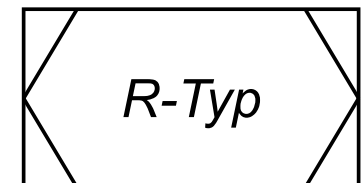
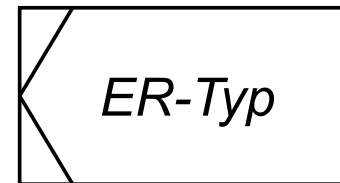
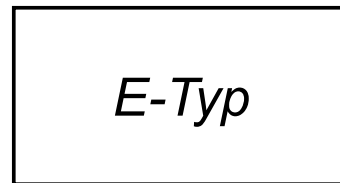


3 Lernziele

- Sie können
 - die Motivation zur Erweiterung des ERM durch das SERM wiedergeben.
 - den Nutzen des Strukturierten Entity Relationship Modells erläutern.
 - die Elemente des Strukturierten Entity Relationship Modells beschreiben.
 - bestehende SERMs analysieren und interpretieren.
 - eigene SERMs modellieren.

- Viele Objekttypen
- Unübersichtlich
- Unstrukturiert
- Abhängigkeiten kaum sichtbar

Objekttypen



Beziehungen

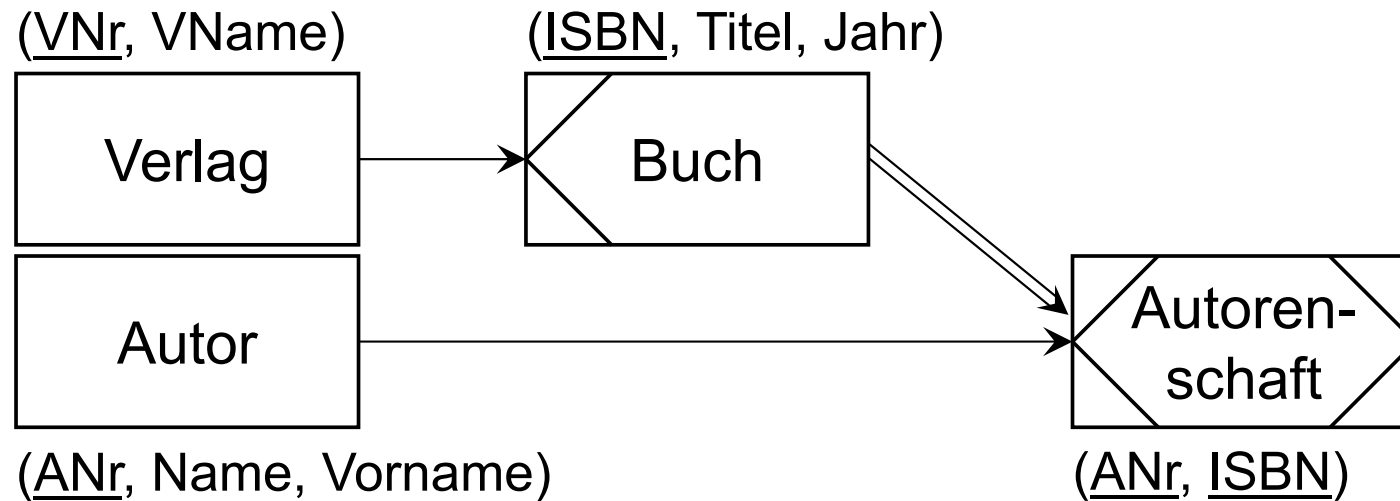
min-Eckwert (0) einfache Linie
(1) doppelte Linie
max-Eckwert (1) ohne Pfeilspitze
(*) mit Pfeilspitze

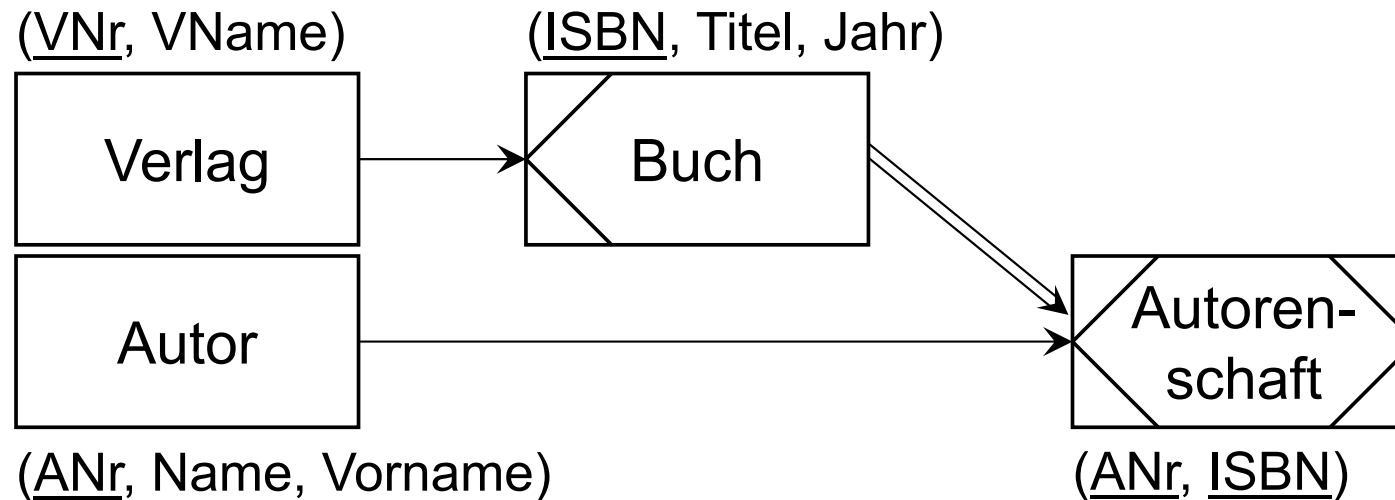
min \ max	0	1
1	<u>(0,1)</u>	<u><u>(1,1)</u></u>
*	<u>(0,*)</u> →	<u><u>(1,*)</u></u> →

Parallele Kanten zwischen zwei Objekttypen sind erlaubt, benötigen aber eine Beschriftung, einen sogenannten Rollennamen.

3 Eigenschaften SERM

- Übersichtlichkeit
 - Definition einer Anordnungsspezifikation im Sinne einer Links-Rechts-Anordnung gemäß Existenzabhängigkeiten
 - Quasi-hierarchischer Existenzgraph
- Transparenz semantischer Eigenschaften
 - ER-Typ als zusätzlicher Objekttyp
 - Transparenz semantischer Abhängigkeiten zwischen Objekttypen zur Konsistenzsicherung
 - Vererbung von Attributen „leichter“ durchzuführen



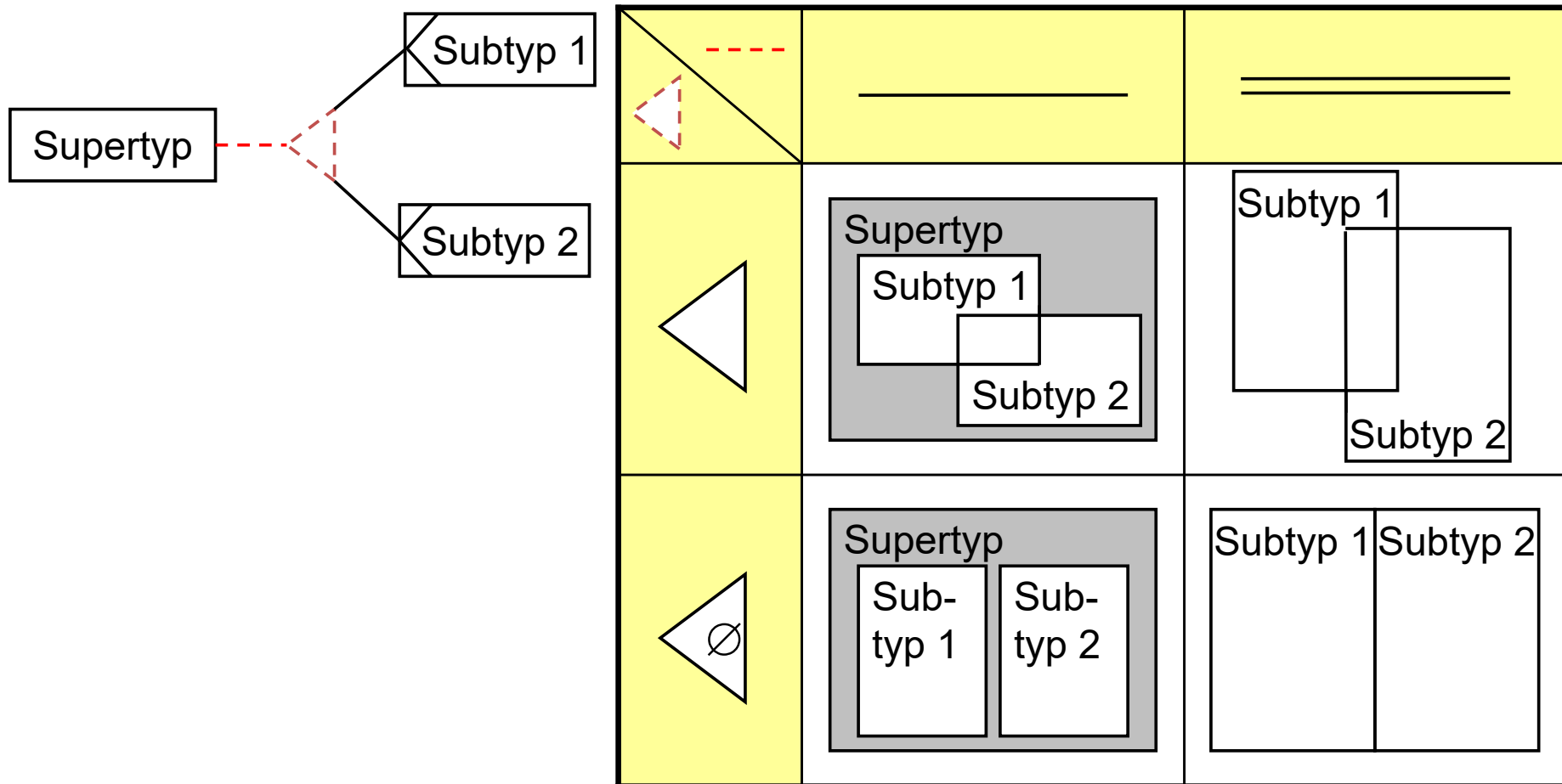


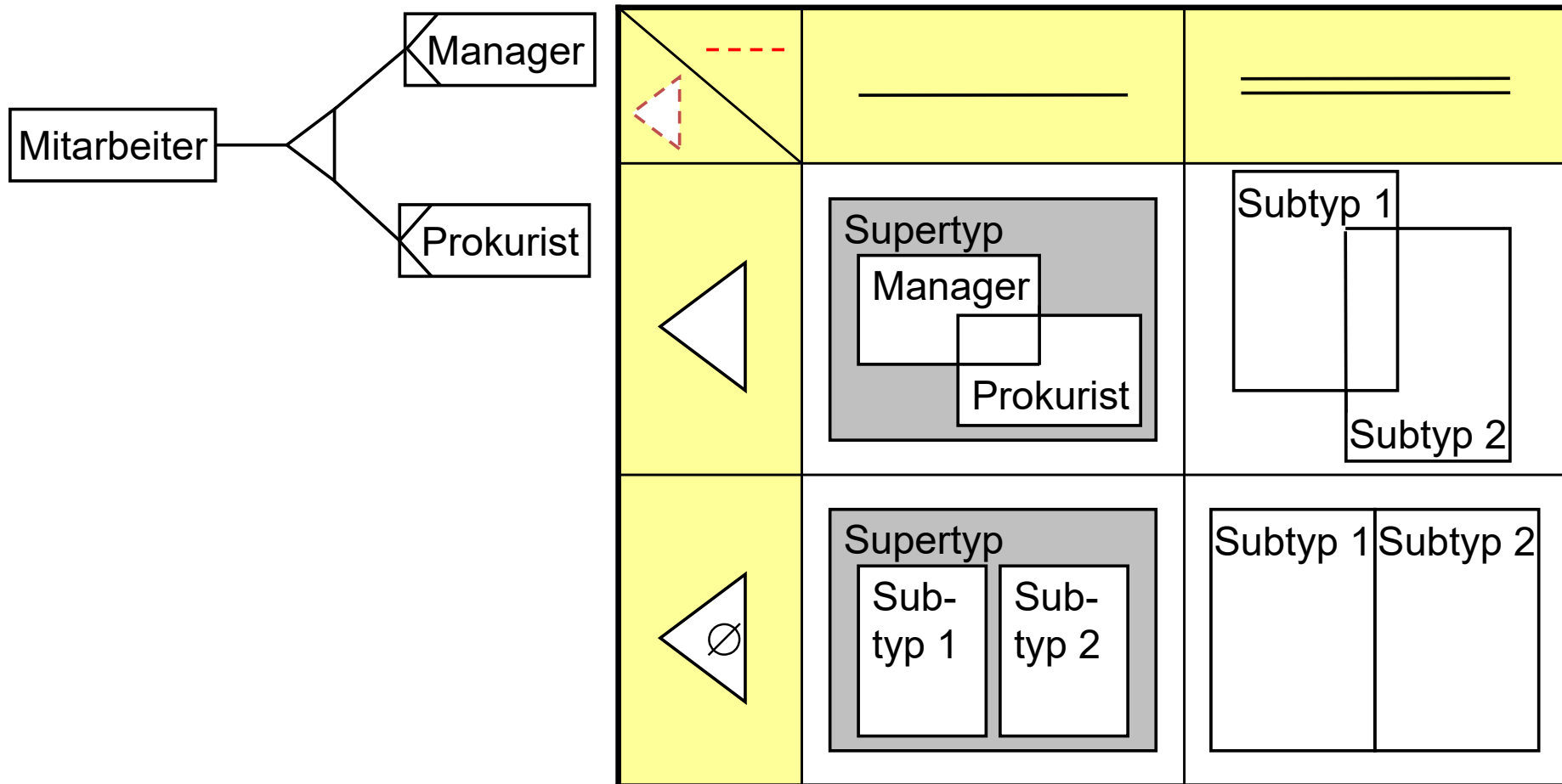
- Eine Kante verläuft von links (Quelle) nach rechts (Senke).
- Die Quelle einer Kante ist immer ein E- bzw. ER-Typ.
- Die Senke einer Kante ist immer ein ER- bzw. R-Typ.
- Ein E-Typ steht ganz links und kann niemals Senke einer Kante sein.
- Ein R-Typ ist die Senke mindestens zweier Kanten.
- Parallele Kanten zwischen zwei Objekttypen sind erlaubt, benötigen aber eine Beschriftung, einen sogenannten Rollennamen.

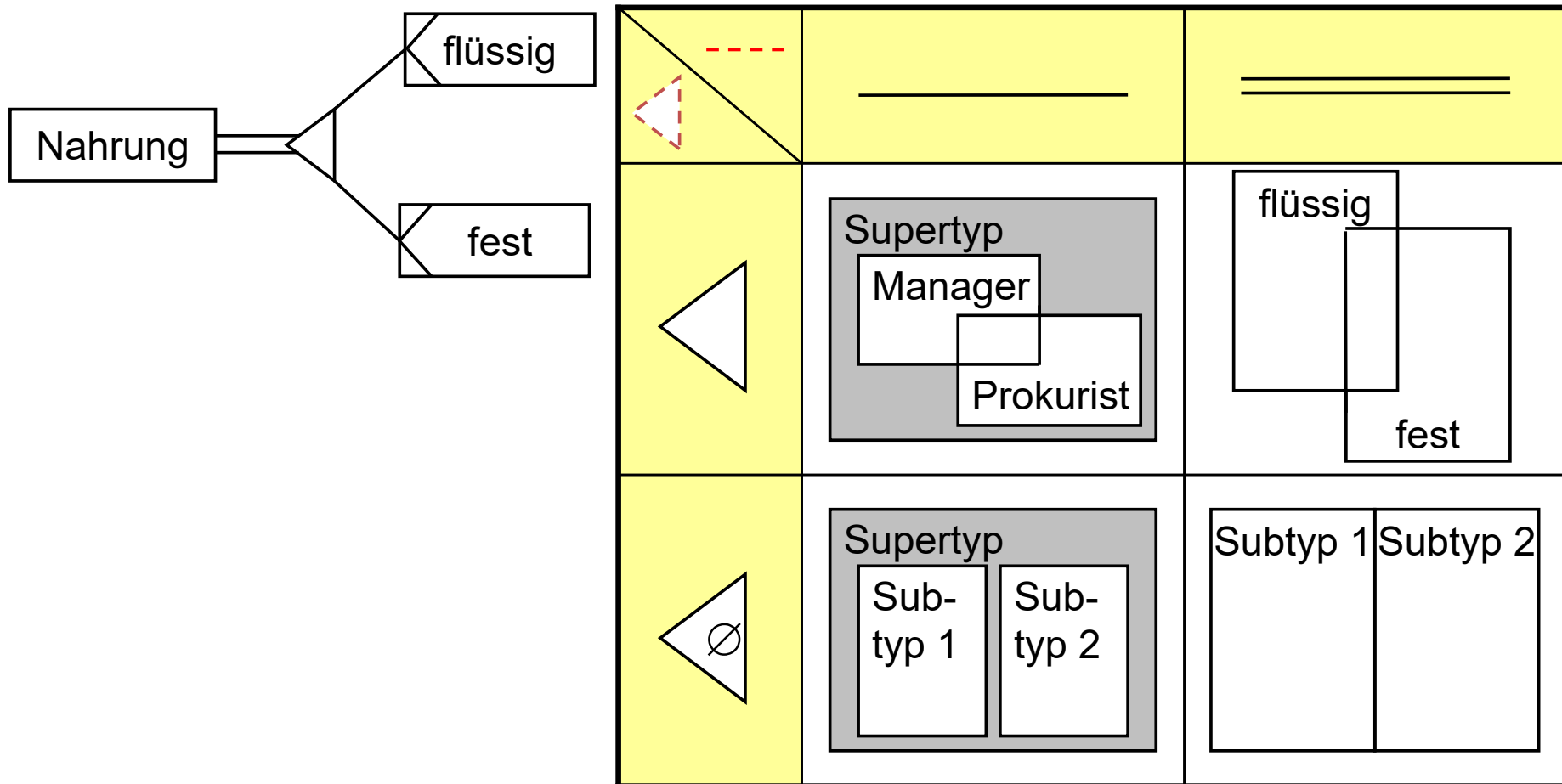
3 Schlüsselvererbung im SER-Modell

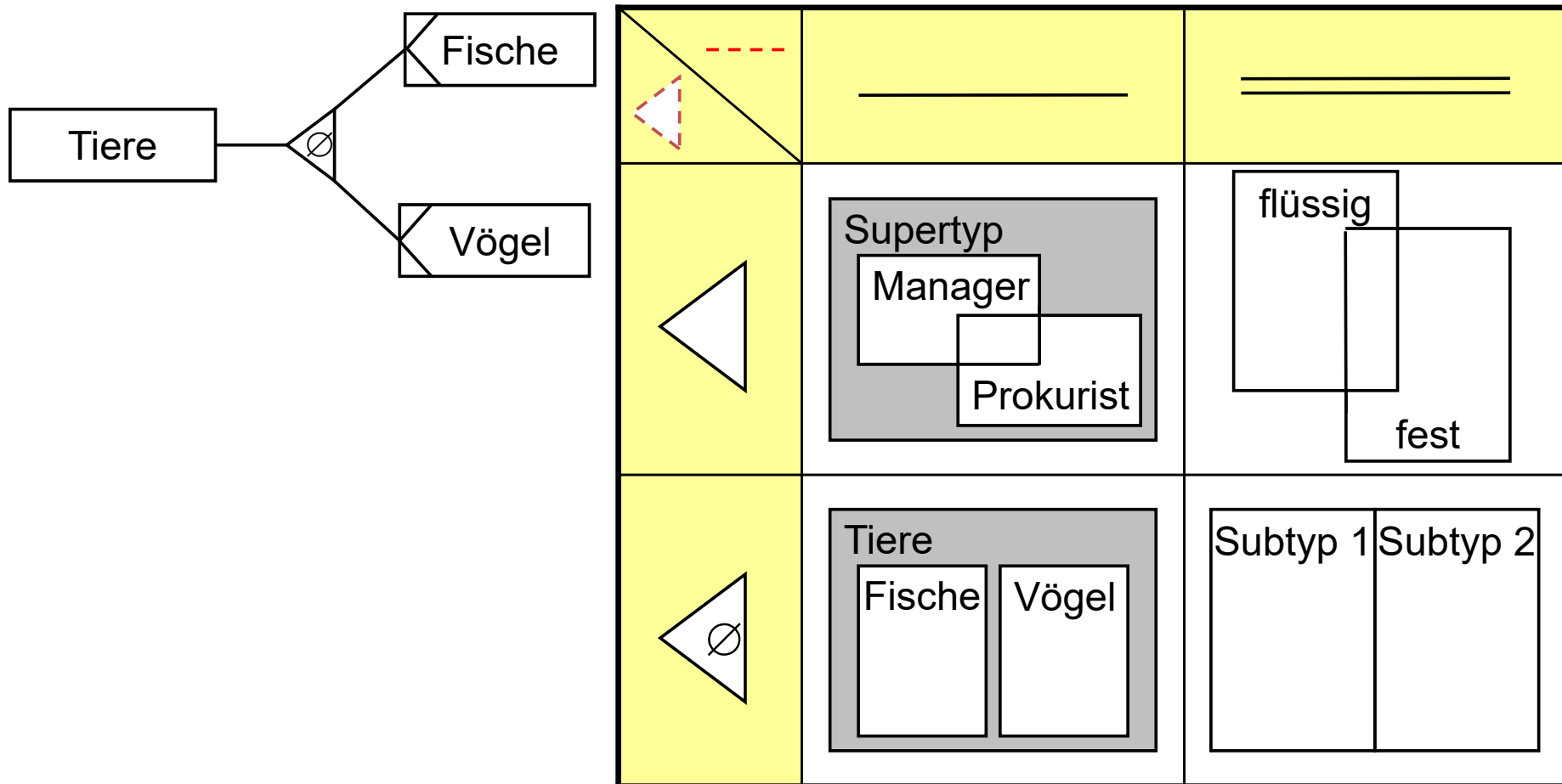
- **Vererbungsrichtung:** gemäß Existenzabhängigkeit (Links-Rechts-Abhängigkeit)
- **Vererbungsobjekt:** Schlüsselattribute
- **Vererbungsart**
 - Primary-Key-Vererbung (PK-Vererbung): Geerbtes Attribut ist im Schlüssel
 - Foreign-Key-Vererbung (FK-Vererbung): Geerbtes Attribut ist nicht im Schlüssel
- **Vererbung:** Aufnahme „geerbter“ Attribute in die Attributliste des erbenden Objekttyps.
Dies ist obligatorisch und darum müssen die Attribute nicht explizit angegeben werden.

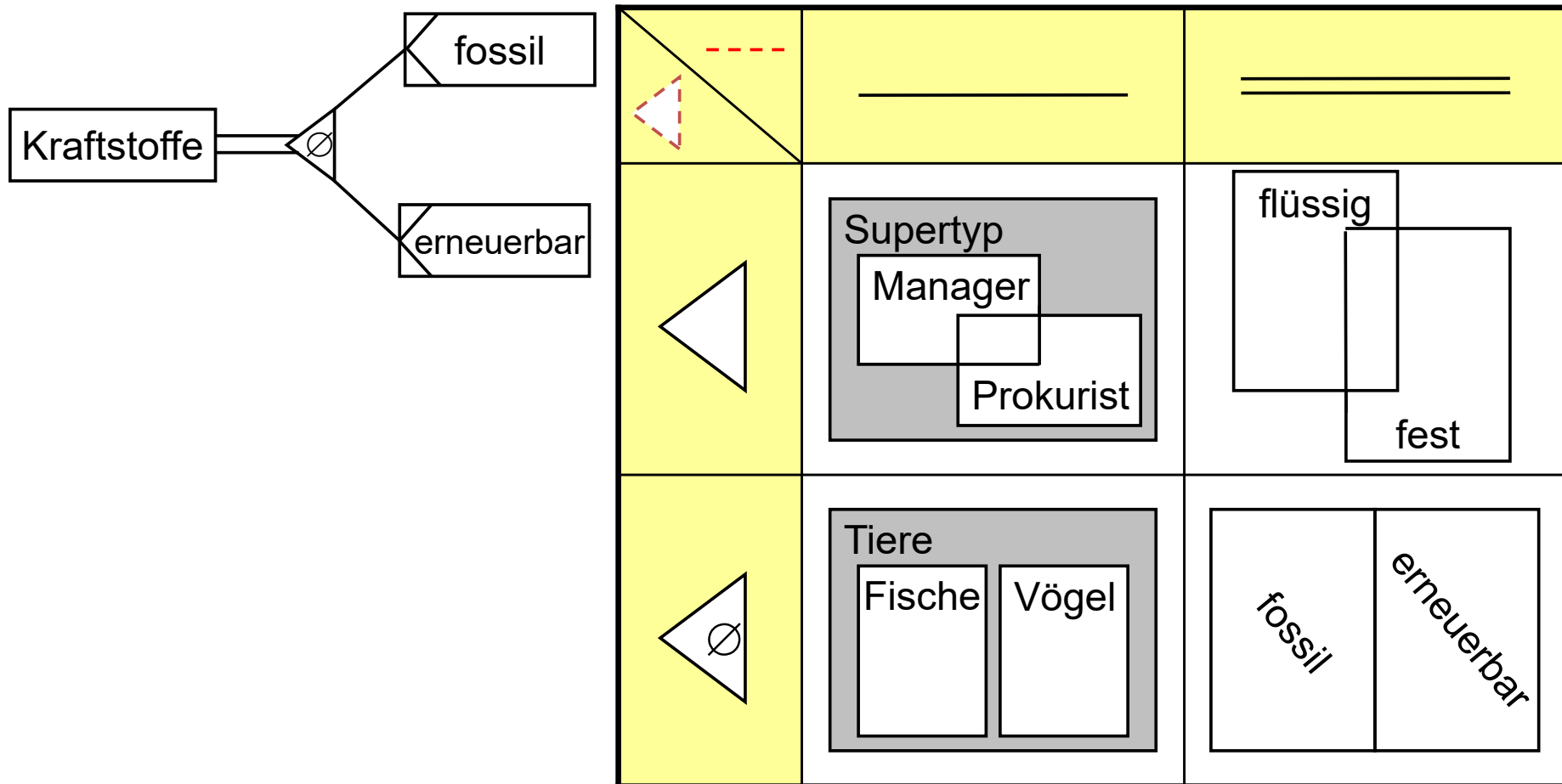
3 Generalisierung



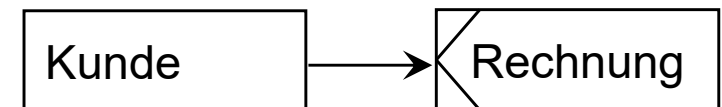
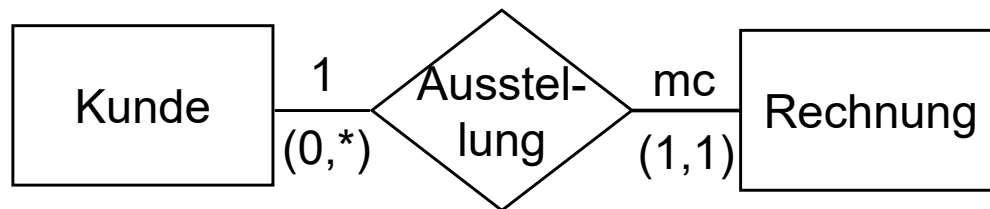
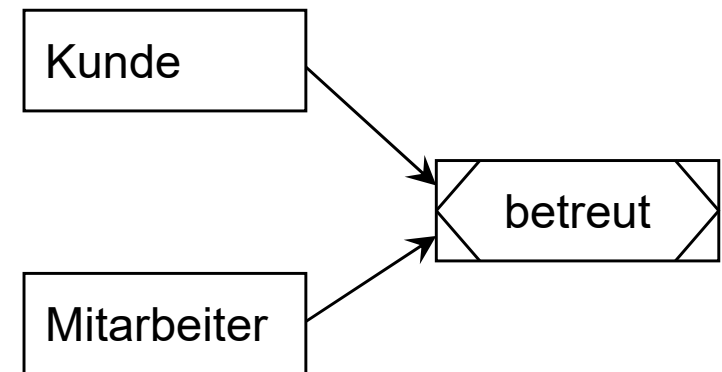
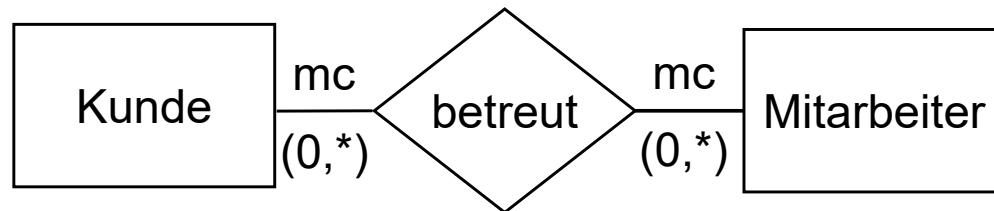




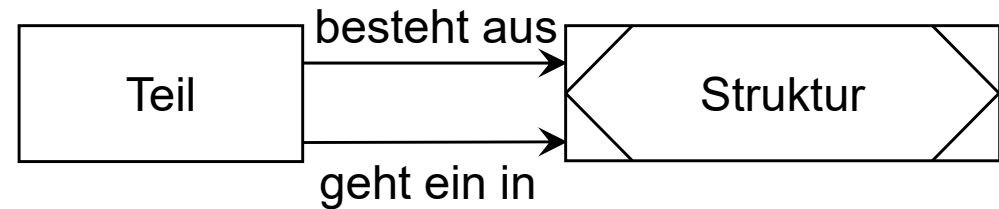
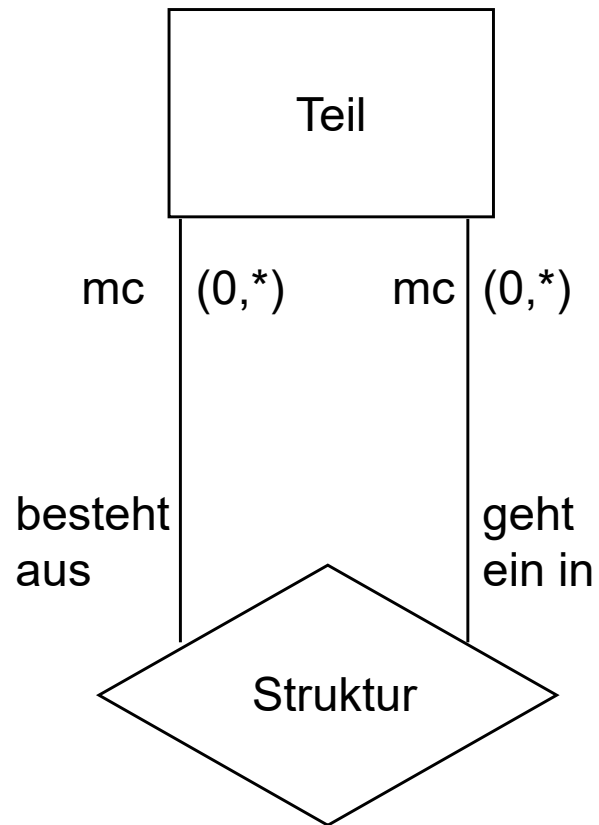




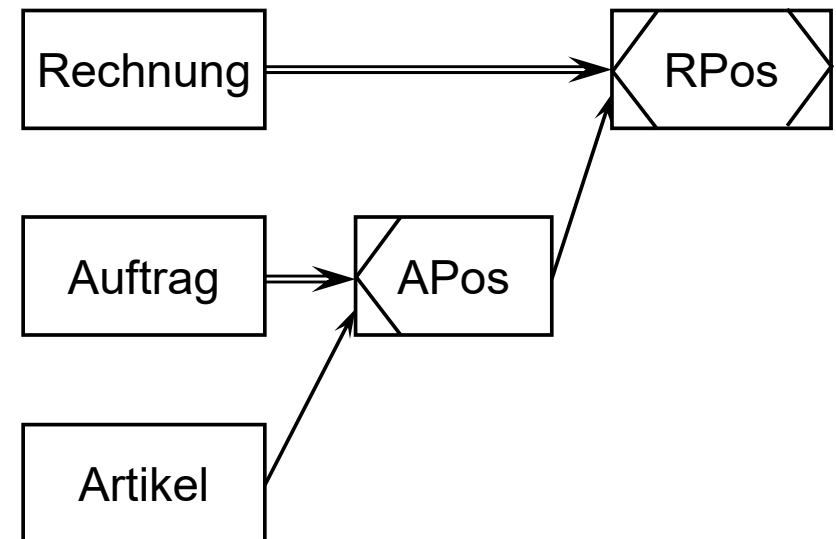
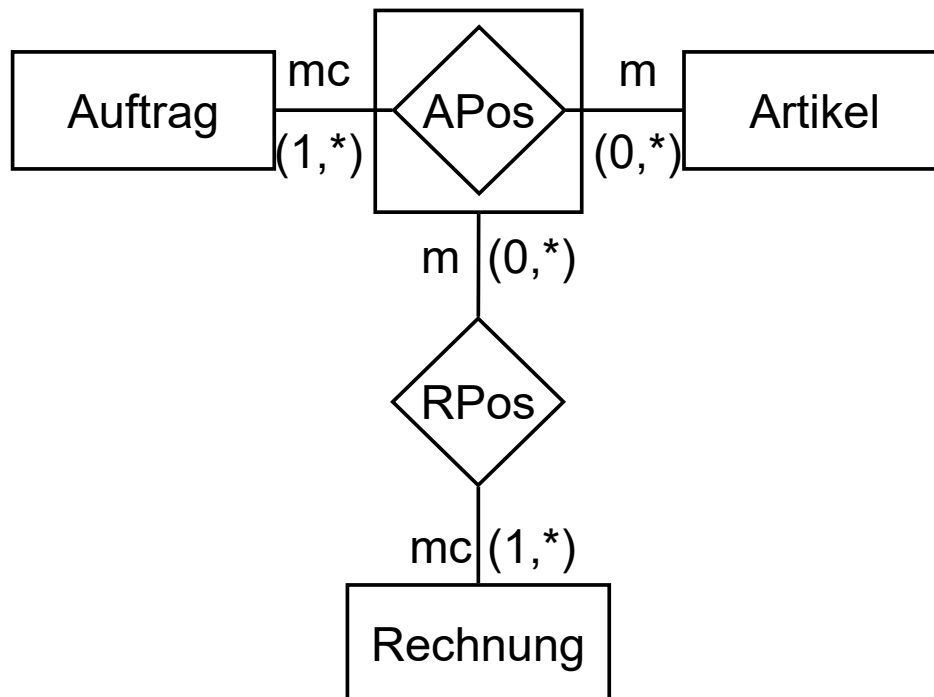
3 Vergleich ERM vs. SERM – Beziehungen



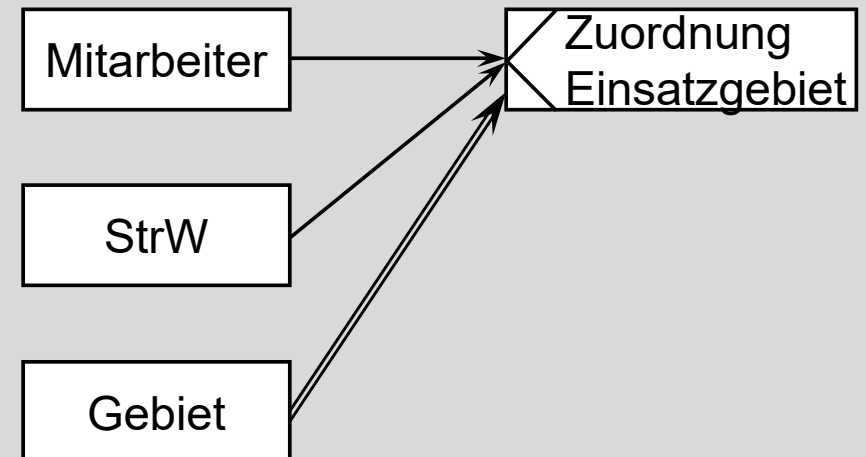
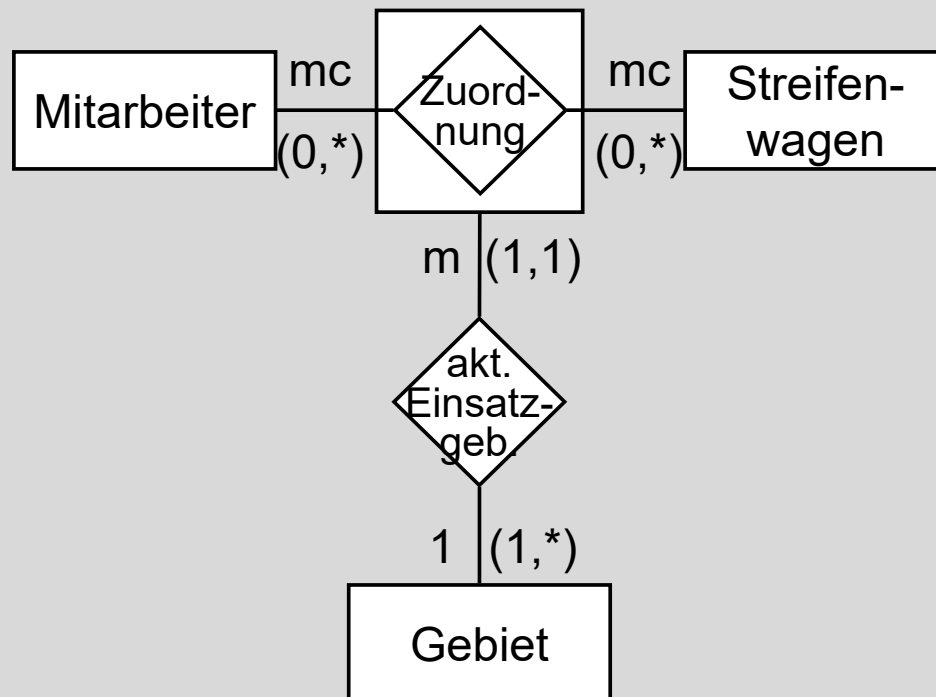
3 Vergleich ERM vs. SERM – Rekursion



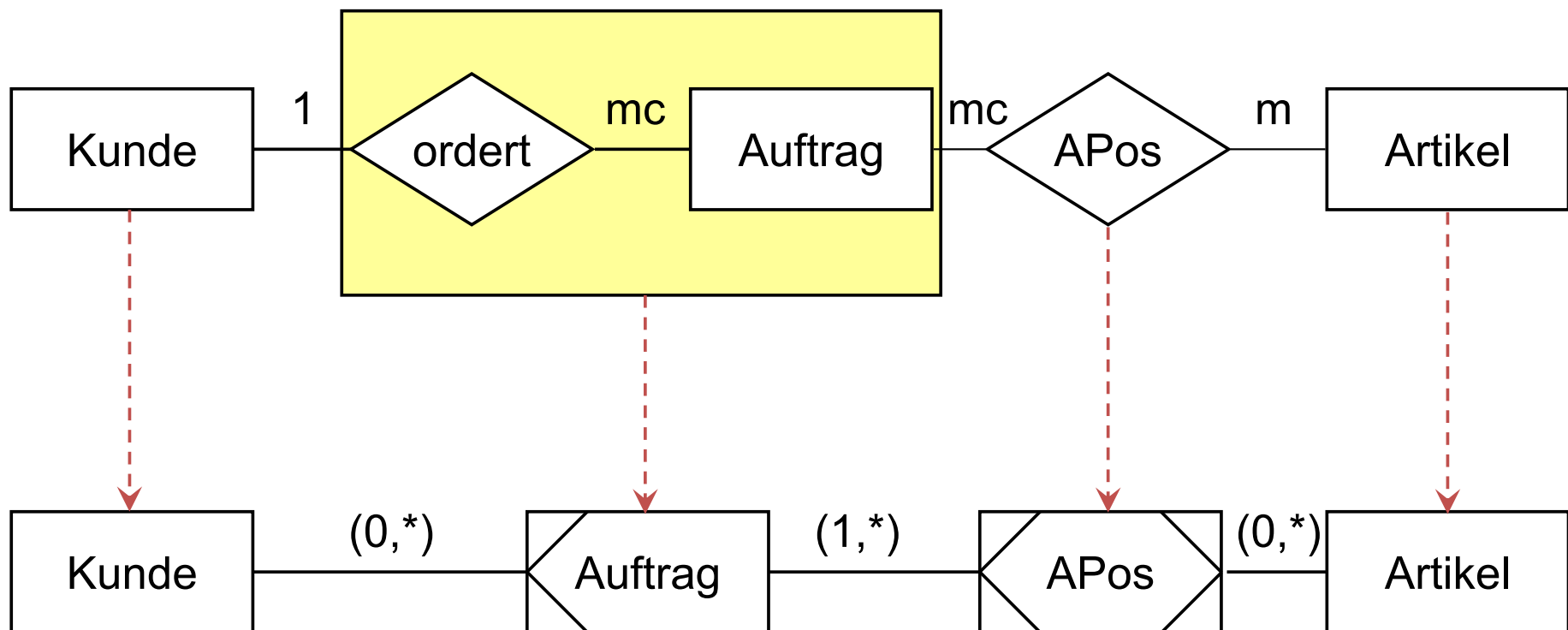
3 Vergleich ERM vs. SERM – Aggregation



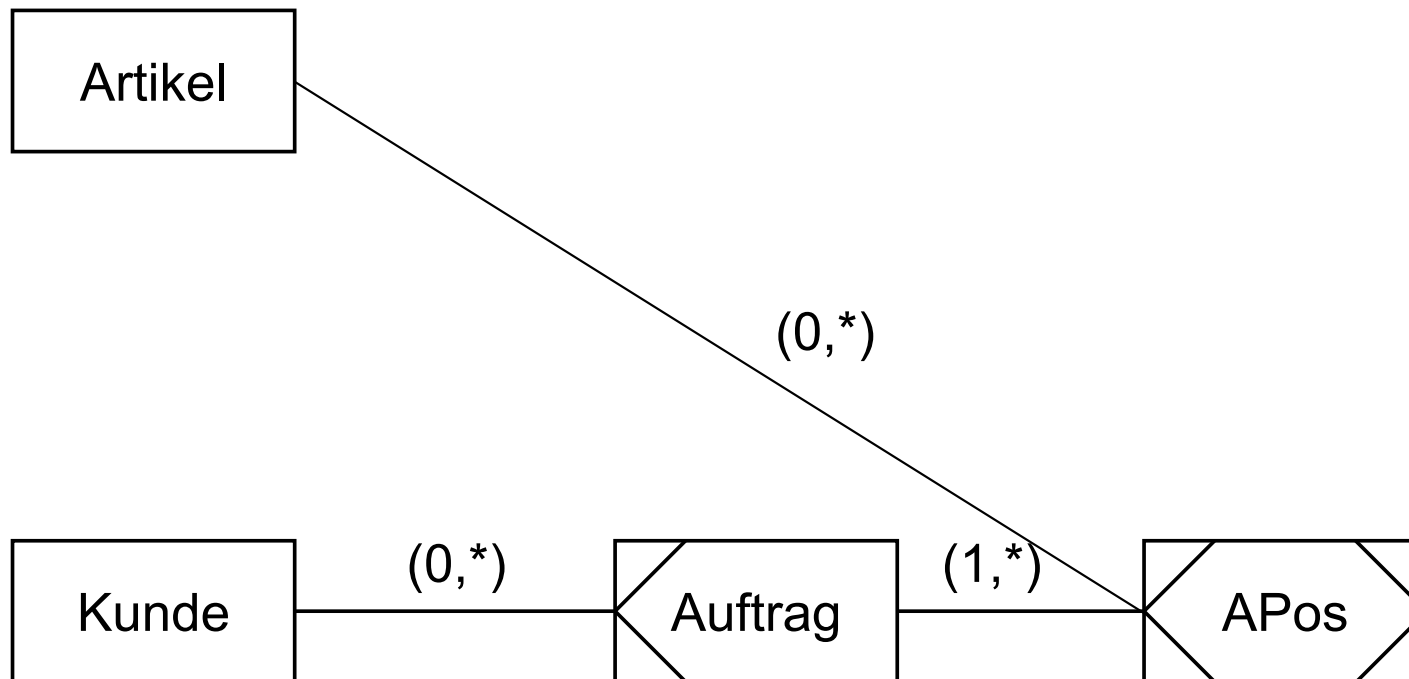
3 Vergleich ERM vs. SERM – Aggregation



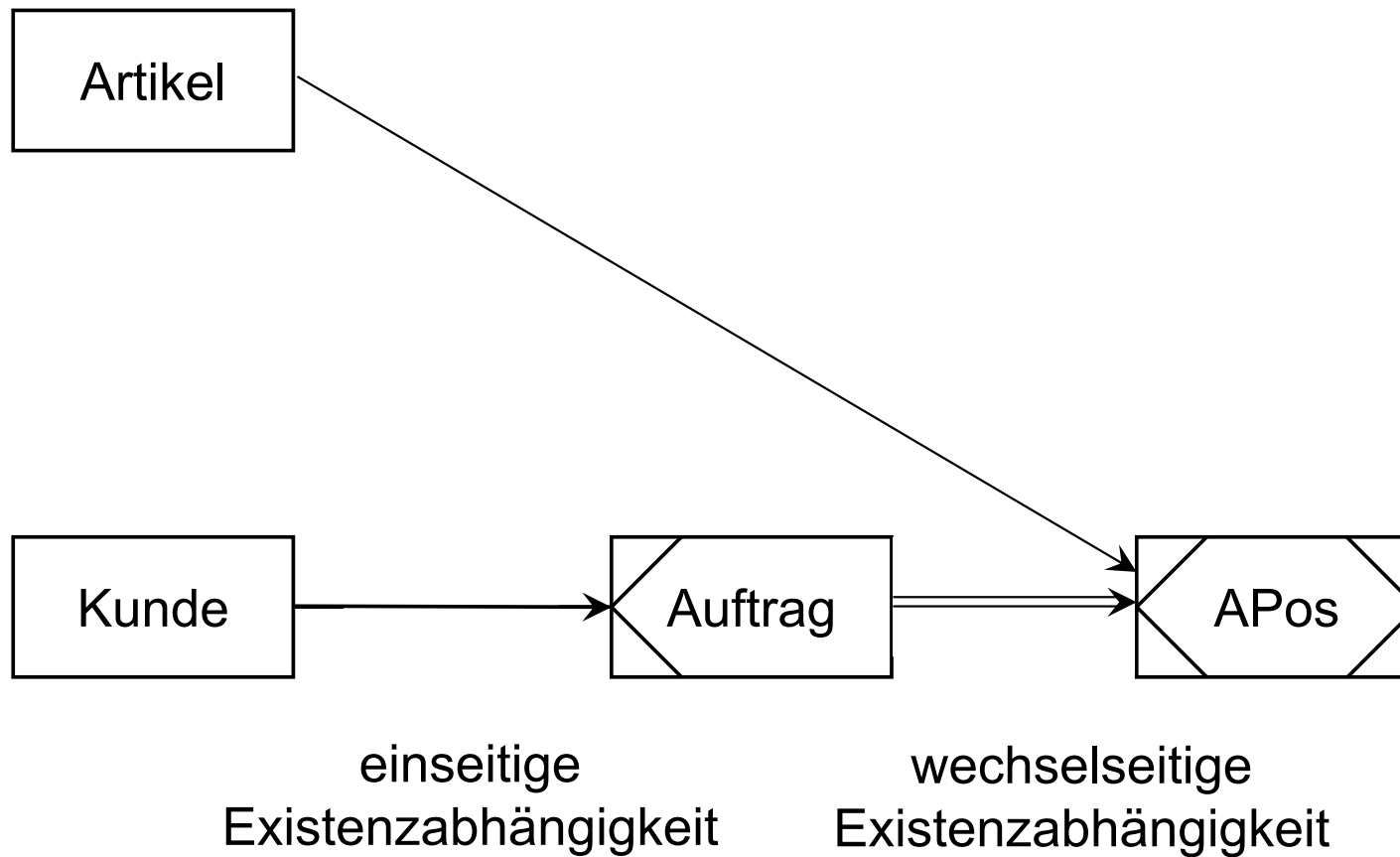
3 Vergleich ERM vs. SERM



3 Vergleich ERM vs. SERM

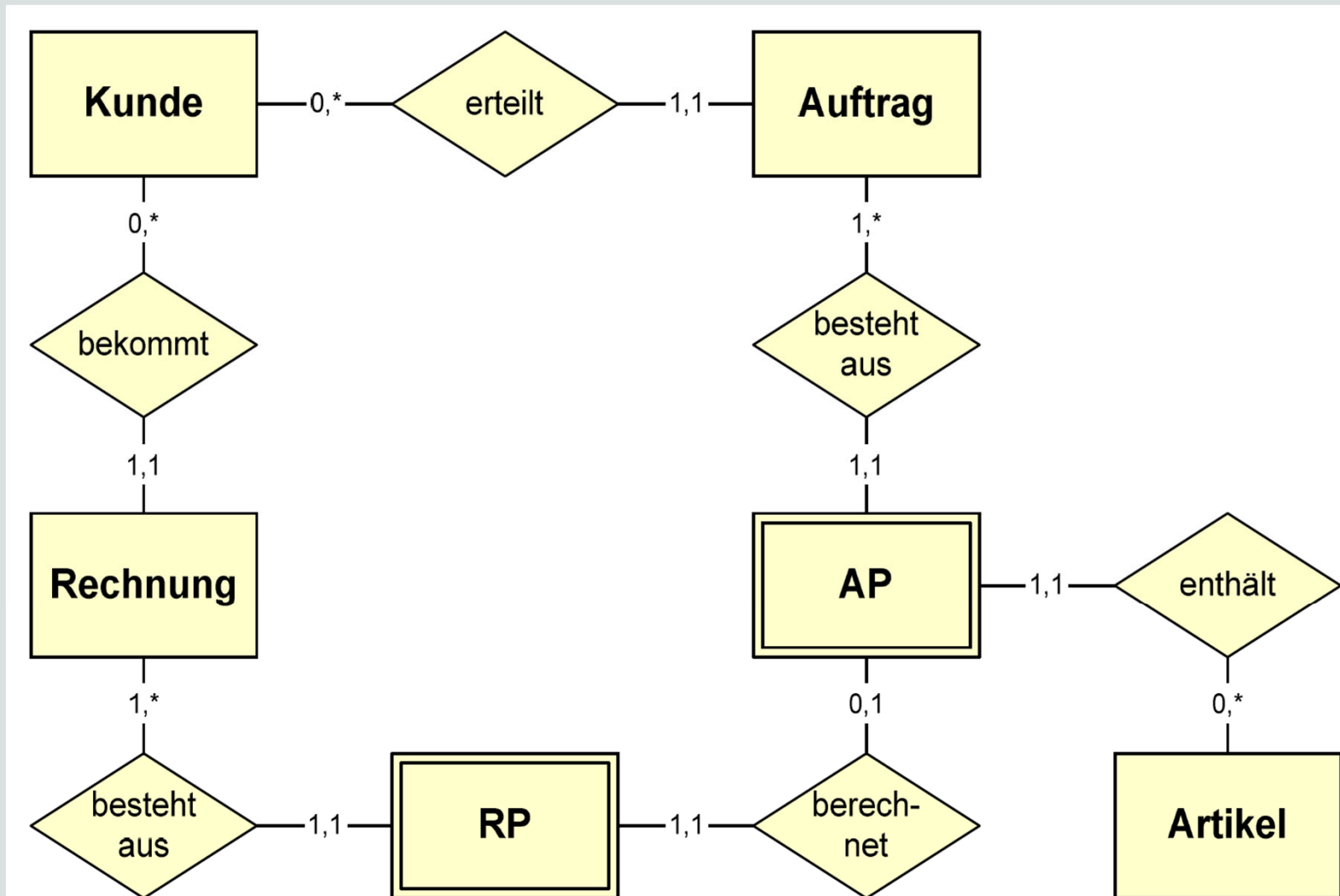


3 Vergleich ERM vs. SERM



3 Aufgabe: SERM -> ERM

Wie sieht das SERM zu dem ERM unten aus:



3 Aufgabe – Hotelkette

Jedes Hotel bietet eine unterschiedliche Anzahl von Ein- und Zweibettzimmern zu festen Preisen an. Alle Zimmer besitzen eine beliebige Anzahl unterschiedlicher Ausstattungsmerkmalen (z.B. Bad, Dusche, Faxanschluss usw.). Neben diesen festen Ausstattungen ist es auch möglich Sonderausstattungen tageweise zu reservieren (z.B. Faxgerät, zweiter Telefonanschluss usw.). Jeder Gast kann eine beliebige Anzahl von Zimmern reservieren. Bei jedem Hotel dieser Hotelkette kann ein Gast die Reservierung von Zimmern und Sonderausstattungen durchführen, die sich auch auf die Kapazitäten anderer Hotels der Hotelkette beziehen können.

Erstellen Sie ein SERM zu dem Szenario.

(AusstattungsNr, Bezeichnung,
Zusatzkosten)

Ausstattung

Normal

(HoNr, ZNr, AusstattungsNr)

(HoNr, HName, Ort, Straße,
HausNr, PLZ)

Hotel

(HoNr, ZNr, ZName,
Kapazität, Preis)

Zimmer

Zusatz

(RNr, RPos,
AusstattungsNr, von, bis)

Gast

(GNr, Name, Vorname
Ort, PLZ, Straße, HausNr)

Reser-
vierung

(RNr, Datum, HoNr, GNr)

Position

(RNr, RPos, von, bis,
HoNr, ZNr)

Unified Modeling Language (UML)

Objektorientierte Modellierungssprache für

- Design / Spezifikation
- Dokumentation
- Konstruktion / Implementierung
- Analyse

Softwarebasierter Systeme

- Einsatz vom Business-Modeling bis zum Coding
- Industriestandard seit UML 1.4
- Urheber
 - Grady Booch, Ivar Jacobson, James Rumbaugh
 - Object Management Group



Diagramme der UML2

Strukturdiagramme

Klassendiagramm

Paketdiagramm

Objektdiagramm

Kompositionsstrukturdiagramm

Komponentendiagramm

Verteilungsdiagramm

Verhaltensdiagramme

Use-Case-Diagramm

Aktivitätsdiagramm

Zustandautomat

Interaktionsdiagramme

Sequenzdiagramm

Kommunikationsdiagramm

Timingdiagramm

Interaktionsübersichtsdiagramm

- Datenmodellierung: Modellierung statischer Strukturen
- ⇒ Objektdiagramm
- ⇒ Klassendiagramm

3 Objekte & Klassen

Objekt

Eine im laufenden System konkret vorhandene Einheit
Instanz einer Klasse

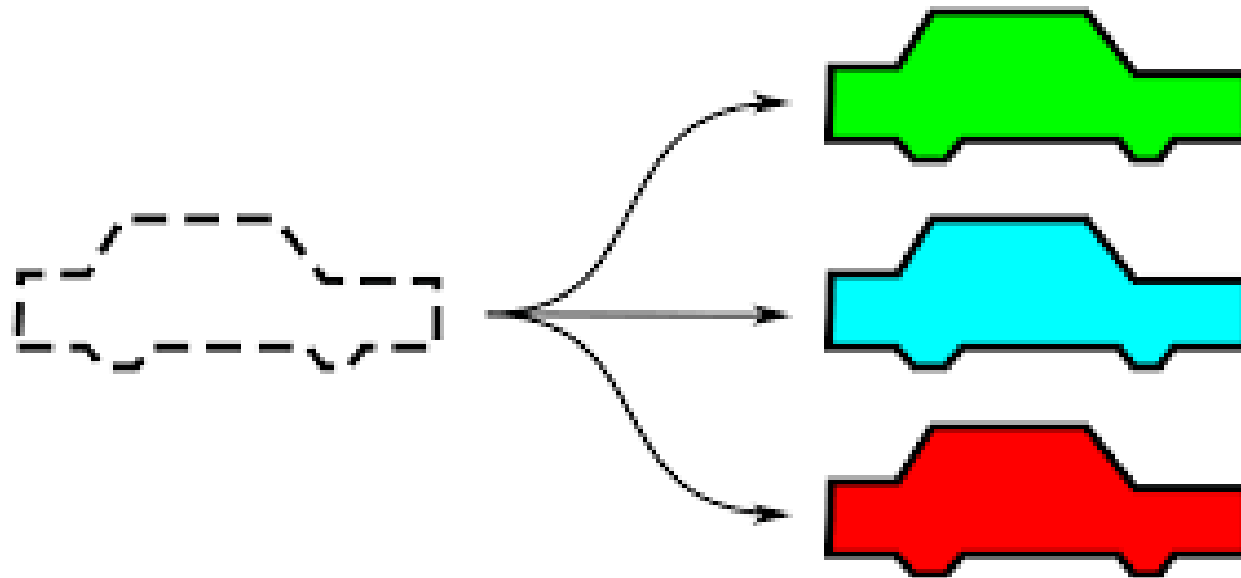


Klasse

Schablone / Typ für Objekte
Definition der Attribute, Operationen und der Semantik
einer gleichartigen Menge an Objekten



- Objektmodell: Objektdiagramm
- Klassenmodell: Klassendiagramm
- Datendiktionär (data dictionary, repository, glossary)
 - Begriffsdefinitionen und -abgrenzungen



Auto im Klassenmodell Auto im Objektmodell

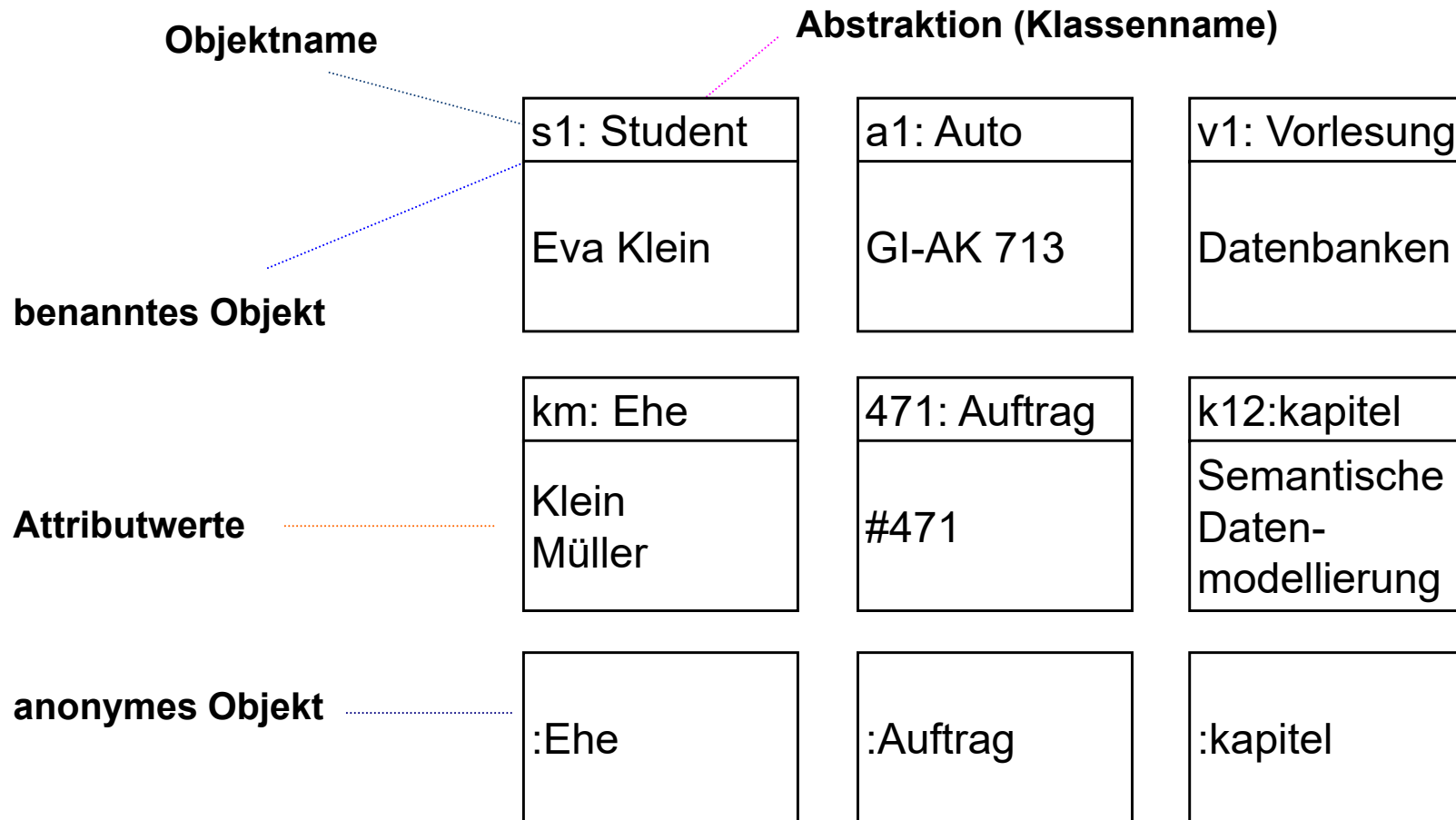


Objektdiagramm

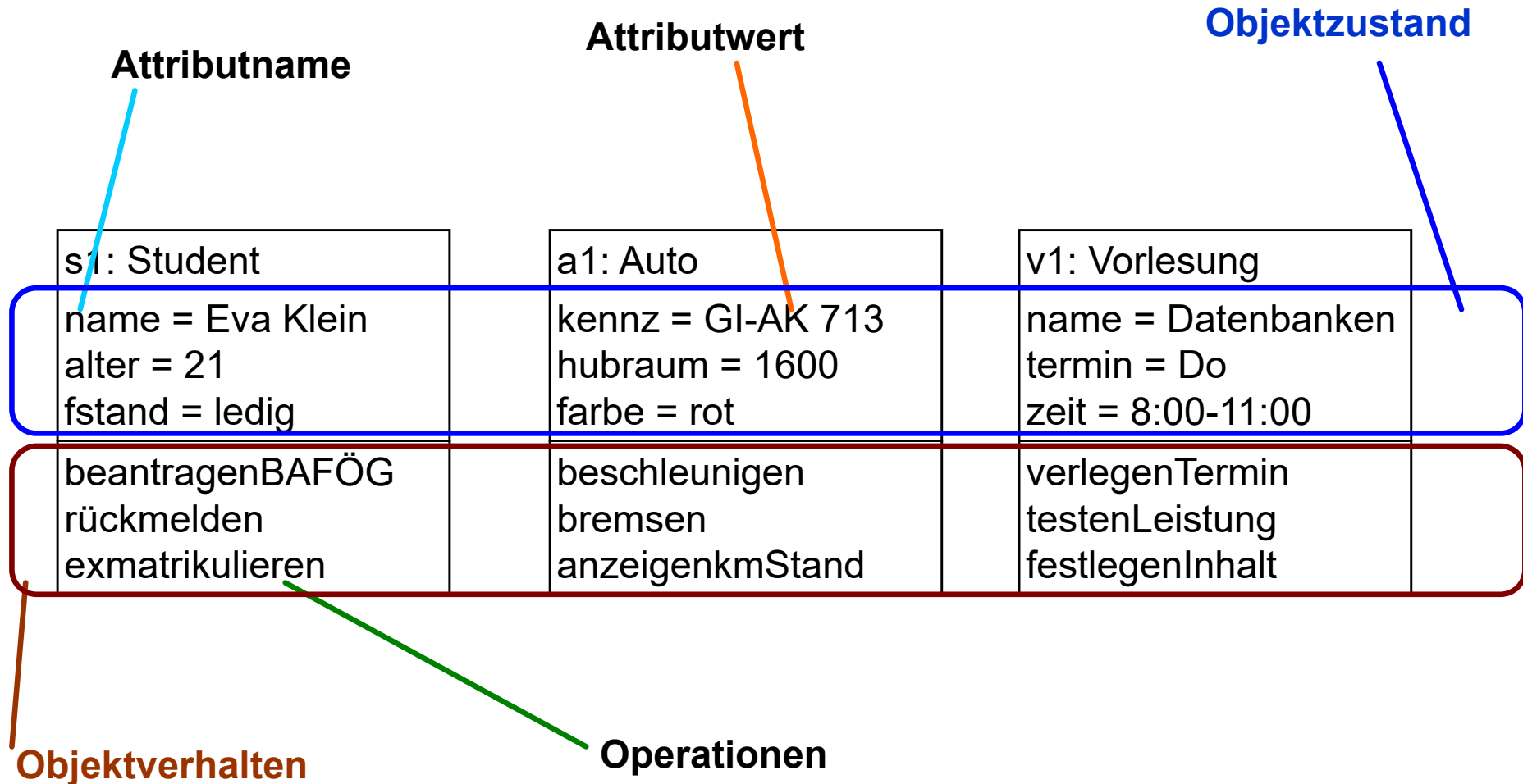
Momentaufnahme des Zustands eines Systems zu einem Zeitpunkt zur Visualisierung, Spezifikation und Dokumentation der Existenz bestimmter Instanzen und deren Beziehungen untereinander

- Modellierung der Instanzen von Klassen und deren Beziehungen
- Enthaltene Elemente
 - Objekte
 - Objektbeziehungen
- Betrachtung eines Szenarios und der daran beteiligten Objekte, deren Attributwerte und aktueller Beziehungen

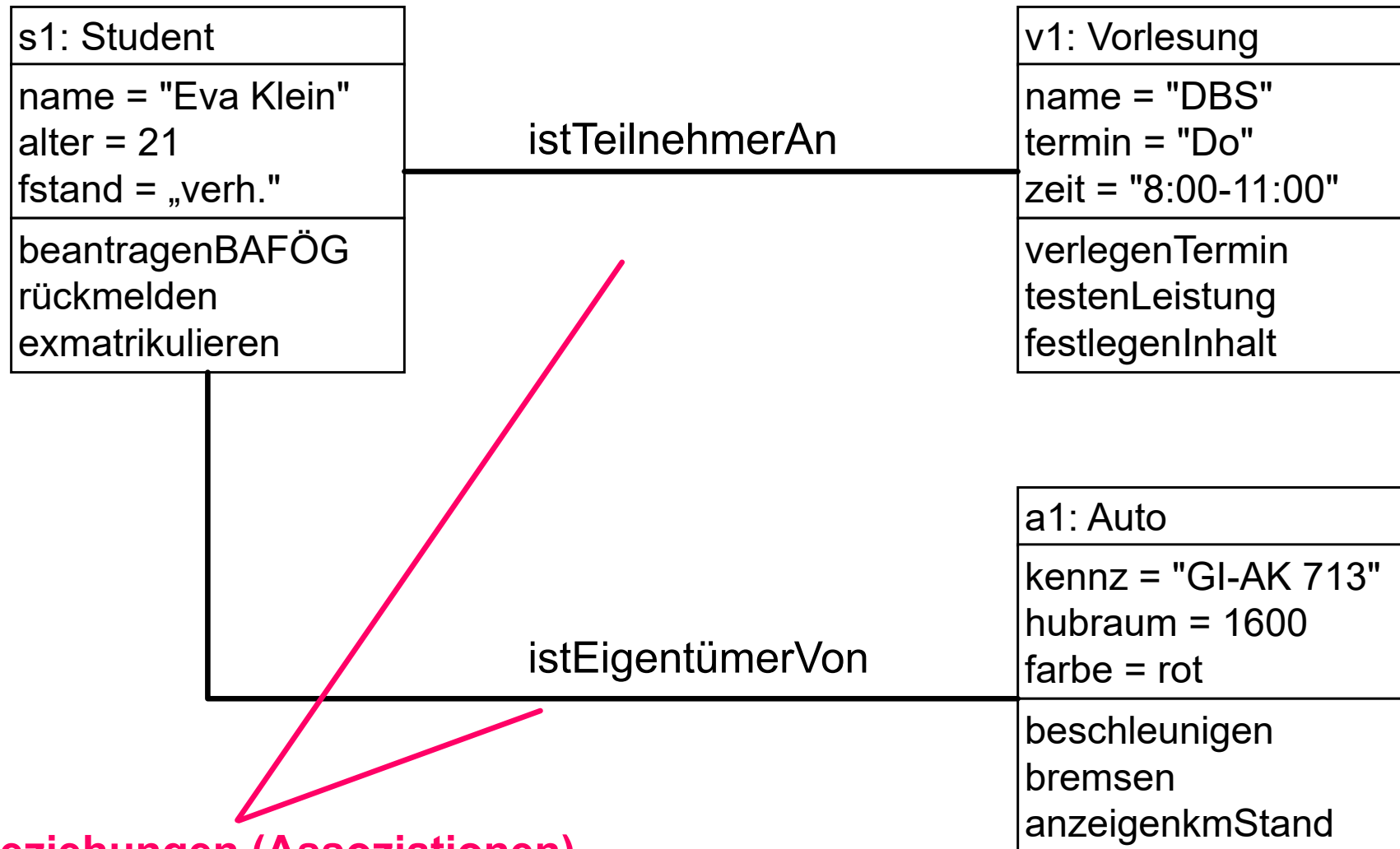
3 Objektdiagramm – Beispiele für Objekte



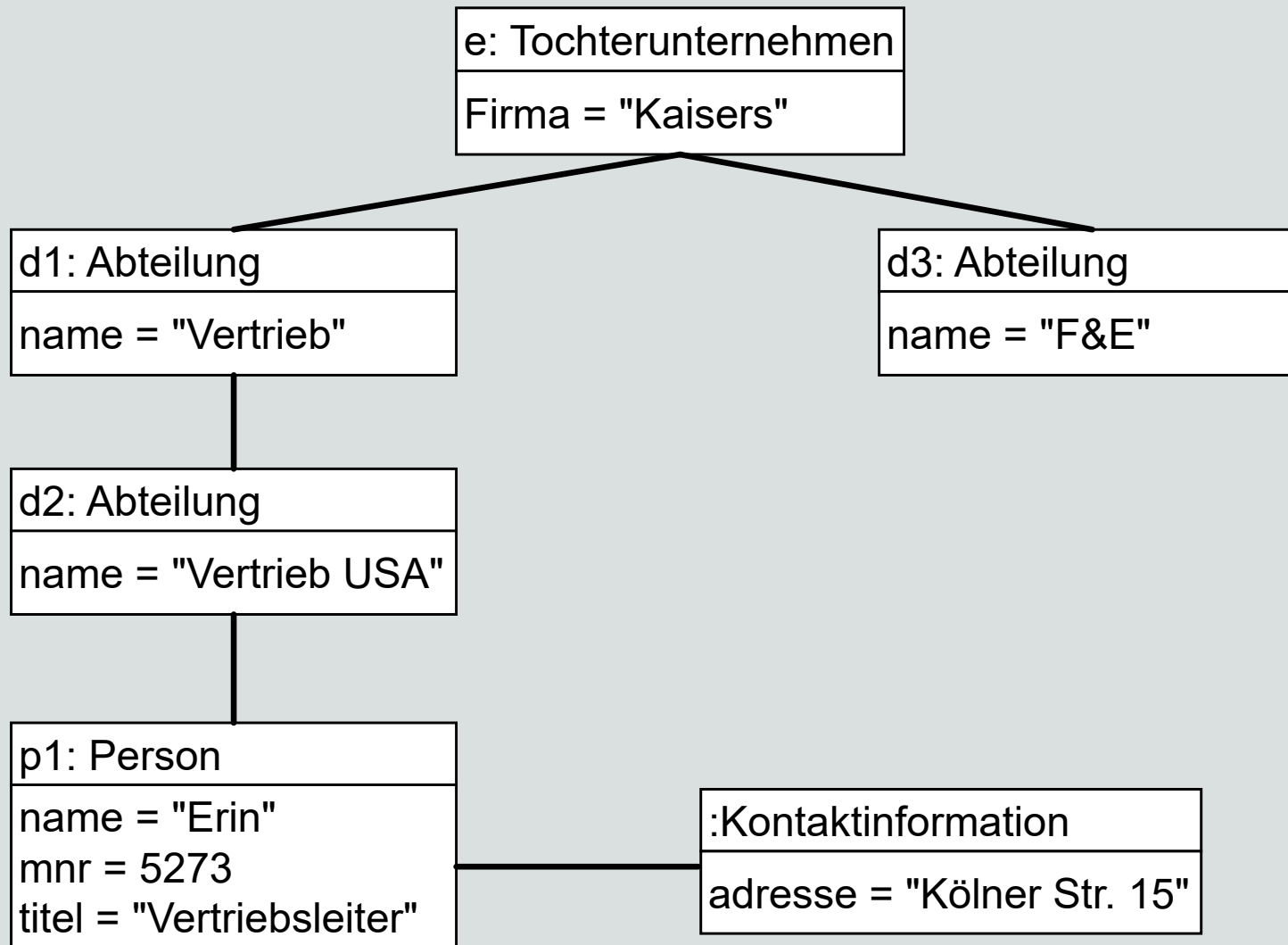
3 Objektdiagramm – Beispiele für Objekte



Objektdiagramm – Beispiel für Objektbeziehungen



Beziehungen (Assoziationen)



3 Zusammenfassung Objektdiagramm

- Zentrale Frage:
Welche innere Struktur besitzt ein System zu einem bestimmten Zeitpunkt zur Laufzeit?
Schnappschuss (eines Klassendiagramms)
- Stärken
 - Zeigt Objekte und deren Attributbelegungen zu einem bestimmten Zeitpunkt
 - Beispielhafte Verwendung zur Veranschaulichung
 - Detailniveau wie im Klassendiagramm
 - Sehr gute Darstellung von Mengenverhältnissen

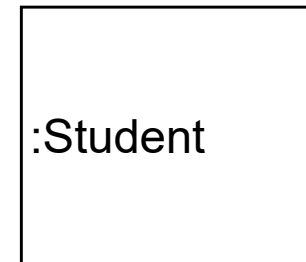
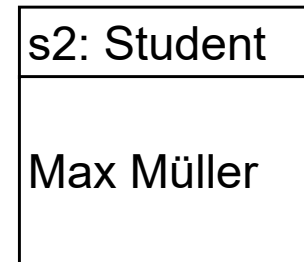
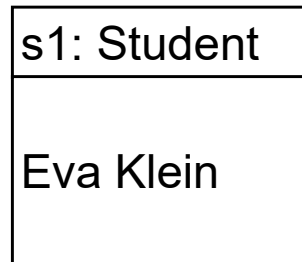
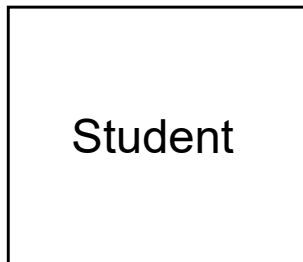


Klassendiagramm

Beschreibung der generellen statischen Struktur eines Systems auf Objekttyp-Ebene (=Klassenebene)

- Klassen
- Schnittstellen
- Kollaborationen
- Beziehungen
 - Aggregation
 - Komposition
 - Generalisierung

3 Klasse – Beispiel



- Mindestens ein Attribut
 - Eindeutig identifizierbar
 - Mindestens ein Objekt; in der Datenmodellierung ≥ 1
 - Attribute, die eigenständige Objekte identifizieren, werden als eigene Klassen modelliert
- Ausnahmen:
- Erhöhung der Übersichtlichkeit
 - Objekte liegen außerhalb des Erhebungsbereichs
- Lokalisierungsprinzip: Minimierung der Nachrichtenbeziehungen zwischen Klassen und Objekten

Klassenname
+Attribut 1: Typ1 = Default1 - Attribut 2: Typ2 = Default2 #Attribut 3: Typ3
Methode1(Parameter) Methode2(Parameter)

- Klassenart
 - {abstract}
 - <<interface>>
 - Parametrisiert
- Attribute
 - + public
 - - private
 - # protected
 - «key»
 - readonly
 - / berechnet
 - Klassenattribut

3 Vergleich UML – ERM

UML

- Objektebene
- Attribute & Methoden
- Beziehungen
 - Klassen
 - Assoziationen
- Vererbung
- Kardinalitäten
- Zusammensetzung
 - Komposition
 - Aggregation
- Angabe der Leserichtung
- Unterschiedliche, formale Modellierungsmöglichkeiten

ERM

- Keine Objektebene
- Attribute
- Beziehungen
 - R-Typen
 - Aggregation
- Generalisierung/Spezialisierung
- Kardinalitäten
- Zusammensetzung
 - R-Typen
- Keine Leserichtung
- Einheitliche Modellierung

- Was versteht man in der Informatik unter einem Modell?
- Geben Sie ein Vorgehensmodell für einen Datenbankentwurf an.
- Erläutern Sie die zentralen Begriffe eines ERM- bzw. UML-Datenmodells.
- Geben Sie Beispiele für die 10 verschiedenen Beziehungstypen (Kardinalitäten, Assoziationstypen) an.
- Was versteht man unter einem ERM? Welche Ziele sind mit diesen ER-Diagrammen verbunden?
- Erläutern Sie, was man unter einem „schwachen Entitätstyp“ versteht.
- Geben Sie Beispiel für Generalisierung, Spezialisierung und Aggregation in ER-Diagrammen an
- Wie nennt man einen Entitätstyp, der eine Beziehung auf sich selbst besitzt? Geben Sie ein Beispiel an.
- Geben Sie die Vorgehensweise zum Entwurf eines ER-Modell an.
- Was versteht man unter einem Objekt- und unter einem Klassendiagramm?
- Geben Sie ein Beispiel für ein UML-Diagramm mit Assoziationsklasse(n) an.
- Wie unterscheidet sich eine Aggregation und eine Komposition als Zerlegung in einem UML-Diagramm?
- Vergleichen Sie UML- SERM-, und ER-Datenmodelle.