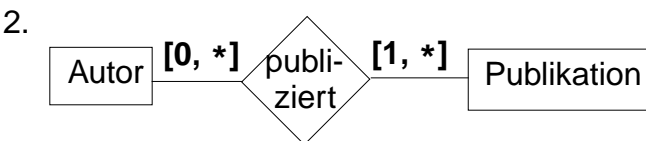
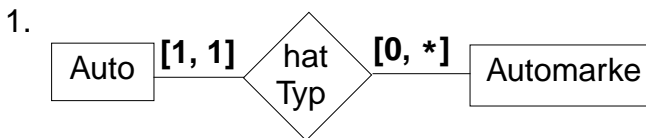
Kurznotation für 2-stellige Relationen:



bedeutet:



# Beispiele zu Kardinalitäten in Relationen

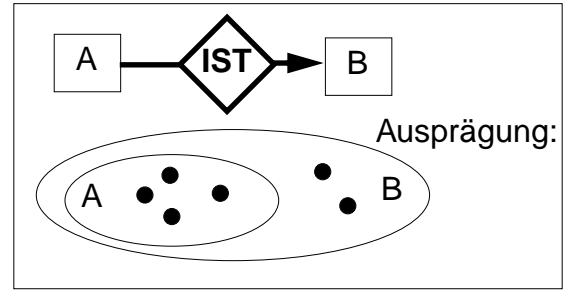


# IST-Hierarchie

Die spezielle **Relation IST** (engl. is-a) definiert eine **Spezialisierungs-Hierarchie** für Entity-Mengen:

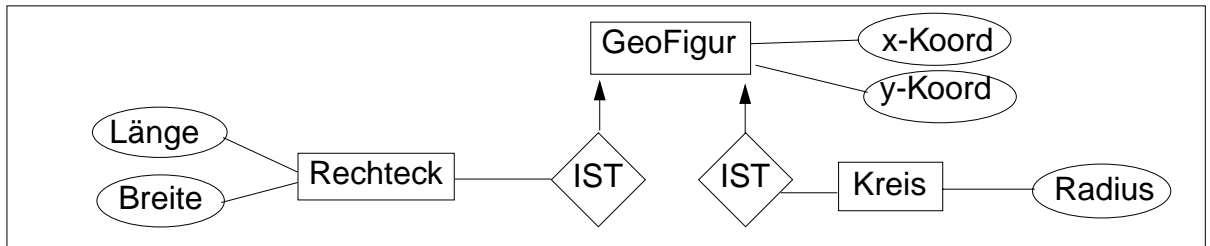
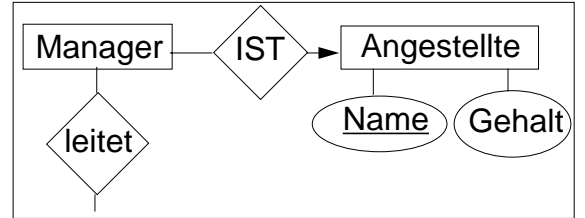
**A IST B:** Einige Entities der **allgemeineren Menge B** gehören auch der **spezielleren Menge A** an.

Jede konkrete Ausprägung zu A ist **Teilmenge** der konkreten Ausprägung zu B.  
Es kann Entities in B geben, die nicht in A sind.

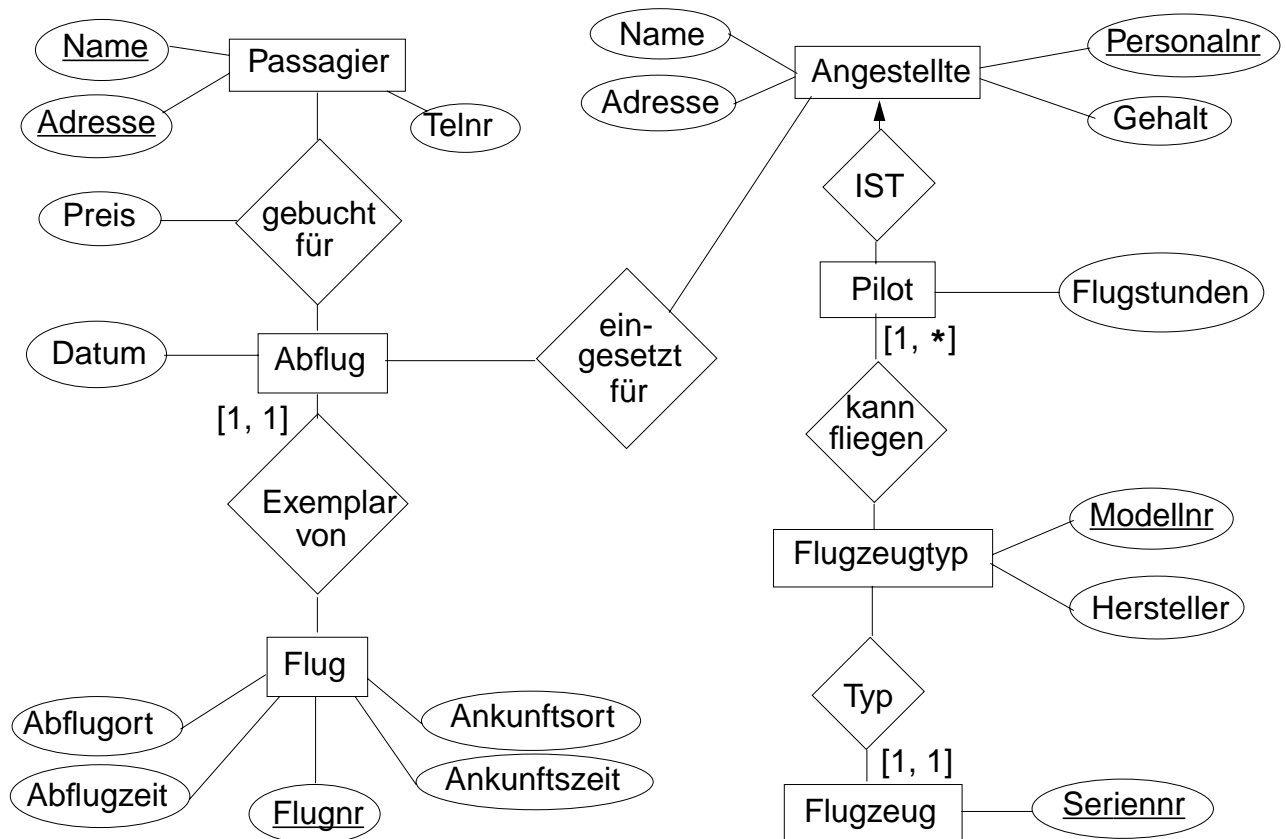


Die **Entities in A** „erben“ **alle Attribute von B** und können noch weitere Attribute haben, die **spezielle A-Eigenschaften** beschreiben.

Auch **Schlüsselattribute** werden als solche **geerbt**.

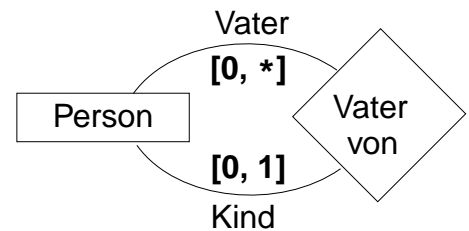


# Beispiel: Fluggesellschaft

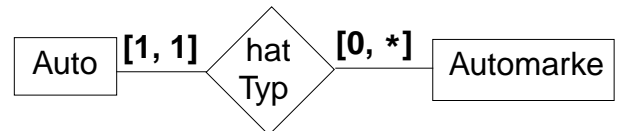


## Hinweise zur Modellierung mit ER

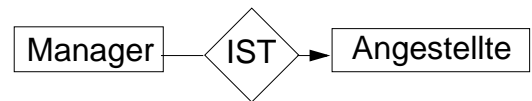
- In einem ER-Modell kommt **jede Entity-Menge nur einmal** vor.
- **Rollen** zu Relationen **angeben**, wo es nötig ist.
- Bedeutung der Kardinalitäten klarstellen.



- **Typ - Exemplar - Relationen** bewusst einsetzen.



- **Spezialisierung** sinnvoll einsetzen.



- Typ - Exemplar - Relation **nicht** mit Spezialisierung **verwechseln**

## 6.4 Klassendiagramme in UML Übersicht

1. **UML (Unified Modelling Language):**  
die derzeit wichtigste Sprache zur **Modellierung von Systemen**
2. Als **Zusammenfassung mehrerer Modellierungssprachen**  
**1997** in der Version 1.1 definiert;  
Version 2.0 von 2005 ist Grundlage aktueller UML-Versionen.
3. **Object Management Group** macht aktuelle Dokumente zu UML verfügbar:  
Object Management Group: UML Resource Page. [www.uml.org](http://www.uml.org) (2010)
4. UML umfasst **13 Teilsprachen (Diagrammtypen)**, um unterschiedliche Aspekte von Systemen zu beschreiben, z. B.  
**Klassendiagramme** für Systemstruktur, statische Eigenschaften und Beziehungen,  
**Statecharts** für Abläufe von Operationen.
5. Für den Gebrauch durch Menschen hat UML graphische Notationen (visuelle Sprachen);  
Software-Werkzeuge verwendendie XML Sprache **XMI (XML Metadata Interchange)**
6. **Einführendes Buch:**  
Chris Rupp, Stefan Queins, Barbara Zengler:  
UML 2 glasklar. 3. Auflage; Carl Hanser Verlag (2007)

## Bezug zum ER-Modell

**Klassendiagramme** dienen zur Modellierung von **Systemstruktur, statischen Eigenschaften** und **Beziehungen**.

Sie basieren auf den gleichen Grundkonzepten wie das Entity-Relationship-Modell:

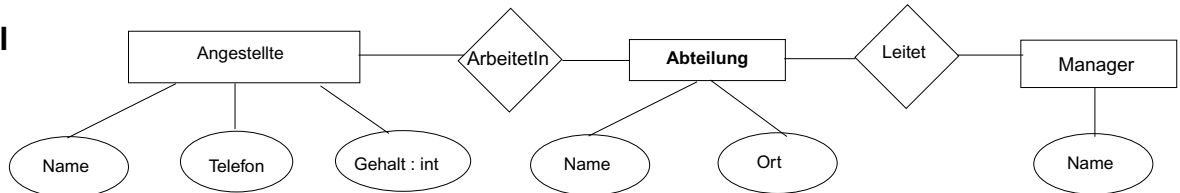
### ER-Modell

Entity-Menge  
Attribut  
Relation

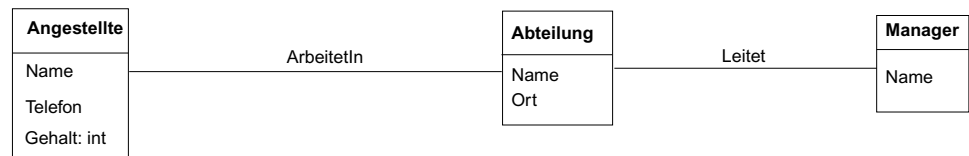
### UML Klassendiagramm

Klasse  
Attribut  
Assoziation

#### ER-Modell



#### UML Klassendiagramm

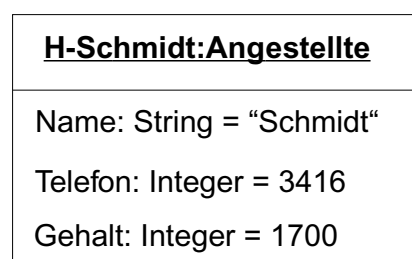


## Klasse mit Attributen

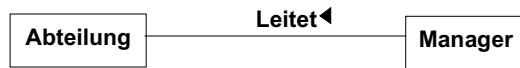
**Klasse:** repräsentiert eine Menge gleichartiger Objekte (wie im ER-Modell); Attribute (und ggf. Operationen) werden im Rechteck der Klasse angegeben.



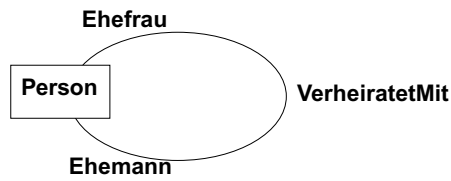
Objekte einer Klasse werden so dargestellt:



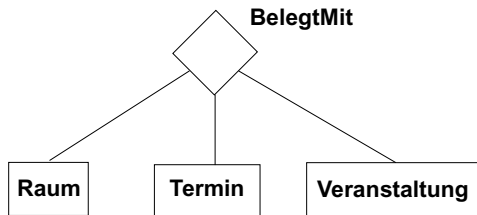
# Assoziationen



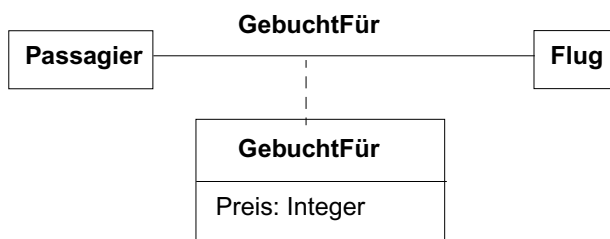
zweistellig  
 ◀ gibt die Leserichtung an



zweistellig  
 mit Angabe der Rollen



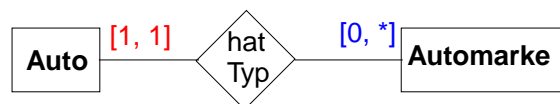
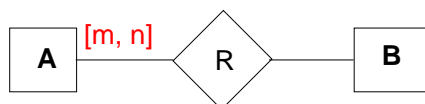
mehrstellig



Assoziation mit Attributen

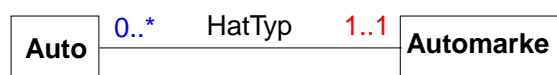
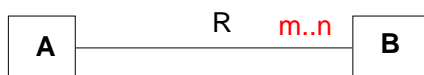
# Kardinalität von 2-stelligen Assoziationen

## ER:



Jedes Objekt aus A kommt in den Tupeln der Relation R **mindestens m und höchstens n** mal vor.

## UML:



Jedem Objekt aus A ordnet die Relation R **mindestens m und höchstens n** verschiedene Objekte aus B zu.



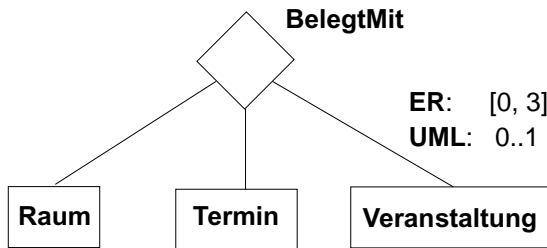
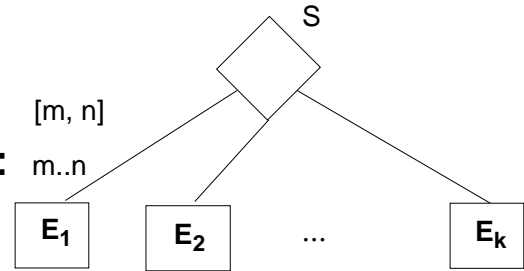
## Kardinalität von k-stelligen Assoziationen

Jedes Objekt aus E1 kommt in den Tupeln der Relation S mindestens m und höchstens n mal vor.

Jeder Kombination von Objekten aus E2, ..., En ordnet die Relation S mindestens m und höchstens n Objekte aus E1 zu.

ER: [m, n]

UML: m..n



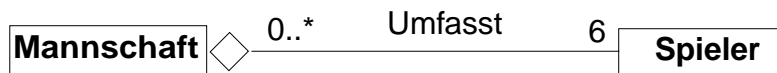
ER: [0, 3]  
UML: 0..1

Für jede Veranstaltung sind zwischen 0 und 3 Raum-Termin-Kombinationen vorgesehen. (nicht in UML formulierbar)

Für jede Raum-Termin-Kombination ist höchstens eine Veranstaltung vorgesehen. (nicht in ER formulierbar)

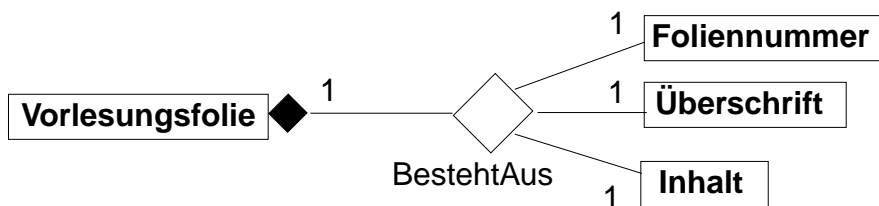
## Aggregation und Komposition

**Aggregation:** Objekte werden zu einem größeren Objekt zusammengefasst. sie können prinzipiell auch allein existieren.



- Eine Mannschaft umfasst immer 6 Spieler
- Ein Spieler kann einer, mehreren oder auch keiner Mannschaft angehören

**Komposition:** Jedes Teilobjekt gehört unverzichtbar zu genau einem ganzen Objekt.



Eine Vorlesungsfolie besteht immer aus einer Foliennummer, einer Überschrift und dem Folieninhalt.

# Generalisierung, Spezialisierung

Die Generalisierung (Spezialisierung) dient zur Modellierung von **Abstraktionshierarchien** (wie die **IST-Relation** in ER):

SK1 und SK2 sind **speziellere** Arten der **allgemeineren** GK.

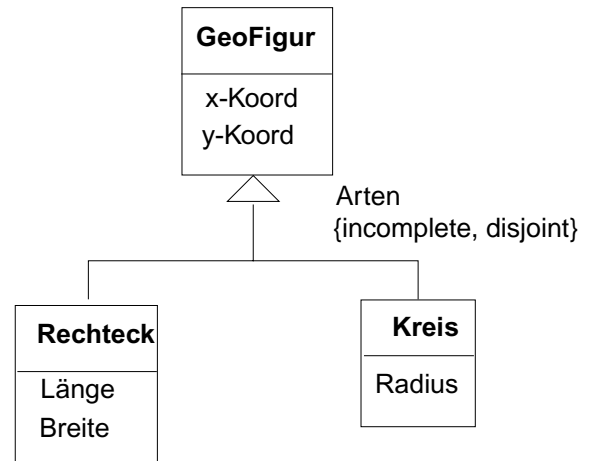
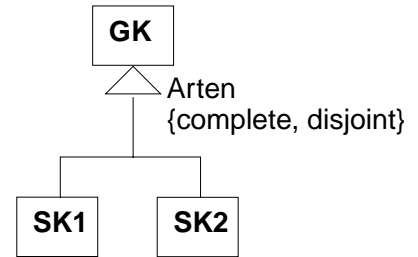
GK heißt auch **Oberklasse** der **Unterklassen** SK1 und SK2.

Die Assoziation kann **benannt** werden, hier *Arten*.

Hinsichtlich der Objekte gilt: SK1 und SK2 sind **Teilmengen** von GK.

Das Verhältnis der Unterklassen zueinander kann weiter charakterisiert werden:

- **disjoint**: Die Teilmengen sind paarweise disjunkt.
- **complete**: Es gibt in dem Modell **keine weiteren Unterklassen** von GK



# Modell einer Fluggesellschaft

vergl. Folie 6.17

