

Informatik II: Modellierung  
Prof. Dr. Martin Glinz

Kapitel 12

# Abstraktion



Universität Zürich  
Institut für Informatik

---

# Inhalt

---

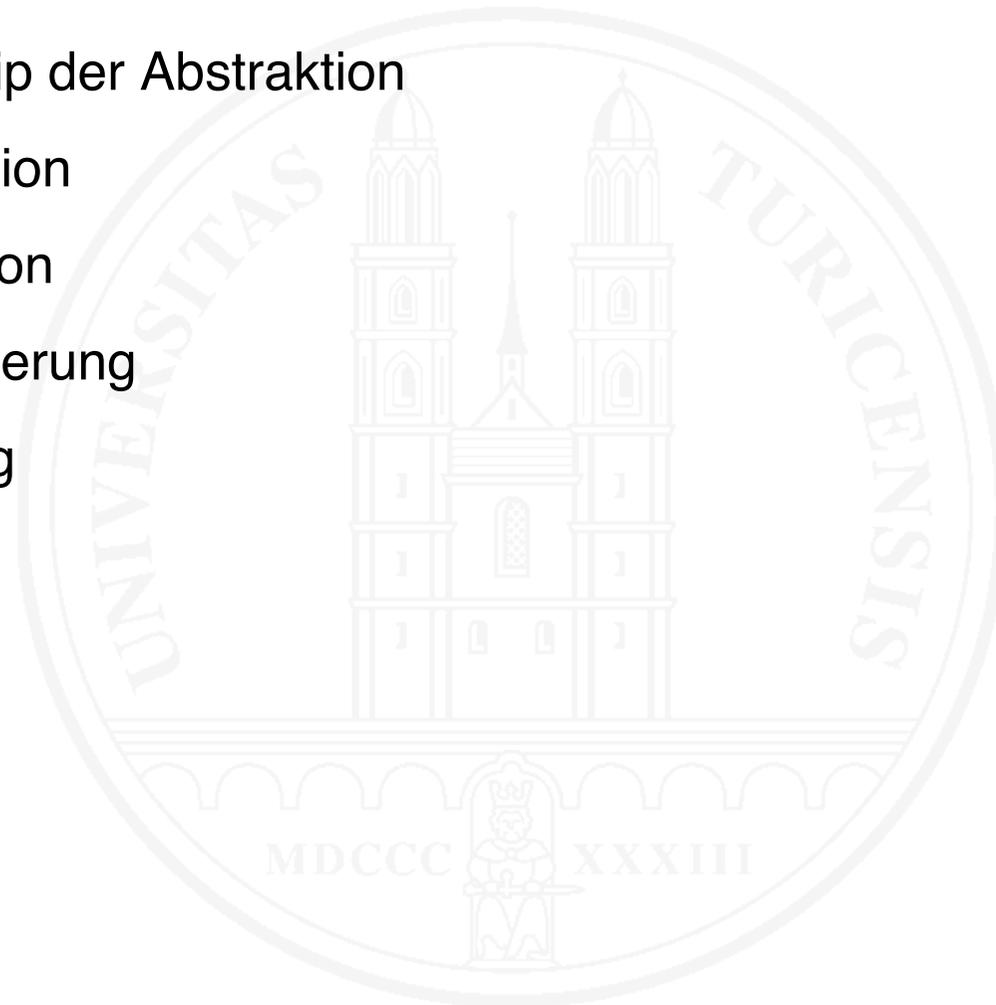
12.1 Das Prinzip der Abstraktion

12.2 Klassifikation

12.3 Komposition

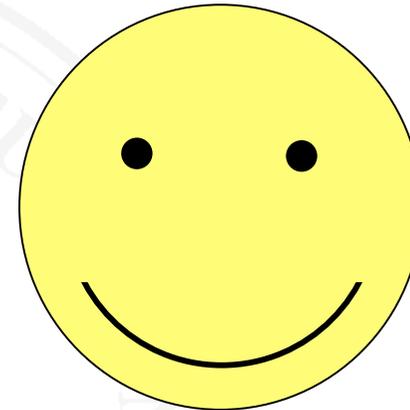
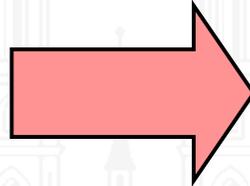
12.4 Generalisierung

12.5 Benutzung



# 12.1 Das Prinzip der Abstraktion

---



- **Abstraktion (abstraction)** – Das **Ableiten** oder **Herausheben** des unter einem bestimmten Gesichtspunkt **Wesentlichen/Charakteristischen/ Gesetzmäßigen** aus einer Menge von Individuen (Dingen, Beobach-tungen, ...)
- Abstraktionen sind ein zentrales Mittel für das **Erstellen** und **Verstehen** von Modellen.

# Abstraktion beim Erstellen von Modellen

---

Abstraktion ist erforderlich:

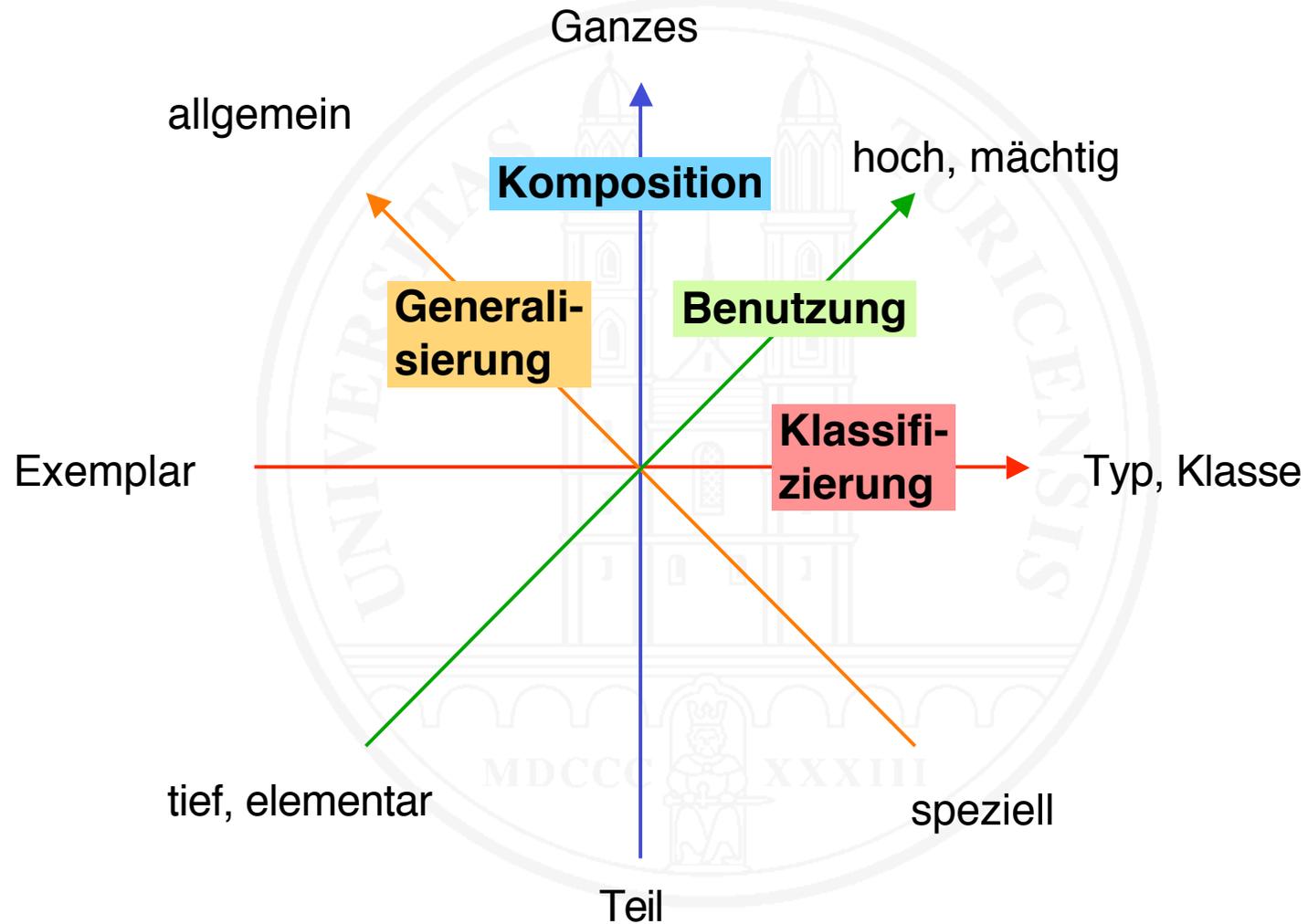
- Zur **Umsetzung der Pragmatik**: Was ist unter welchem Gesichtspunkt wichtig / hervorzuheben?
- Beim **Verkürzen**: welche Eigenschaften des Originals werden nicht modelliert?

# Abstraktion beim Verstehen von Modellen

---

- Große Modelle sind ohne einen klar strukturierten Aufbau **nicht verstehbar**.
- Abstraktion ermöglicht
  - das **Sichtbarmachen** großer **Zusammenhänge** unter **Weglassung** der **Details**
  - die **Darstellung** eines **Details** unter **Weglassung/** starker Vergrößerung **des Rests**
  - die **Herstellung** eines **systematischen Zusammenhangs** zwischen Übersichten und Detailsichten.
- Abstraktion ist notwendig zur Beherrschung der **Komplexität** großer Modelle

# Abstraktionsarten



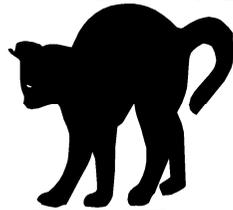
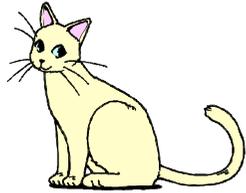
# Abstraktionsarten – 2

---

Die wichtigen **Abstraktionsarten** für Modelle sind:

- **Klassifizierung:** Charakterisierung der Gemeinsamkeiten von Individuen durch Typen oder Klassen
- **Komposition:** Zusammensetzen einer Menge zusammenhängender Individuen zu einem Ganzen
- **Generalisierung:** Verallgemeinerung der Merkmale einer Menge ähnlicher Typen
- **Benutzung:** Nutzung von Leistungen Dritter durch ein Individuum zwecks Erbringung eigener, höherwertiger Leistungen
- Alle vier Abstraktionsarten sind **unabhängig** voneinander

# 12.2 Klassifikation



**Katze**  
Name: Zeichenkette  
Geschlecht: (w, m)  
Alter: Jahre  
Rasse: Zeichenkette

Instanzen

Typ / Klasse

# Klassifikation – 2

---

- **Klassifizierung (classification)** – Charakterisierung einer Menge von Individuen (Exemplaren) mit gemeinsamen Attributen durch einen Typ oder eine Klasse.
- **Typ (type)** – Intensionale Interpretation: Der Typ beschreibt den Aufbau der zulässigen Exemplare dieses Typs
- **Klasse (class)** – Extensionale Interpretation: Die Klasse ist ein Behälter für ihre aktuell vorhandenen Exemplare. Aber: Klassen werden oft gleichzeitig auch intensional interpretiert.
- **Instanz (instance)** – Exemplar eines Typs bzw. einer Klasse

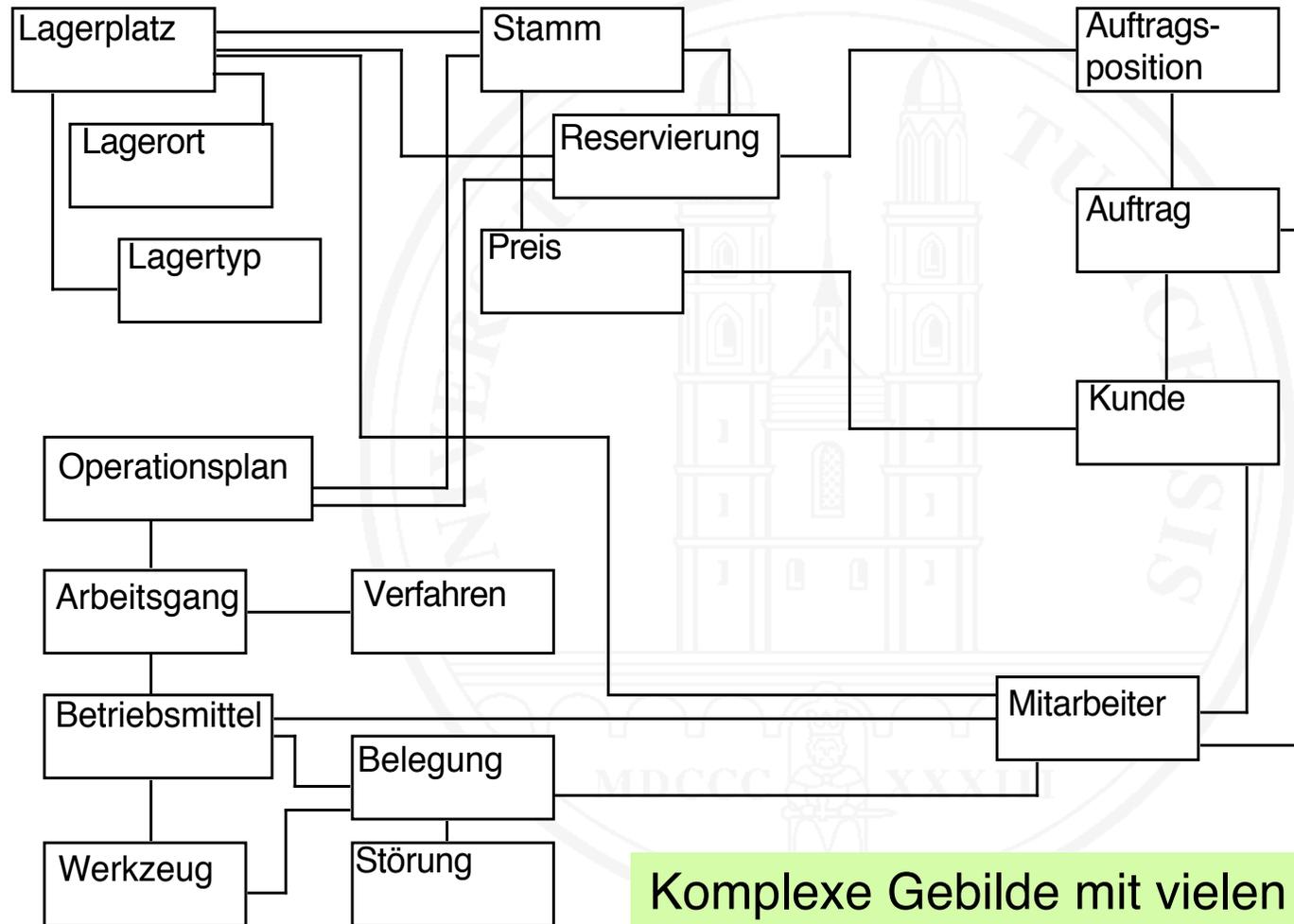
# Klassifikation – 3

---

## Der Typ bzw. die Klasse

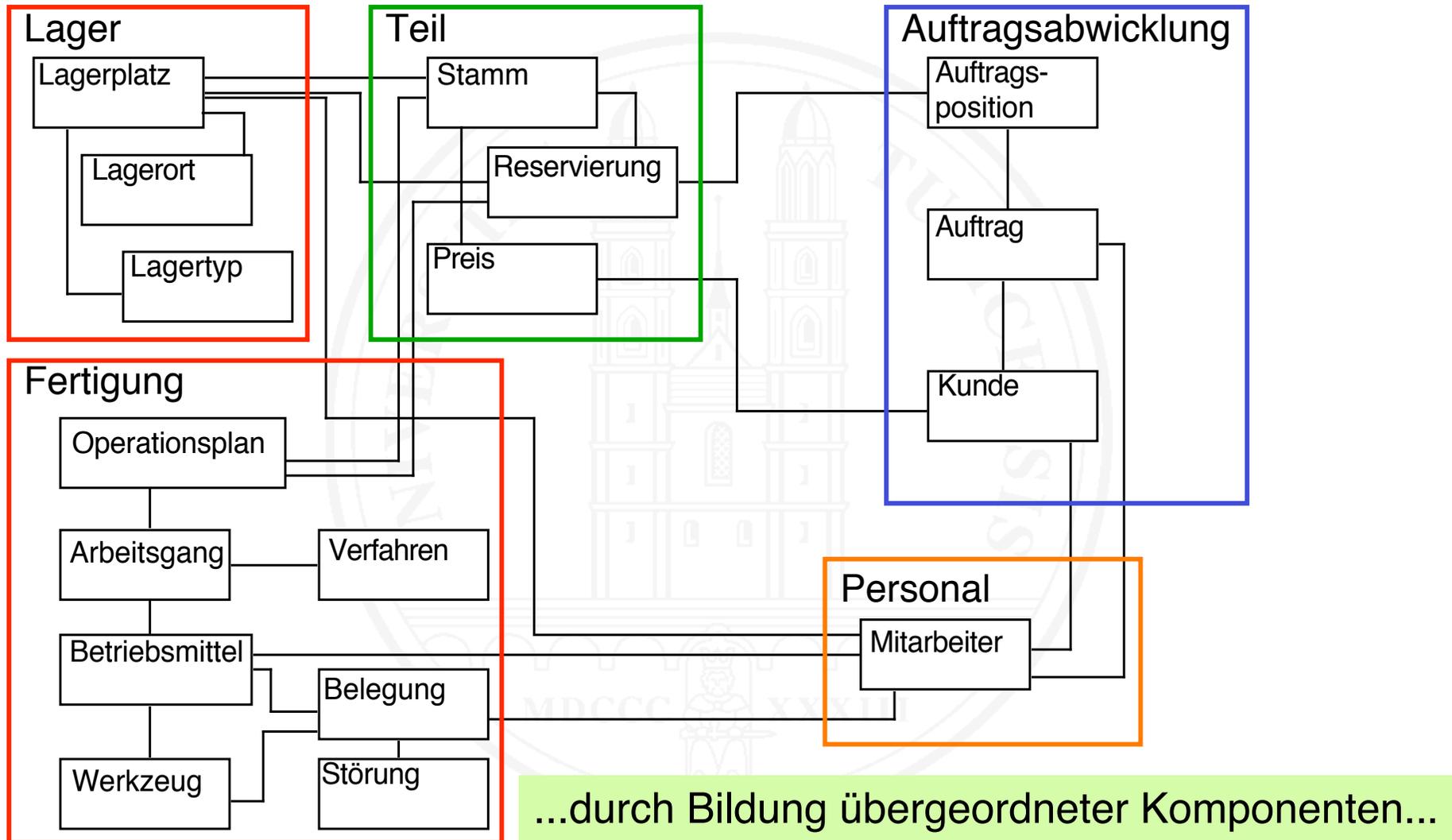
- beschreibt die **Attribute** der Exemplare
  - beschreibt den **zulässigen Wertebereich** zu jedem Attribut
  - **lässt** alle Information zu **individuellen Attributausprägungen** der Exemplare **weg** (diese wird zu Wertebereichen abstrahiert)
- ⇒ **Zentrale Abstraktion** bei jeder Modellbildung (vgl. insbesondere Kapitel 3 und 8)

# 12.3 Komposition



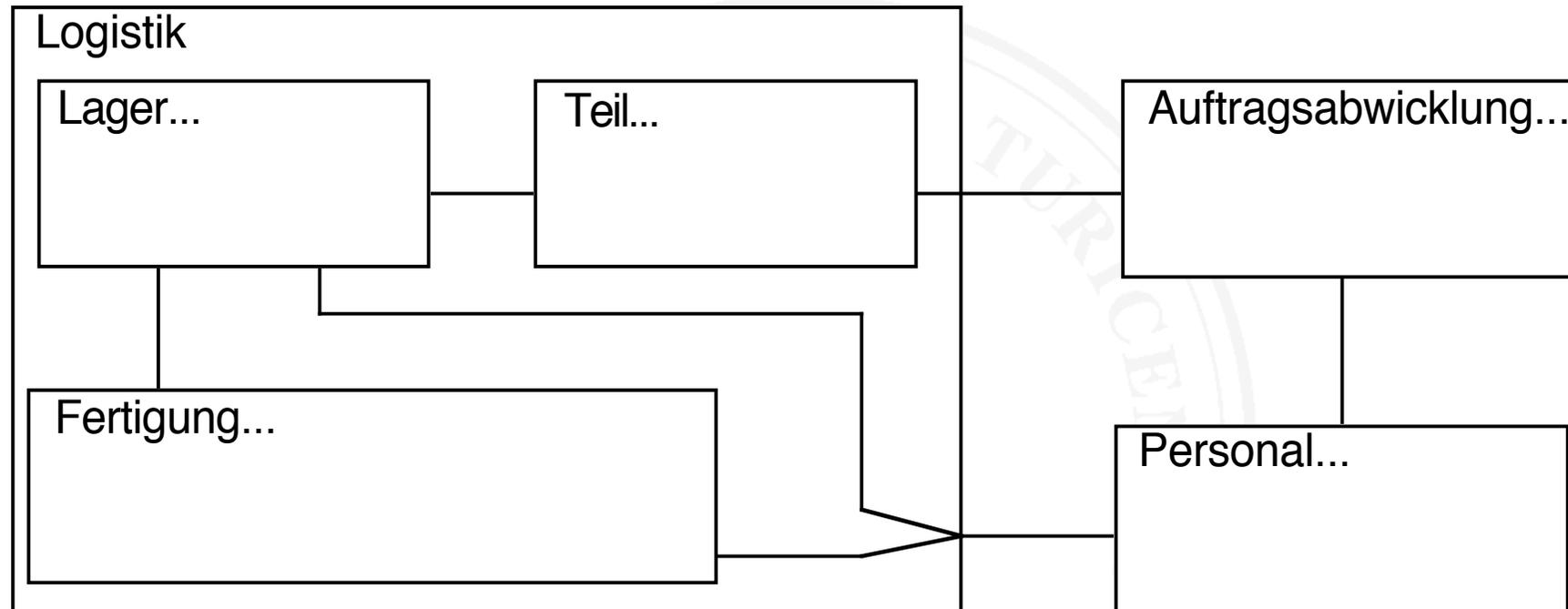
Komplexe Gebilde mit vielen Komponenten...

# Komposition – 2



# Komposition – 3

---



...abstrahieren und übersichtlicher machen

# Komposition – 4

---

- **Komposition (composition)** – Zusammenfassung einer Menge von Individuen mit teilweise gemeinsamen Merkmalen zu einem **neuen Individuum** mit neuen Merkmalen, welches die **Gesamtheit** der zusammengefassten Merkmale **repräsentiert**
- Die Komposition
  - fasst eine Menge von Einzelkomponenten (**Teilen**) unter Weglassung von Details zu einer übergeordneten Komponente (einem **Ganzen**) zusammen
  - fasst nicht irgendetwas irgendwie zusammen, sondern nur **logisch zusammengehörende** Komponenten
- Die übergeordnete Komponente ( das Ganze) ist ein in sich möglichst **geschlossenes** Teilmodell

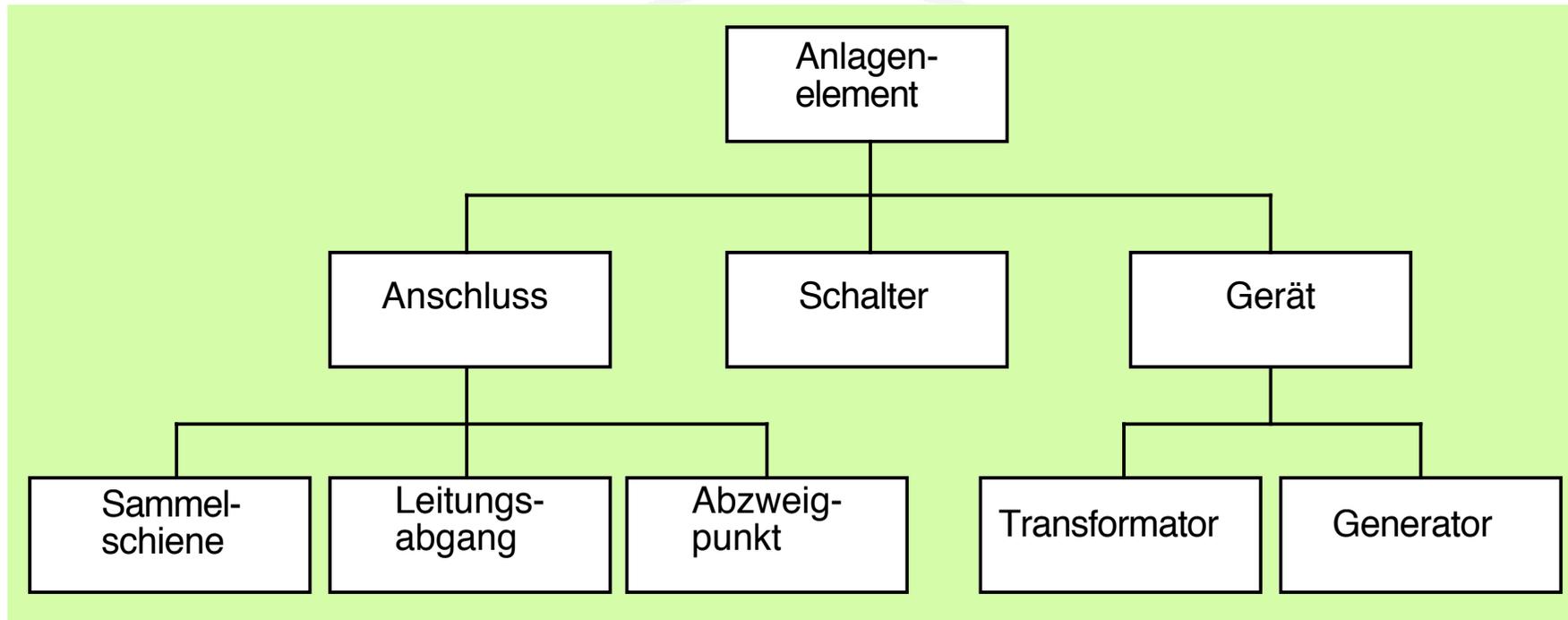
# Komposition – 5

---

- Zusammenfassung über mehrere Stufen  
⇒ **Kompositionshierarchie** (Teil-Ganzes-Hierarchie)
- Kompositionshierarchien bilden in umgekehrter Richtung eine **hierarchische Zerlegung** eines Modells
- **Zentrales Mittel zur Beherrschung der Komplexität** von Modellen mit vielen Einzelkomponenten

## 12.4 Generalisierung

---



Beispiel einer Generalisierungsabstraktion – Modellierung der Begriffswelt eines Problembereichs (Engineering von Unterstationen in Stromverteilungsnetzen)

# Generalisierung – 2

---

- **Generalisierung (generalization)** – Zusammenfassung einer Menge von Individuen mit teilweise gemeinsamen Attributen durch ein **übergeordnetes Individuum**, welches **nur die gemeinsamen Attribute** aufweist.
- Die Bildung von **Oberbegriffen** zu Begriffen im menschlichen Denken ist eine Generalisierungsabstraktion
- Die Generalisierung erlaubt die **systematische Modellierung ähnlicher Dinge** (solcher mit teilweise gemeinsamen und teilweise unterschiedlichen Merkmalen)
- Die Bildung von **Klassenhierarchien** in Klassen- und Objektmodellen basiert auf der Generalisierung
- Generalisierung ist **auch in Datenmodellen möglich** (in dieser Vorlesung nicht behandelt)

# 12.5 Benutzung

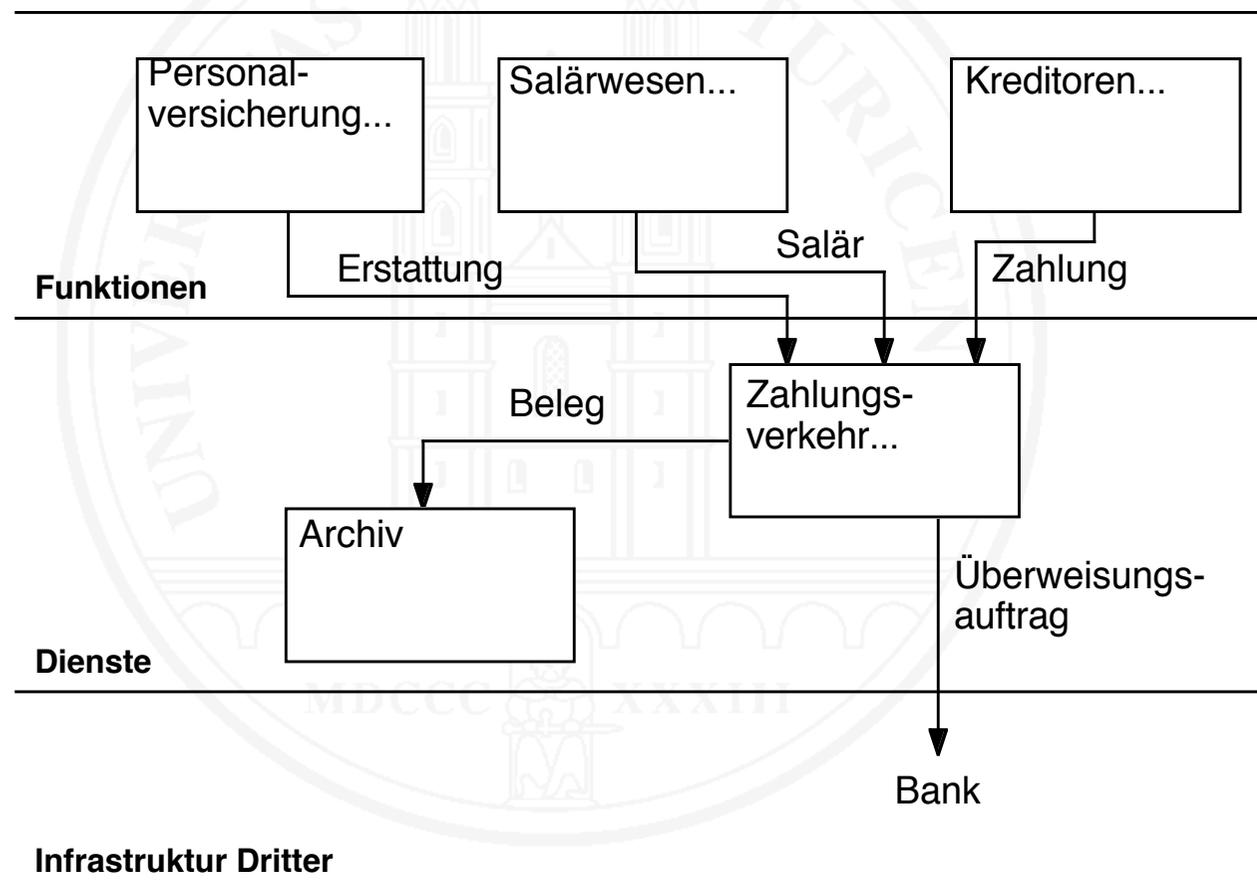
---



Komplexe Gebilde...

# Benutzung – 2

... durch **Delegieren** von Aufgaben und **Anordnung** der Aufgaben in **Schichten** übersichtlich und anschaulich modellieren



# Benutzung – 3

---

- **Benutzung** (delegation, usage) – Verwendung von Leistungen eines Individuums (oder einer Menge von Individuen) durch ein anderes Individuum zum Zweck der Erbringung eigener Leistungen.
- Bündelung von Leistungen zu höherwertigen, komplexen Leistungen; Verwender der komplexen Leistung sehen/kennen die dabei benutzten Leistungen nicht mehr
- Erlaubt die Bildung von Schichten, deren Komponenten
  - sich einerseits auf Leistungen tieferer Schichten abstützen,
  - andererseits Leistungen für höhere Schichten anbieten
  - Metapher der virtuellen Maschinen (vgl. Kap. 11)
- In der Informatik von Dijkstra (1968) erstmals systematisch verwendet

# Benutzung – 4

---

- **Benutzer** der Leistungen einer Schicht müssen nur **wissen**, **was** diese Leistungen sind, nicht aber, **wie** das Modell im Inneren der leistungserbringenden Schicht und in allen darunterliegenden Schichten aussieht
- **Geheimnisprinzip** (**information hiding**, Parnas 1972)
- Verstehen eines Modells, ohne alle Details zu kennen

# Aufgabe 12.1

Überlegen Sie, welche Abstraktionsart in den folgenden vier Situationen verwendet wird. Begründen Sie Ihre Aussage.

- a) Der Rechner besteht aus Zentraleinheit, Bildschirm, Tastatur, Maus.
- b) Eva, Christian, Fritz und Franziska sind Menschen.
- c) Hans fährt Auto. Von Technik versteht er nichts.
- d) Katzen, Hunde und Bären sind Raubtiere.

# Literatur

---

Dijkstra E.W. (1968). The Structure of the THE multiprogramming System. *Communications of the ACM* **11**, 5 (May 1968). 341-346.

Ghezzi, C., M. Jazayeri, D. Mandrioli (1991). *Fundamentals of Software Engineering*. Englewood Cliffs: Prentice-Hall.

Joos, S., S. Berner, M. Arnold, M. Glinz (1997). Hierarchische Zerlegung in objektorientierten Spezifikationsmodellen. *Softwaretechnik-Trends* **17**, 1 (Feb 1997). 29-37.

Parnas, D.L. (1972). On the Criteria To Be Used in Decomposing Systems into Modules. *Communications of the ACM* **15**, 12 (Dec. 1972). 1053-1058.