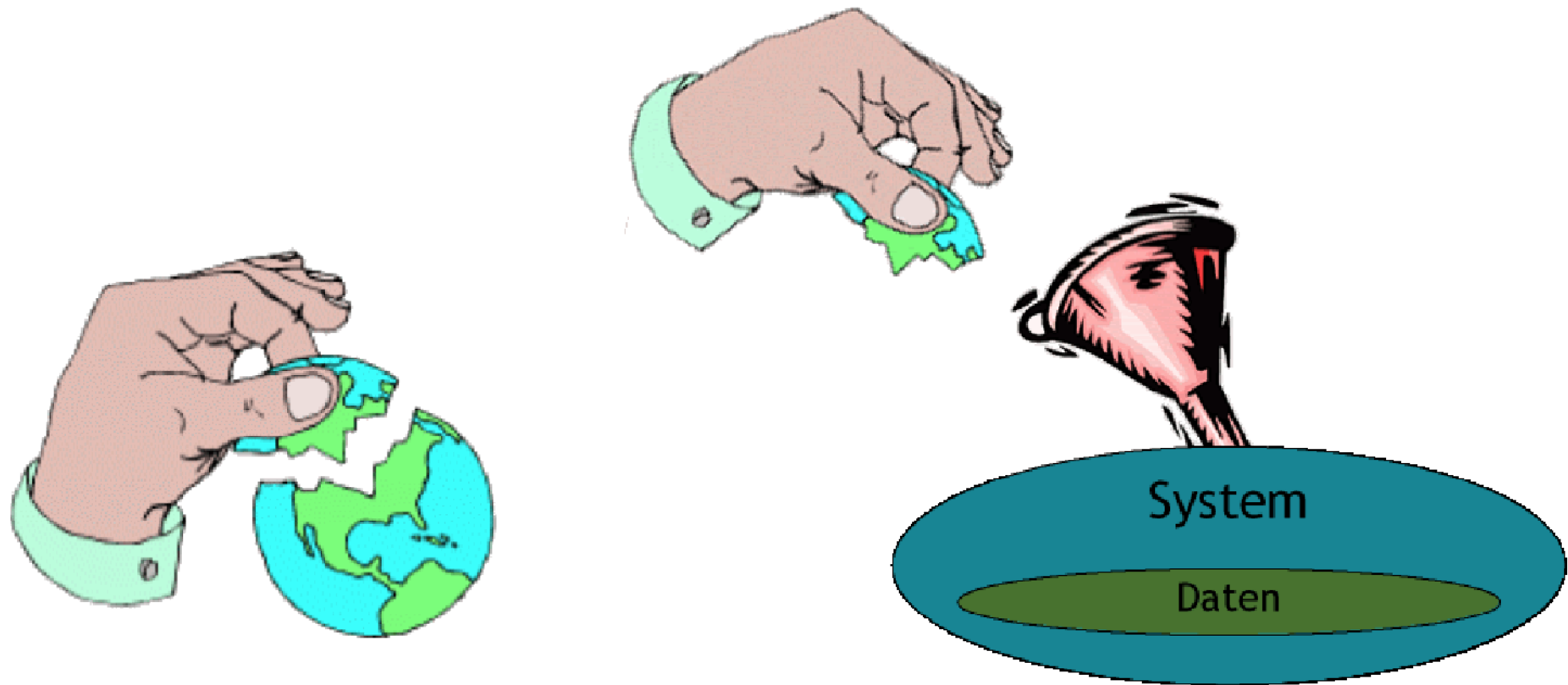


Lernmodul 2 **Datenmodellierung**



Datenmodellierung **Übersicht**

- Datenverwaltung und Datenbanken
- objektorientierte Abbildung der Realität
- Grundlagen der Objektorientierung
- Darstellung der Objektorientierung in UML



Motivation **Datenverwaltung und Datenbanken**



Motivation **Geoinformationssystem**

Wir erinnern uns...

Ein Geoinformationssystem (GIS) ist ein DV-gestütztes Informationssystem zur

- **Erfassung**
- **Verwaltung**
- **Analyse**
- **Verknüpfung**
- **Visualisierung**

von Geoinformationen.

Die zugrundeliegenden Geodaten beschreiben die Geometrie, Topologie, Thematik und Dynamik der Geoobjekte.



Motivation **Datenverwaltung**

Die Datenverwaltung umfasst:

- **Speichern von Daten,**
- **Aktualisierung gespeicherter Daten,**
- **Kontrolle des Zugriffs auf die Daten**
- **Ausgabe und Weitergabe von Daten**

Zur Datenverwaltung werden heute **Datenbanken** eingesetzt; sie bilden den Kern eines Geoinformationssystems.



Motivation **Datenbanken**

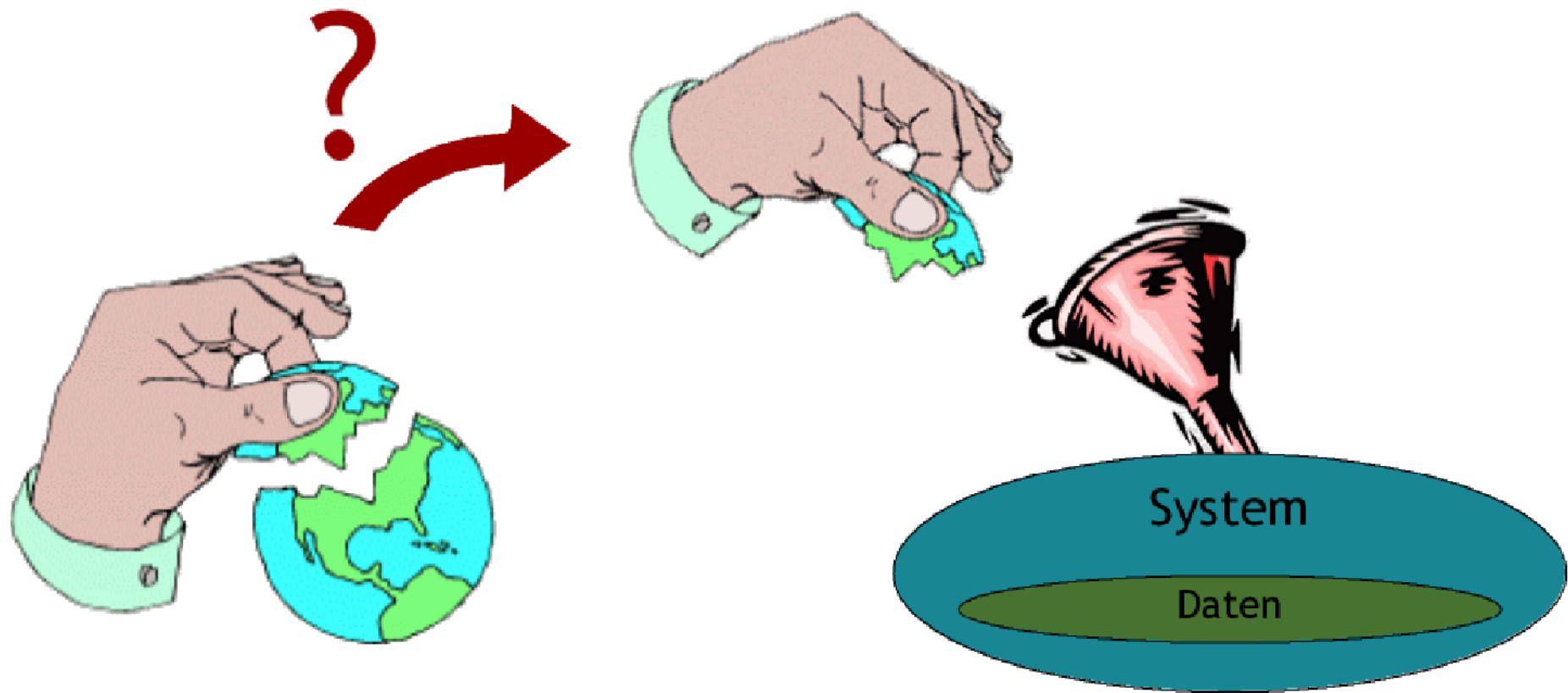
Datenbanken sind große Softwaresysteme zur

- **Beschreibung ("Modellierung")**
- **dauerhaften ("persistenten") und zuverlässigen Speicherung**
- **effizienten Wiedergewinnung**

(sehr) umfangreicher Datenmengen, die von mehreren Anwendungen gleichzeitig genutzt werden können.



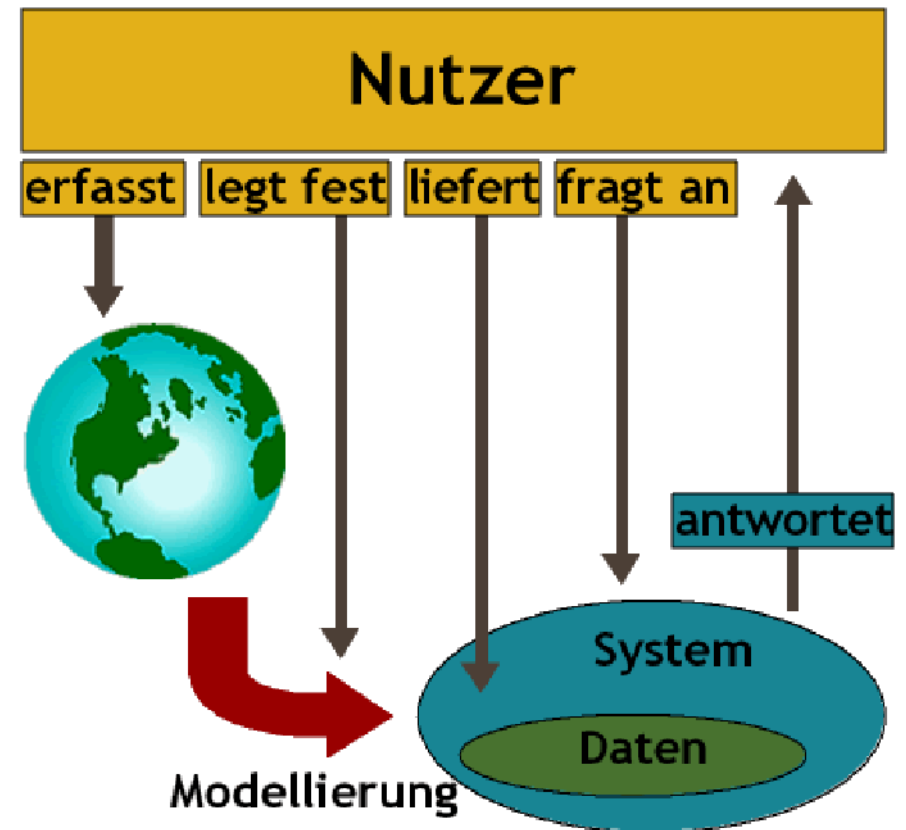
Datenmodellierung **Realität abbilden**



Datenmodellierung **Allgemein**

Datenmodellierung bezeichnet die Abbildung der Realität auf Strukturen und Prozesse eines Informationssystems.

- Das Modell legt fest, was das Informationssystem "weiß".
- Ein Modell begründet einen "Vertrag" zw. Nutzer und Informationssystem:
 - Welche Daten muss der Nutzer liefern?
 - Welche Anfragen kann der Nutzer stellen?
 - Welche Antworten gibt das System?



Datenmodellierung **Objektorientierung (OO)**

- Elemente (Lebewesen, Gegenstände, Konzepte, Prozesse, ...) der Realwelt werden als **Objekte** angesehen
- Grundlegendes Paradigma der Informatik bzgl. Strukturierung von Programmen und Philosophie des Programmierens:

Realität - Modell - Programm

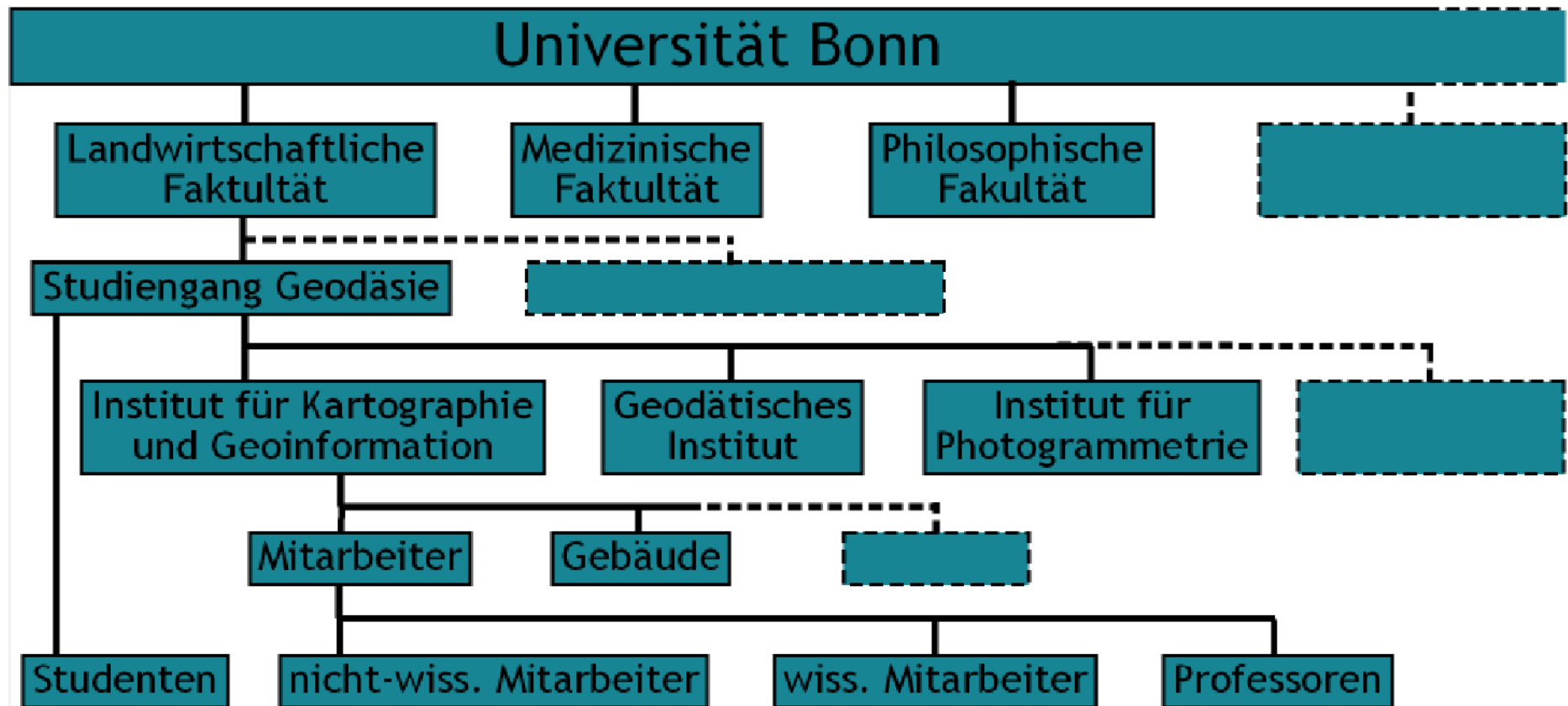
- Die Darstellung des Modells ist normiert:

"Unified Modelling Language" (UML)

Realweltobjekte



Datenmodellierung **Beispiel zur Objektorientierung**



Datenmodellierung **Bilden von Objekten**

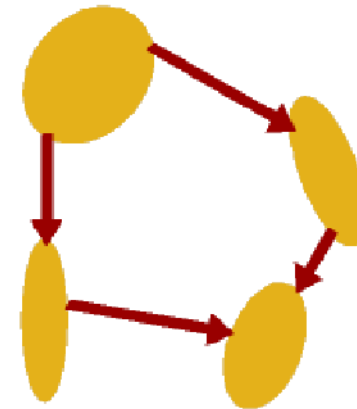
Realweltobjekte



Abstraktion



Modellierte Objekte



- direkte Modellierung von Objekten der Realität
- **Abstraktion:** Konzentration auf das für das Modell Wesentliche; nicht-relevante Details werden weggelassen.



Datenmodellierung **Beispiel zur Objektbildung**

Modellierung des "Objekts Student" aus einem "Realweltobjekt Student"



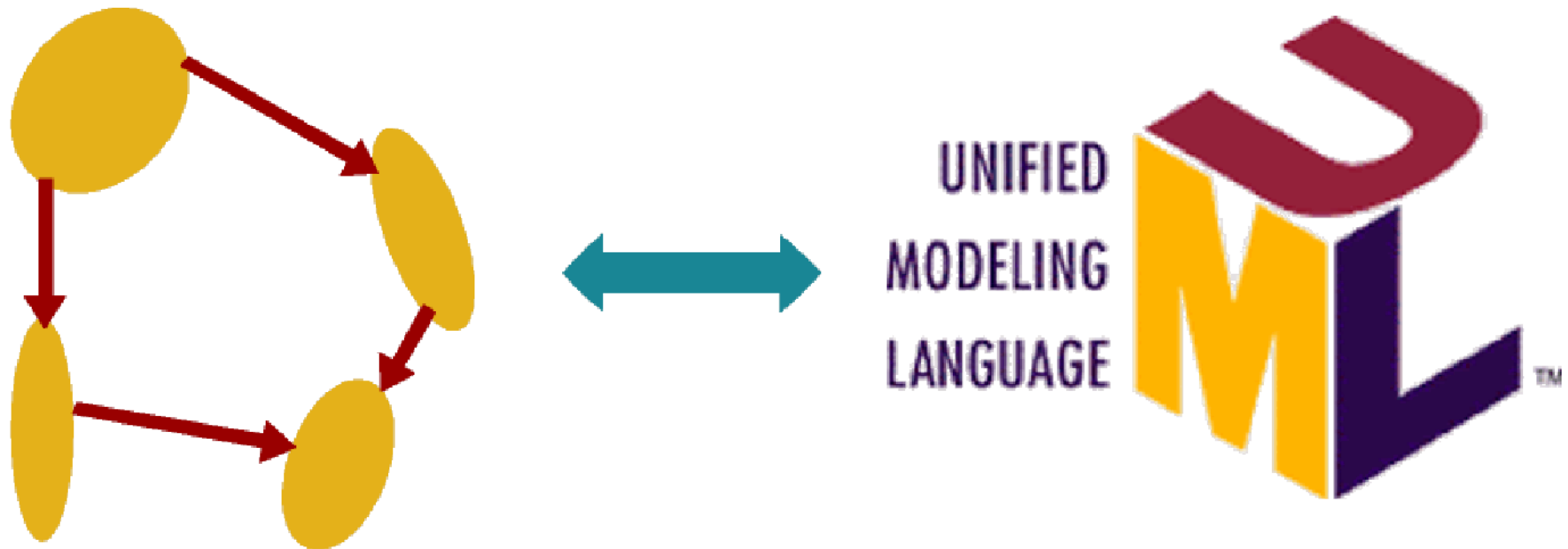
Name, Vorname, Geb.Datum, **Gewicht**,
Größe, **Schuhgröße**, **Haarfarbe**,
Matr.Nr., Adresse, Fach,
Einschreibedatum, Vordiplom, Hauptdiplom



Name, Vorname, Geb.Datum, Matr.Nr.,
Adresse, Fach, Einschreibedatum,
Vordiplom, Hauptdiplom



Objektorientierung Grundlagen in UML



Objektorientierung **Objekte, Instanzen**

- Ein modelliertes Objekt besitzt bestimmte **Eigenschaften**, und reagiert mit einem definierten **Verhalten** auf seine Umgebung
- Statt von Objekten wird häufig von Instanzen einer Klasse gesprochen.

Ab jetzt Darstellung in UML



Objektorientierung Beispiele für Objekte, Instanzen

Die Studenten Michael Müller und Michaela Meier

Michael Müller

Michaela Meier



Objektorientierung **Klassen**

- Zusammenfassung gleichartiger aber unterscheidbarer Objekte zu einer Menge
- "Bauplan" für einzelne Objekte

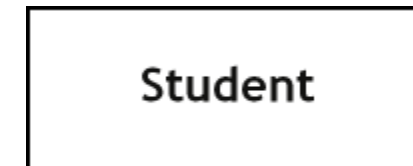


Klassen und Objekte stehen in Beziehung ("*instance of*")

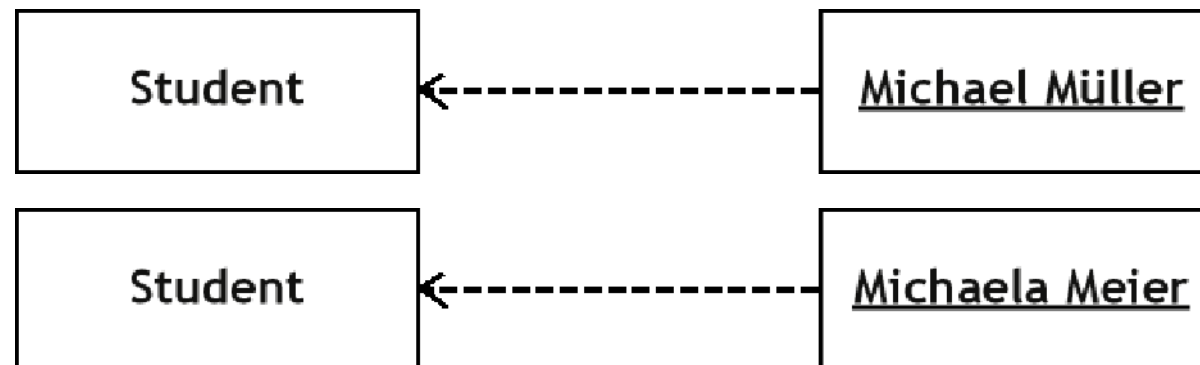


Objektorientierung Beispiele zu Klassen

Zusammenfassung einzelner Studenten zur Klasse
"Student"



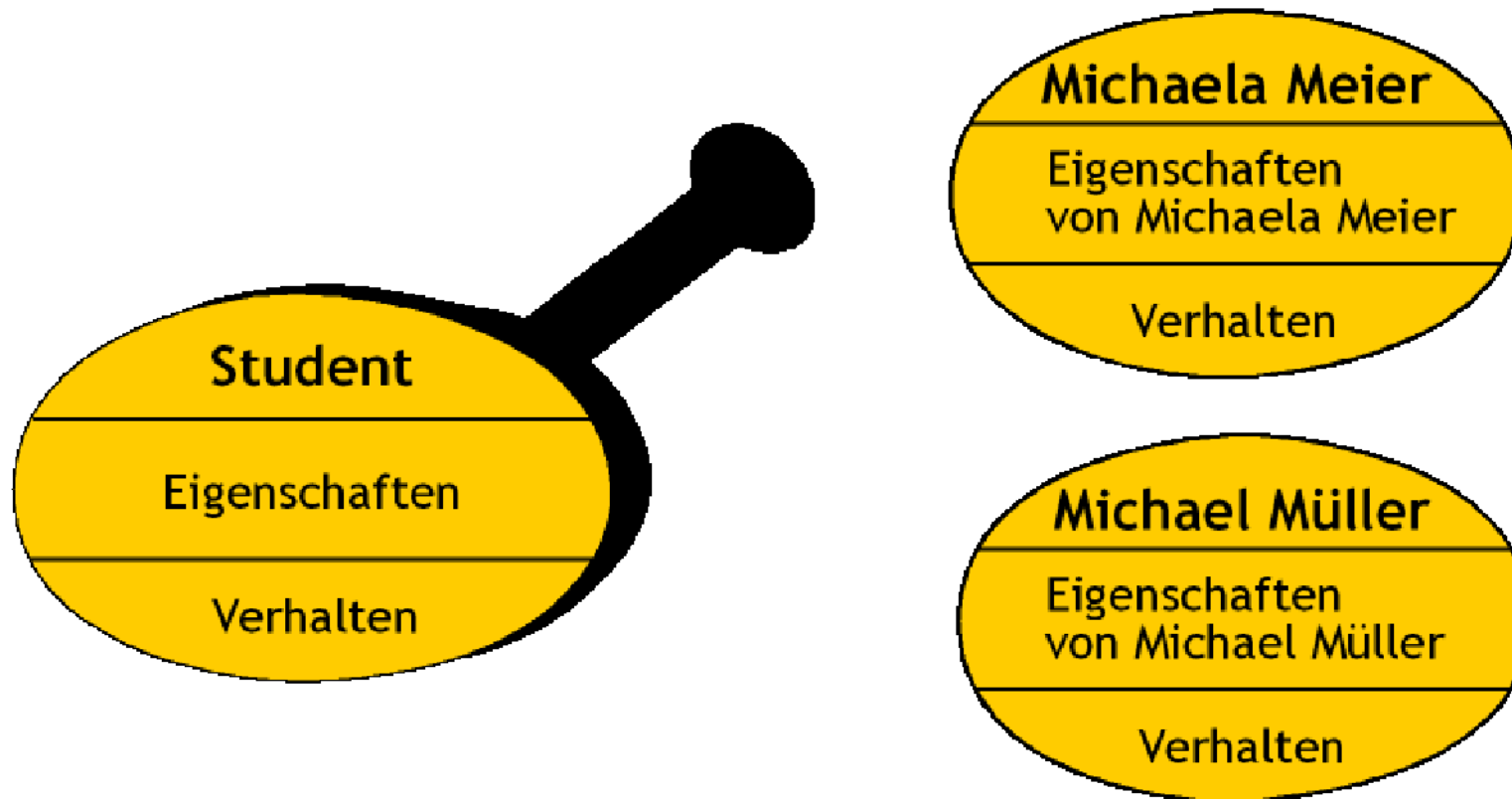
Michael Müller und Michaela Meier sind Instanzen der Klasse Student



Objektorientierung **Stempelmetapher**



Objektorientierung **Klassen und Objekte I**



Objektorientierung **Zur Bezeichnung von Objekten**

1. Jedes Objekt hat einen **identifizier** (Namen), der ihn eindeutig identifiziert.
Vorsicht: Der **identifizier** eines Studenten ist nicht der Name, sondern die Matrikelnummer.

2. Jedes Objekt gehört zu einer Klasse, diese Zuordnung ist stets bekannt.

Es gibt zwei Möglichkeiten der Bezeichnung:

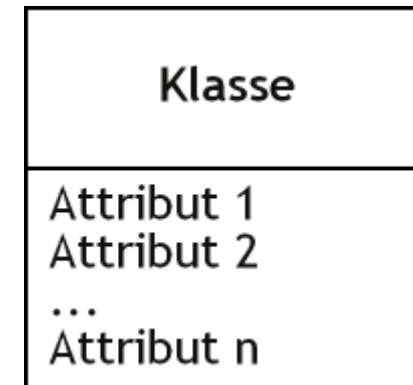
Objekt

Objekt: Klasse



Objektorientierung **Attribute**

- **Eigenschaften**, die zur Beschreibung von Objekten einer Klasse wesentlich sind und von deren Objekten an-genommen werden können – Daten einer Klasse
- Die Kombination der Eigenschaften beschreibt den **Zustand** eines Objektes
- alle Objekte einer Klasse besitzen dieselben Objekt-Attribute, aber unterschiedliche Attributwerte



Objektorientierung **Beispiel zu Attributen**

Attribute zur Beschreibung eines Studenten

Student
Matrikelnummer
Name
Vorname
Fach
Adresse
Geburtsdatum
Einschreibedatum
Vordiplom
Hauptdiplom



Objektorientierung **Methoden (Operationen)**

- Beschreibung des wesentlichen **Verhaltens** von Objekten einer Klasse
- **Verhalten** bezeichnet entweder eine Änderung des Zustands des Objekts oder eine Interaktion des Objekts mit der Umwelt

Klasse
Attribut 1 Attribut 2 ... Attribut n
Methode 1() Methode 2() ... Methode n()



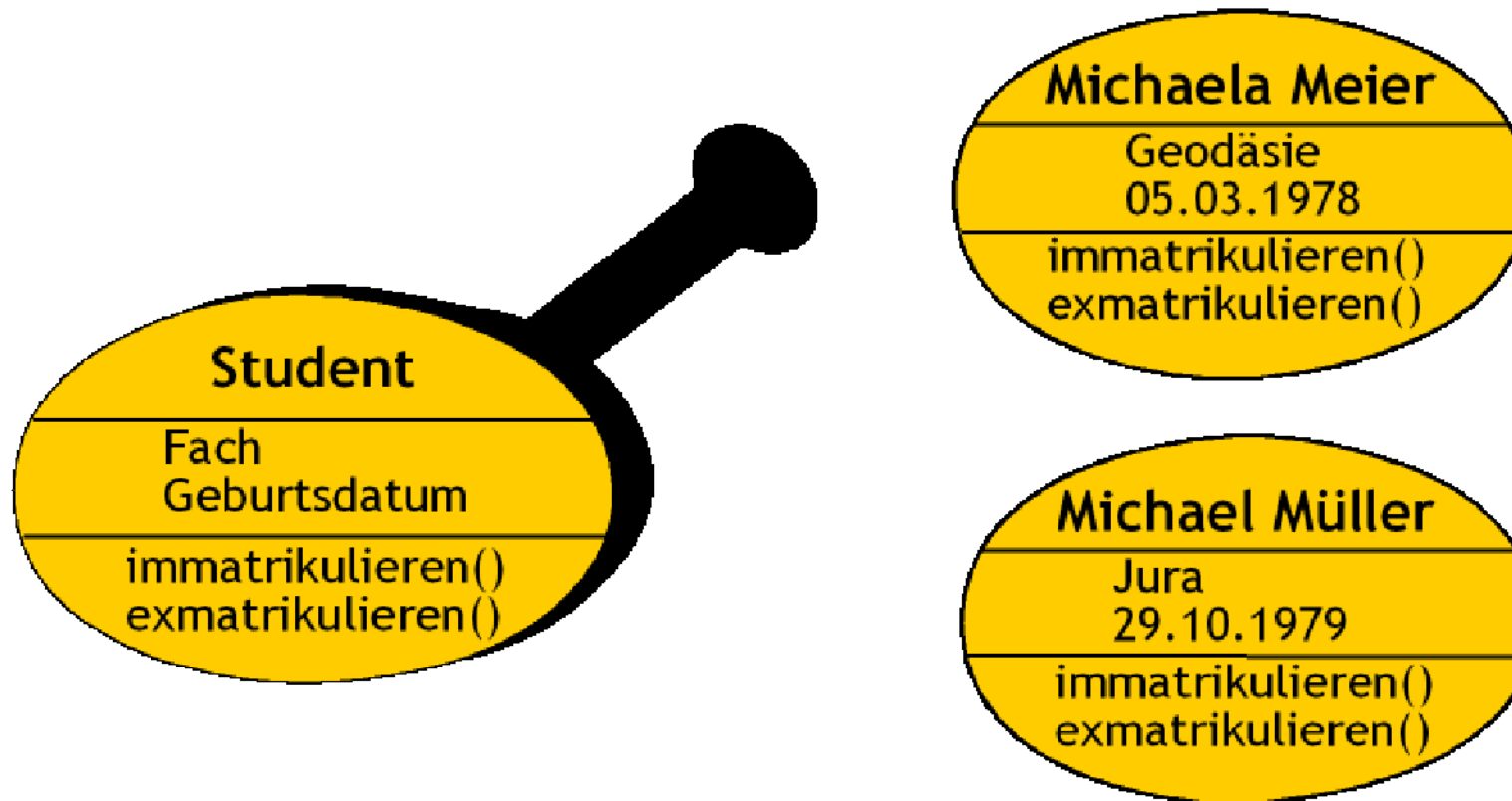
Objektorientierung **Beispiel zu Methoden**

Methoden der Klasse Student

Student
Matrikelnummer Name Vorname Fach Adresse Geburtsdatum Einschreibedatum Vordiplom Hauptdiplom
immatrikulieren() exmatrikulieren() druckeBescheinigung() semesterzahl() setVordiplom() setHauptdiplom()

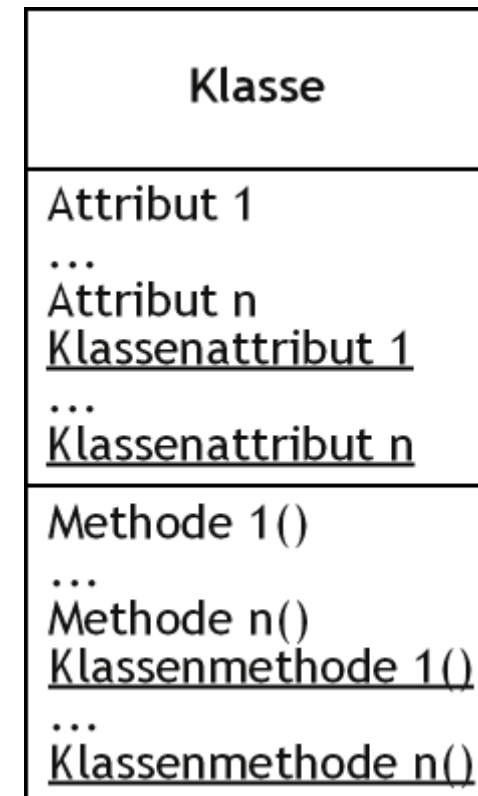


Objektorientierung **Klassen und Objekte II**



Objektorientierung **Klassenattribute und -methoden**

- Bei Klassenattributen existiert nur ein Attributwert für alle Objekte einer Klasse
- Klassenmethoden sind einer Klasse zugeordnet und können nicht auf einzelne Objekte angewandt werden



Objektorientierung **Bsp. Klassenattribute /-methoden**

Klassenattribute und -methoden der Klasse Student

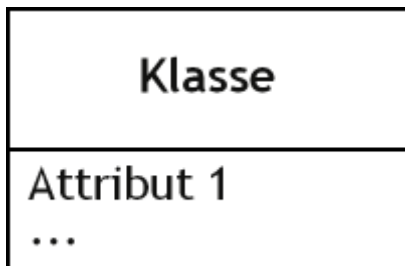
Student
Matrikelnummer Name Vorname ... Vordiplom Hauptdiplom <u>Anzahl</u>
immatrikulieren() exmatrikulieren() ... setVordiplom() setHauptdiplom() <u>durchschnittsalter()</u>



OO Freiheitsgrade bei der Notation von Klassen



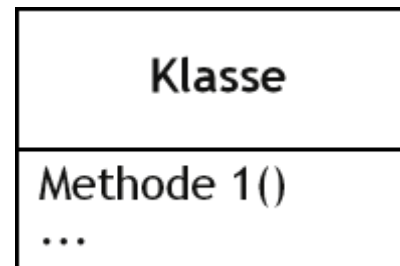
Klasse **ohne** Attribute und Operationen



Fehlende/unwichtige Methoden



Attribute und Methoden **nicht dargestellt**



Fehlende/unwichtige Attribute



Objektorientierung **Attributspezifikation**

Klasse
Attribut1: Typ 1 Attribut2: Typ 2 = Initialwert ...
Methode1(parameter) Methode2(parameter: parametertyp= Initialwert) Methode3(parameter: parametertyp= Initialwert) : Ergebnistyp ...

Typen, Initialwerte und Parameter sind optional!

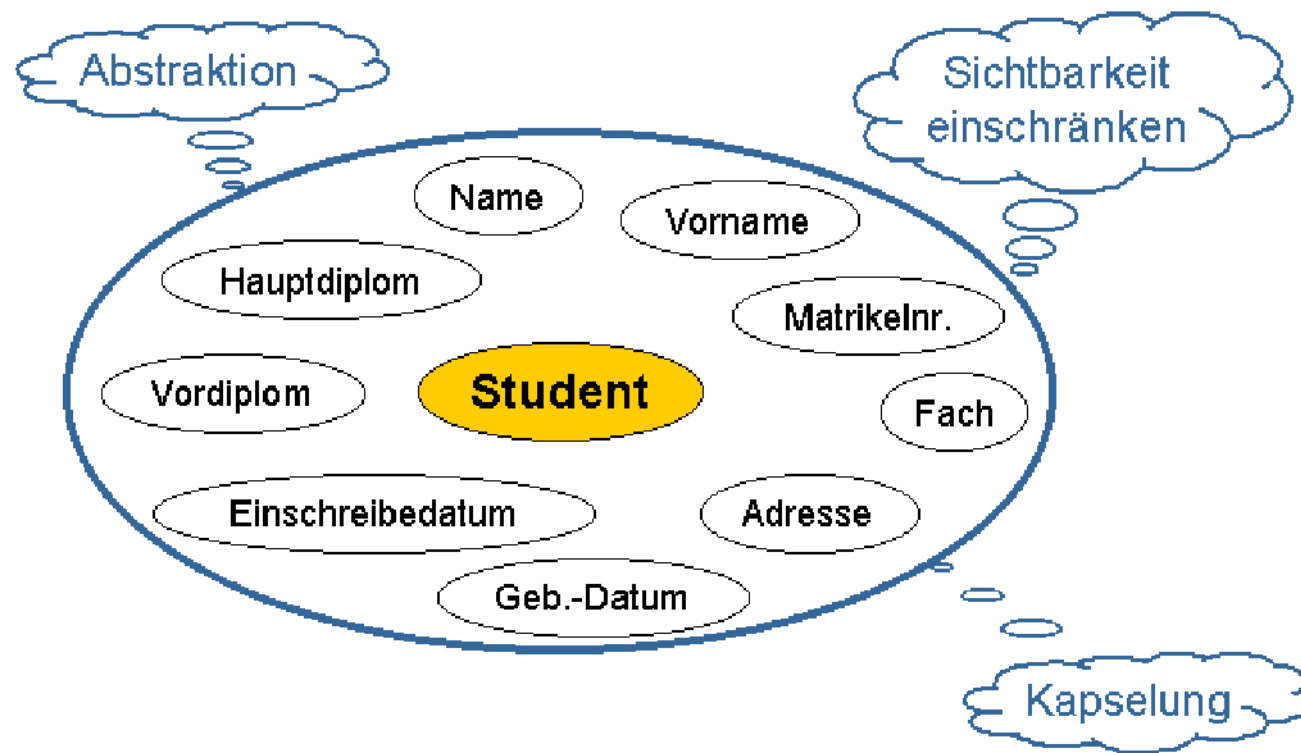


Objektorientierung **Beispiel zur Attributspezifikation**

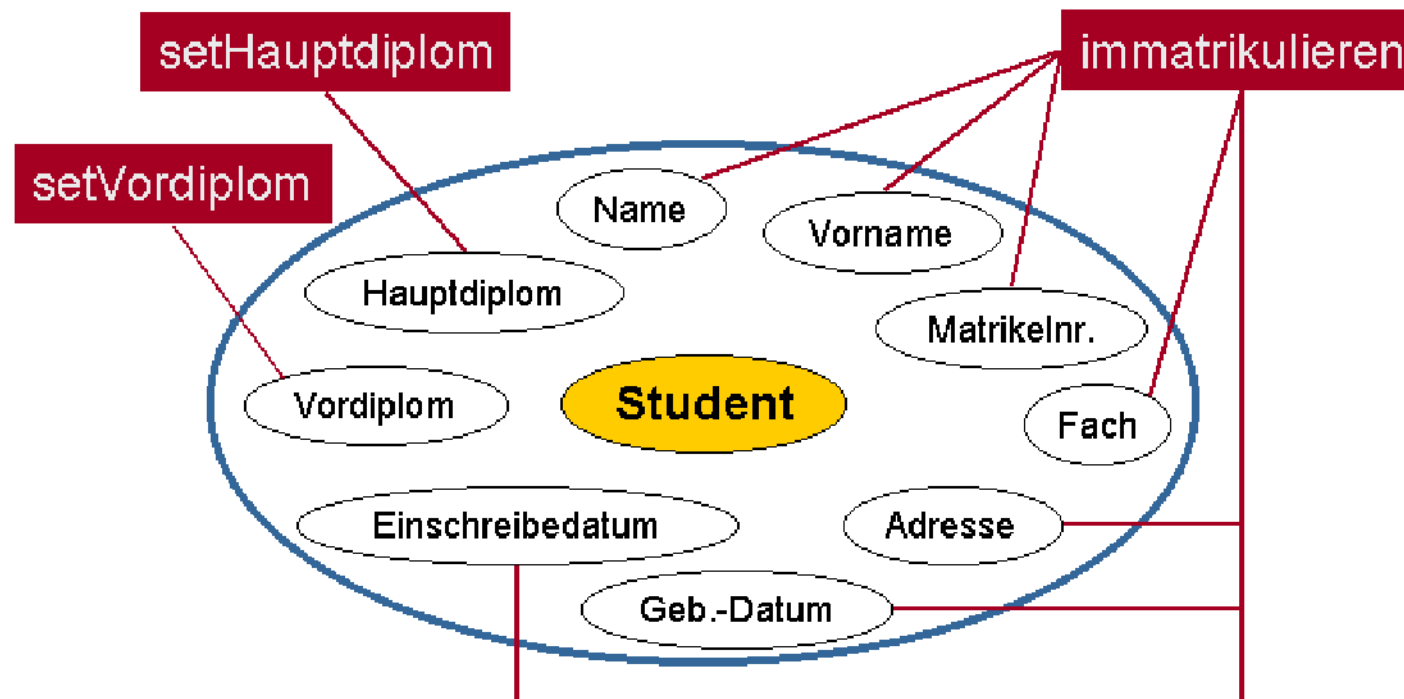
Student
Matrikelnummer: String Name: String Vorname: String ... Vordiplom: Boolean = false Hauptdiplom: Boolean = false <u>Anzahl: Int = 0</u>
immatrikulieren() exmatrikulieren() ... setVordiplom(bestanden) : Boolean setHauptdiplom(bestanden: Boolean) : Boolean <u>durchschnittsalter()</u>



Objektorientierung **Kapselung I**



Objektorientierung **Kapselung II**



OO Idee der Kapselung/eingeschränkten Sichtbarkeit

- Zusammengehörige Attribute und Methoden sind in einer Einheit – der Klasse – **verkapselt**.
- ein Objekt tritt mit seiner Umwelt nur über Methoden in Kontakt
 - Attribute sind
 - nach außen **nicht sichtbar**
 - vor Änderung und Lesen geschützt
 - öffentlich zugängliche Methoden
 - ermöglichen den Zugriff auf Attribute
 - schützen die Daten
 - sind zuständig für Konsistenzwahrung
- Andere Bezeichnungen für "eingeschränkte Sichtbarkeit": "Geheimnisprinzip", "information hiding"



Objektorientierung **Sichtbarkeit**

- Sichtbar innerhalb der Klasse
 - **private**
 - Notation: **-**
- Sichtbar außerhalb der Klasse (in anderen Klassen)
 - **public**
 - Notation: **+**

Klasse
-Attribut 1 ... -Attribut n <u>-Klassenattribut 1</u> ... <u>-Klassenattribut n</u>
+Methode 1() ... +Methode n() <u>+Klassenmethode 1()</u> ... <u>+Klassenmethode n()</u>



Objektorientierung **Beispiel zur Sichtbarkeit**

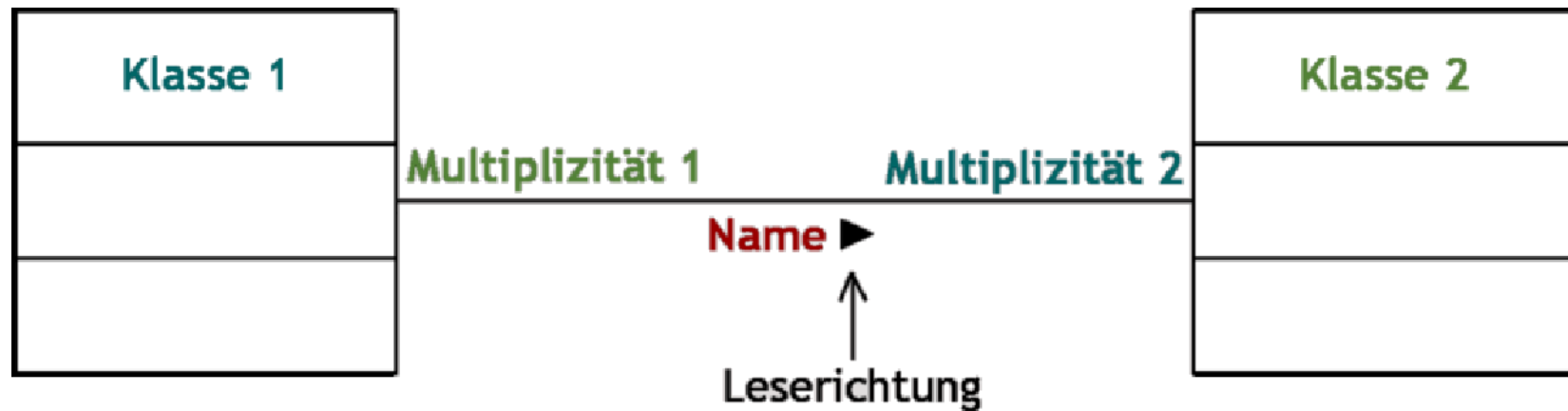
Klasse Student

Student
-Matrikelnummer -Name -Vorname ... -Vordiplom -Hauptdiplom <u>-Anzahl</u>
+immatrikulieren() +exmatrikulieren() ... +setVordiplom() +setHauptdiplom() <u>+durchschnittsalter()</u>



Objektorientierung **Assoziation I**

Assoziation modelliert Beziehungen zwischen Objekten ein oder mehrerer Klassen.



Objektorientierung **Beispiel 1 zur Assoziation**

Beziehung von Partneruniversitäten zur Universität Bonn



Objektorientierung **Mögliche Multiplizitäten**

1	genau eins (Universität und Rektor)
0..1	null oder eins (Lebenspartner)
0..*	größer oder gleich null - Standardfall (erfolgreiche Prüfungen eines Studenten)
0..3	null, eins, zwei oder drei (Wiederholungen einer Prüfung)
3,7	drei oder sieben
1..*	größer oder gleich eins (Studienfächer)
0..3,7,9..*	null, eins, zwei, drei, sieben und größer oder gleich neun



Objektorientierung **Assoziation II**



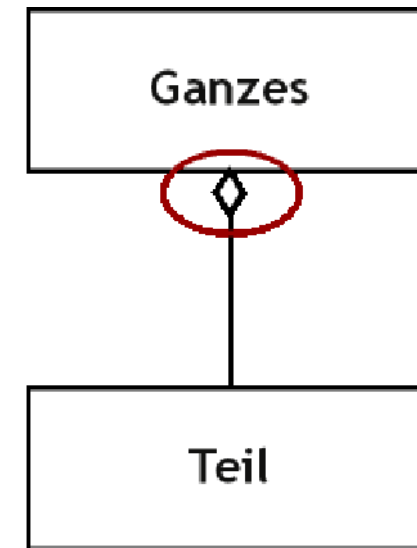
Objektorientierung **Beispiel 2 zur Assoziation**

Beziehung von Professor und Vorlesung



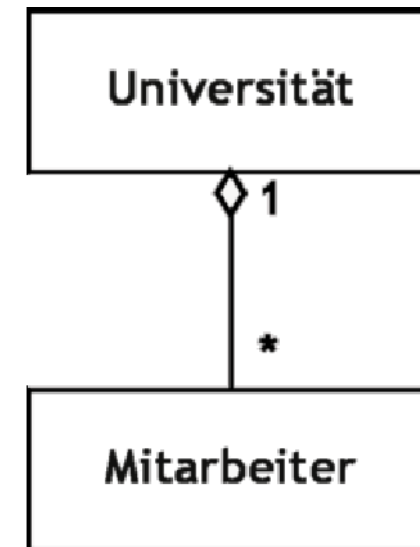
Objektorientierung **Aggregation**

- spezielle Assoziation, deren beteiligte Klassen zueinander in Beziehung stehen, wie ein Ganzes zu seinen Teilen (Ganzes-Teile-Hierarchie)
- Hierarchie lässt sich durch „ist Teil von“ bzw. „besteht aus“ beschreiben



Objektorientierung Beispiel zur Aggregation

Mitarbeiter der Universität



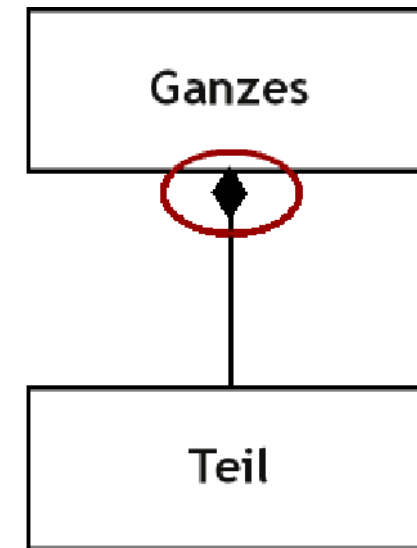
OO Aggregation als spezielle Assoziation

- Eine Assoziation erfüllt die Anforderungen an eine Aggregation, wenn mindestens eine der folgenden Bedingungen erfüllt ist:
 - die umgangssprachliche Verwendung der Bezeichnungen „**Bestandteil**“, „Teil“ - „Ganzes“ ist angemessen
 - Operationen werden sinnvollerweise automatisch auf alle Teile eines Ganzen angewendet (**anzeigen**)
 - es liegt eine systematische **Asymetrie** vor, die eine Klasse der anderen unterordnet



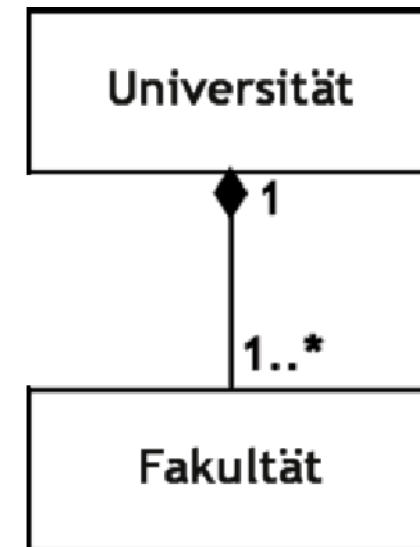
Objektorientierung **Komposition**

- strenge Form der Aggregation, bei der die **Existenz** des Teils von der Existenz des Ganzen **abhängt**
- Wird das Ganze gelöscht, werden auch seine Teile gelöscht
- Wenn ein Teil zu **mehreren** Ganzen gehören kann, liegt **keine** Komposition vor
- Die Multiplizität der "Ganzes"-Klasse ist 1, die der Teilklass 1..n
- Anwendung: Löschung des Ganzen soll die automatische **Löschung** aller Teile hervorrufen



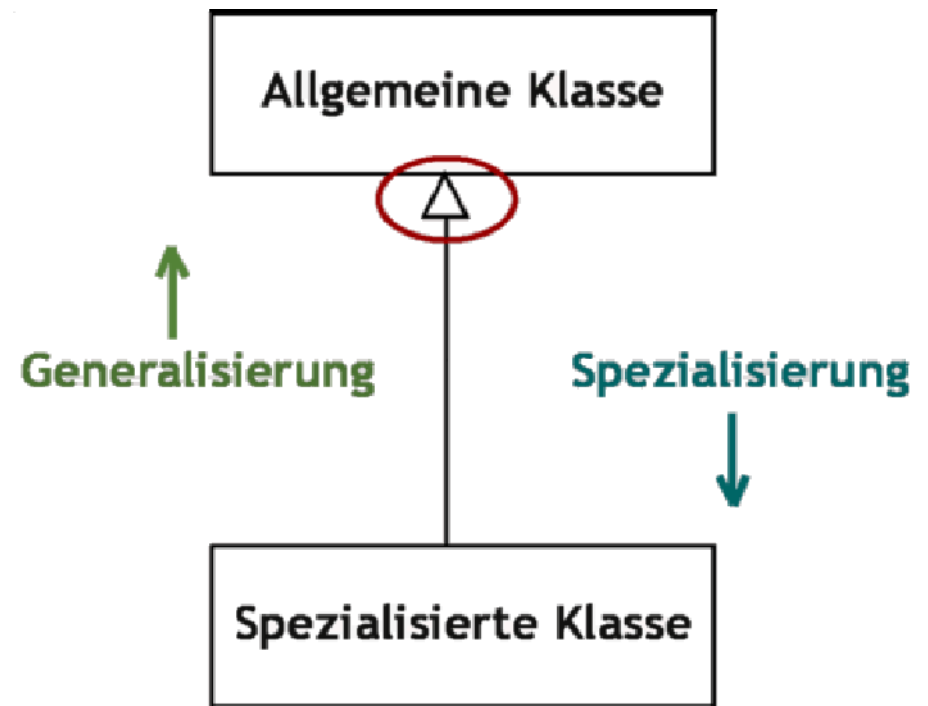
Objektorientierung **Beispiel zur Komposition**

Fakultäten der Universität

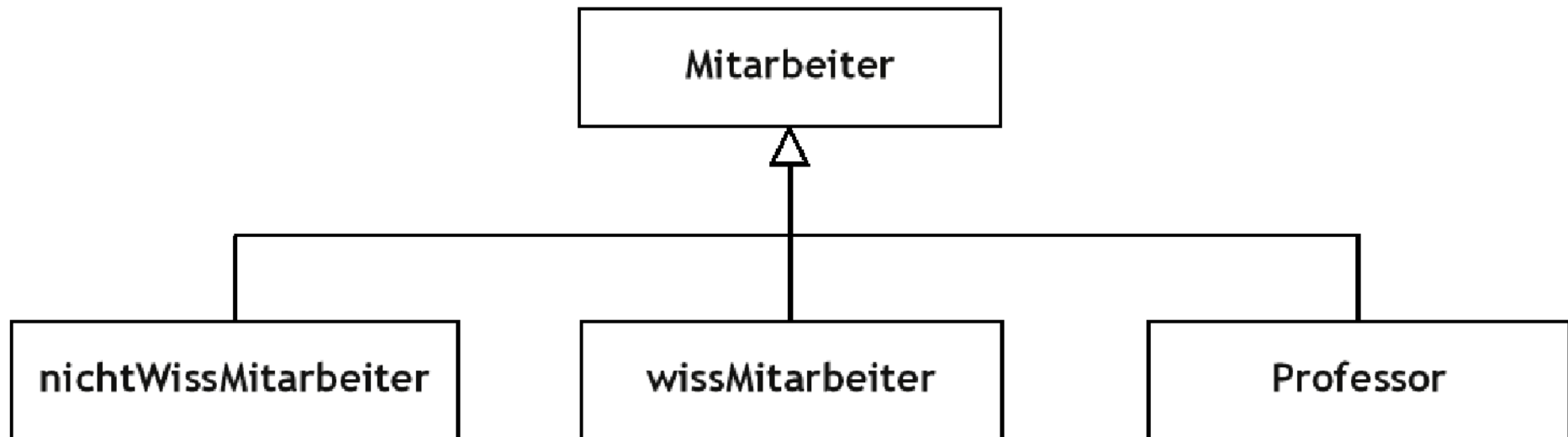


Objektorientierung **Generalisierung, Spezialisierung**

- Abstraktionsprinzipien zur hierarchischen Strukturierung eines Modells
- Eine **spezialisierte Klasse** konkretisiert eine **allgemeinere** (generalisiertere) **Klasse** durch das Hinzufügen von Eigenschaften
- Die allgemeine Klasse ist **Oberklasse**, die spezialisierte Klasse **Unterklasse**



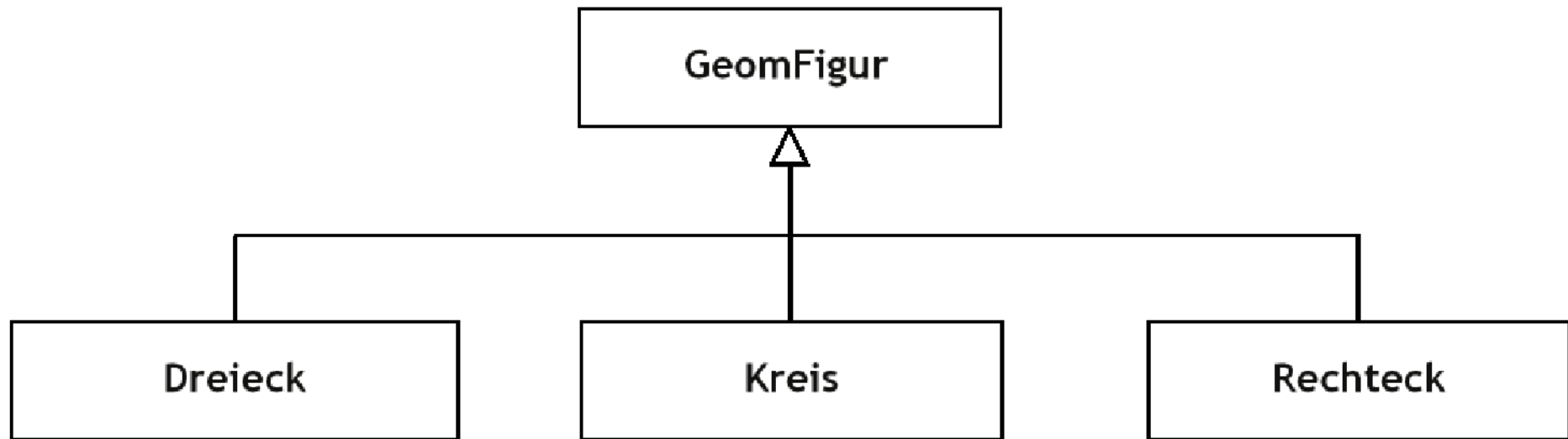
00 Beispiel 1 zu Generalisierung, Spezialisierung



- "Mitarbeiter" ist ein allgemeinerer Begriff als "nicht-wiss. Mitarbeiter", "wiss. Mitarbeiter" oder "Professor"
- "Mitarbeiter" ist **Oberklasse**, "nicht-wiss. Mitarbeiter", "wiss. Mitarbeiter" und "Professor" sind **Unterklassen**



OO Beispiel 2 zu Generalisierung, Spezialisierung



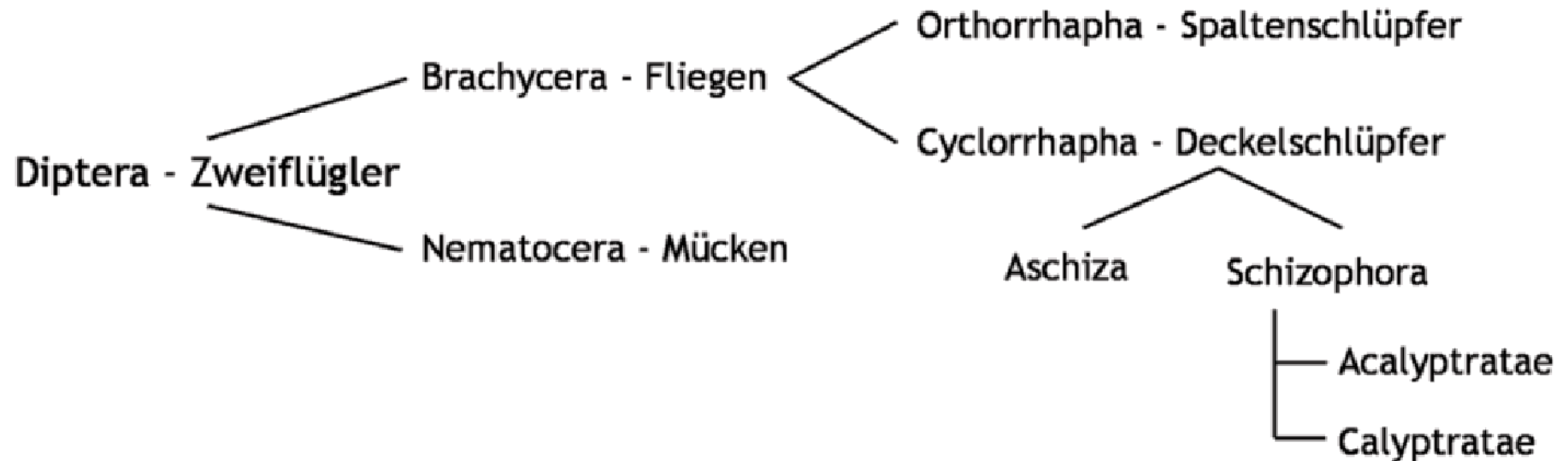
- Die "GeomFigur" ist ein allgemeineres Konzept als "Dreieck", "Kreis" oder "Rechteck"



Objektorientierung **Taxonomie**

Generalisierung und Spezialisierung beschreiben eine **taxonomische** (systematische) Beziehungen zwischen einem allgemeinen und einem speziellen Begriff

Beispiel: Taxonomie der Zweiflügler



Objektorientierung **Aggregation und Generalisierung**

Wie lässt sich der Unterschied zwischen einer **Aggregation** und einer **Generalisierung** feststellen?

Aggregation und **Generalisierung** bilden Hierarchien, aber:

- **Aggregation** bezieht **Objekte** aufeinander
- **Generalisierung** bezieht **Klassen** aufeinander
- beachte den systematischen Unterschied zwischen folgenden Beziehungen:
 - Universität - Mitarbeiter (**Aggregation** zwischen Objekten)
 - Rechteck - Figur (**Generalisierung** desselben Objekts)



Objektorientierung **Vererbung und Spezialisierung I**

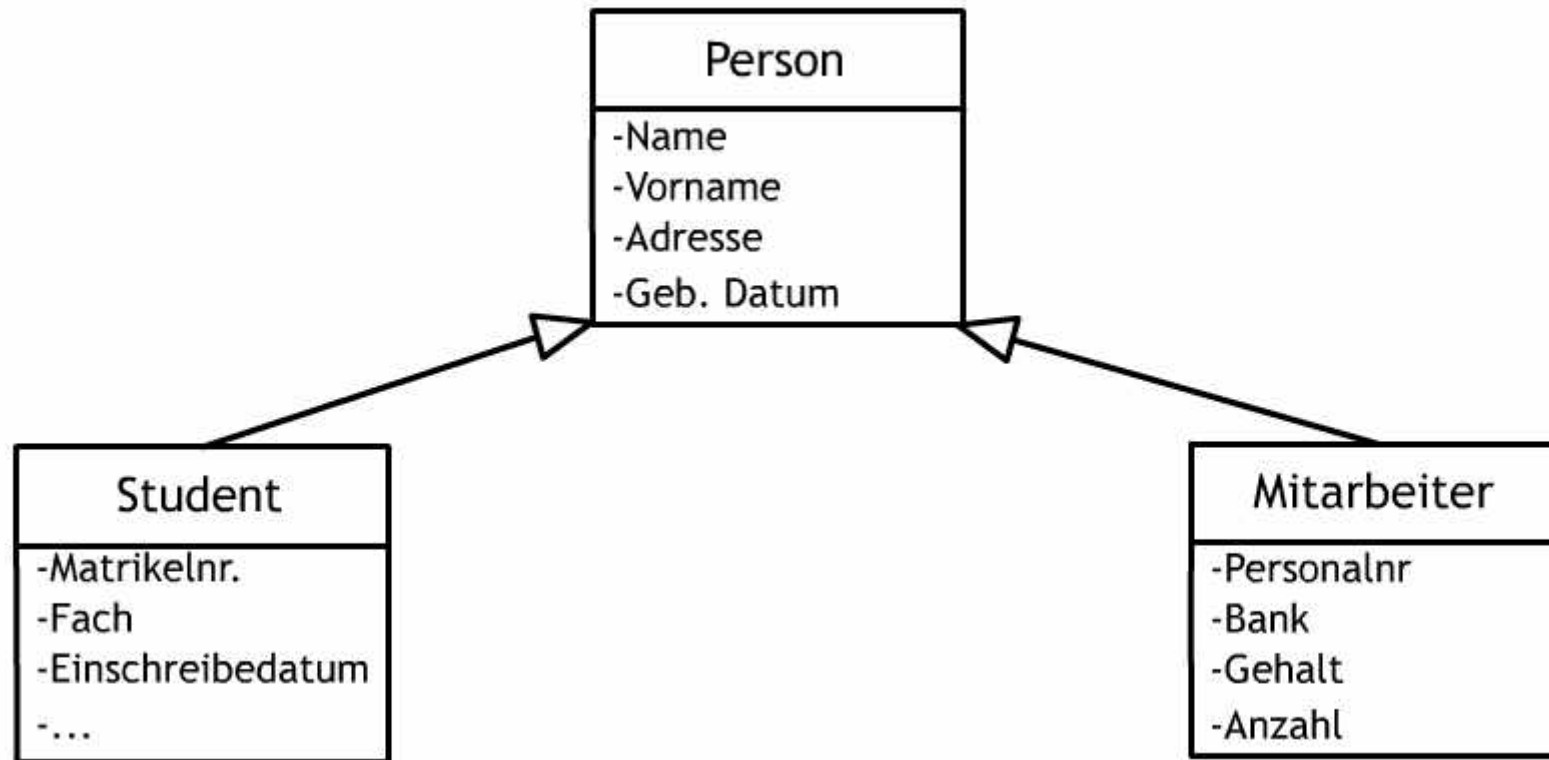
Eine Oberklasse repräsentiert **ein** Konzept.

Eine Unterklasse spezialisiert dieses Konzept, indem sie

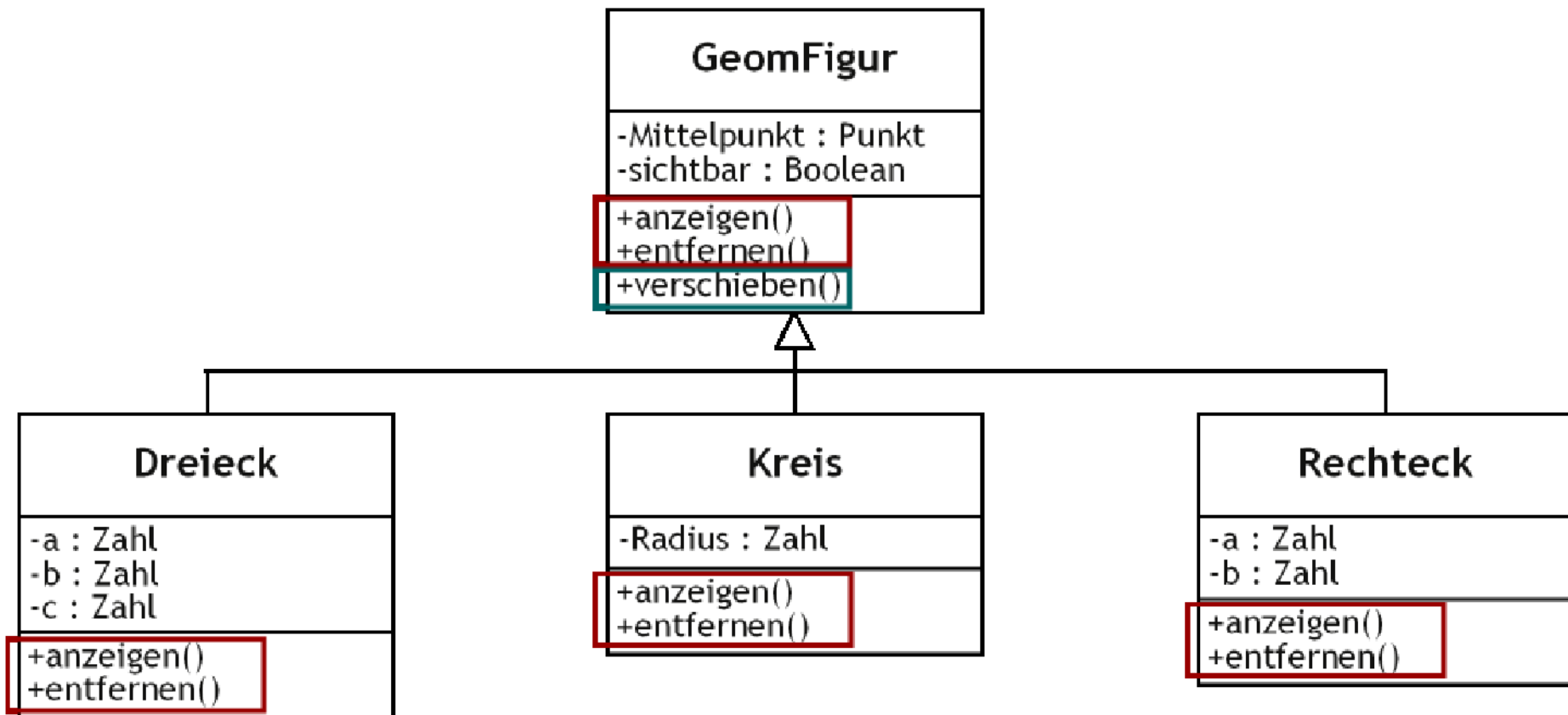
- **Attribute** der Oberklasse explizit **übernimmt**: **Erben** von Eigenschaften
- **Methoden** der Oberklasse explizit **übernimmt** oder **überschreibt**: **Erben** des Verhaltens
- neue **Attribute** definiert
- neue **Methoden** definiert



Objektorientierung **Beispiel 1 zur Vererbung**

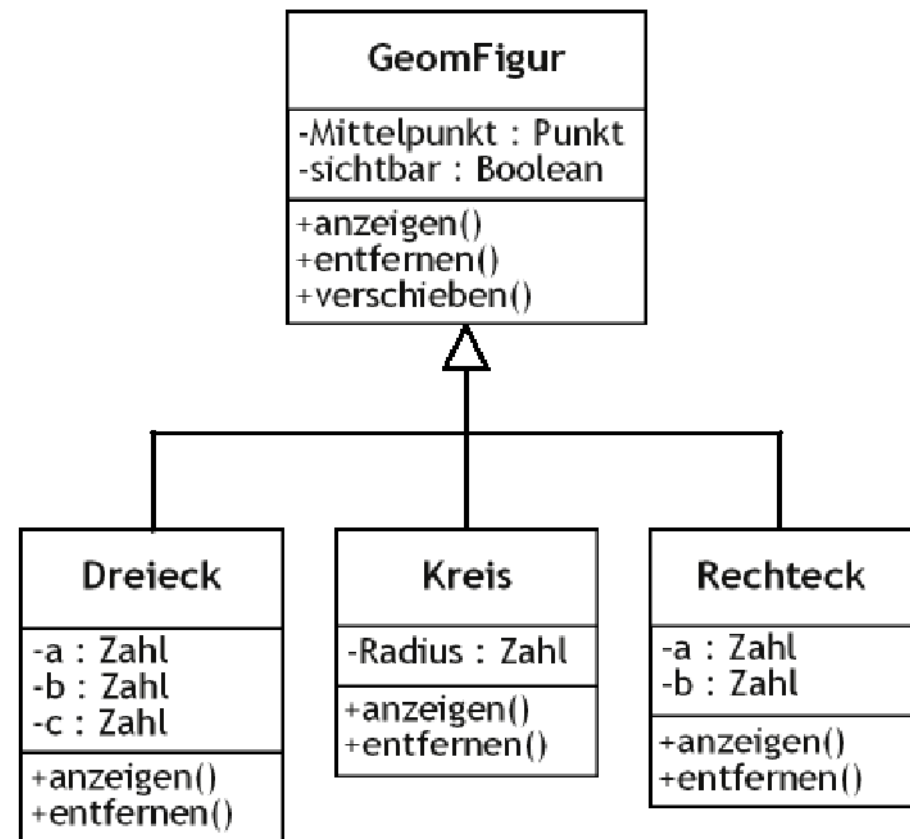


Objektorientierung **Beispiel 2: Überschreiben**



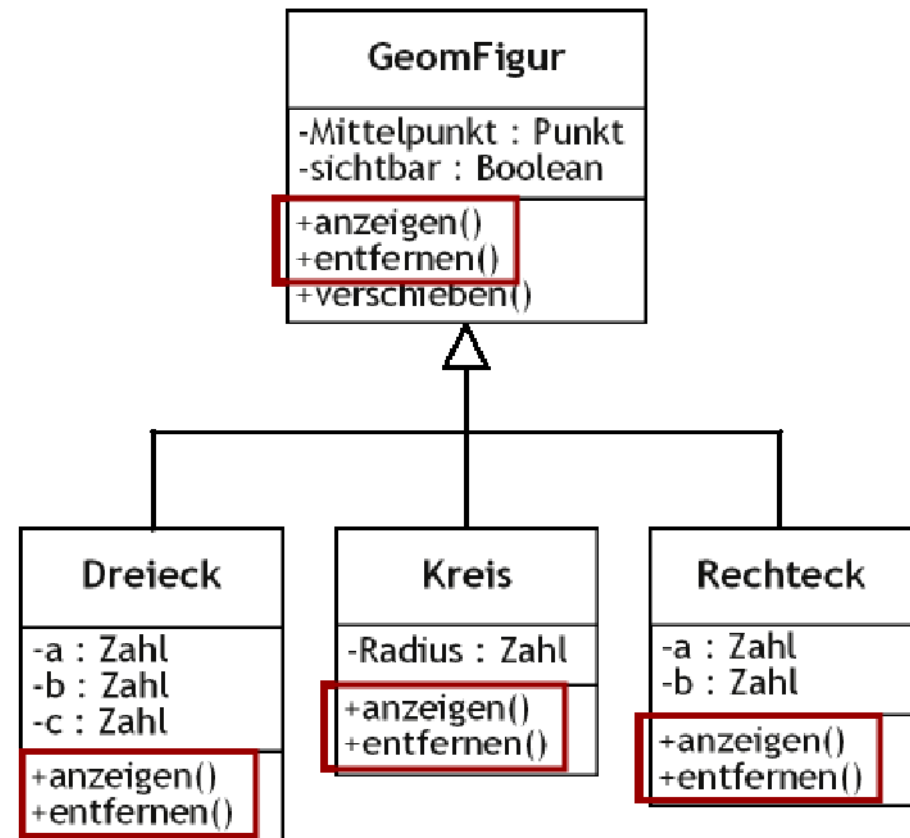
Objektorientierung Vererbung und Spezialisierung II

- Unterklassen unterscheiden sich systematisch voneinander
- Attribute und Methoden einer Klasse bilden jeweils ein in sich geschlossenes Konzept
- Die spezialisierte Klasse ist vollständig kompatibel zur allgemeinen Klasse
- Attribute und Methoden der Oberklasse müssen in der Spezifikation der Unterklasse nicht wiederholt werden

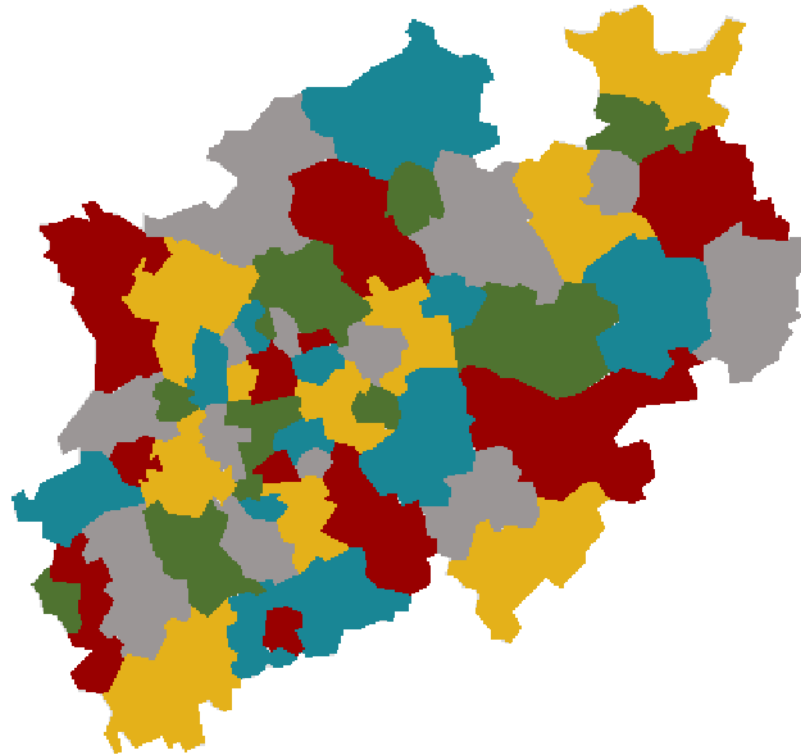


Objektorientierung **Polymorphie (Vielfgestaltigkeit)**

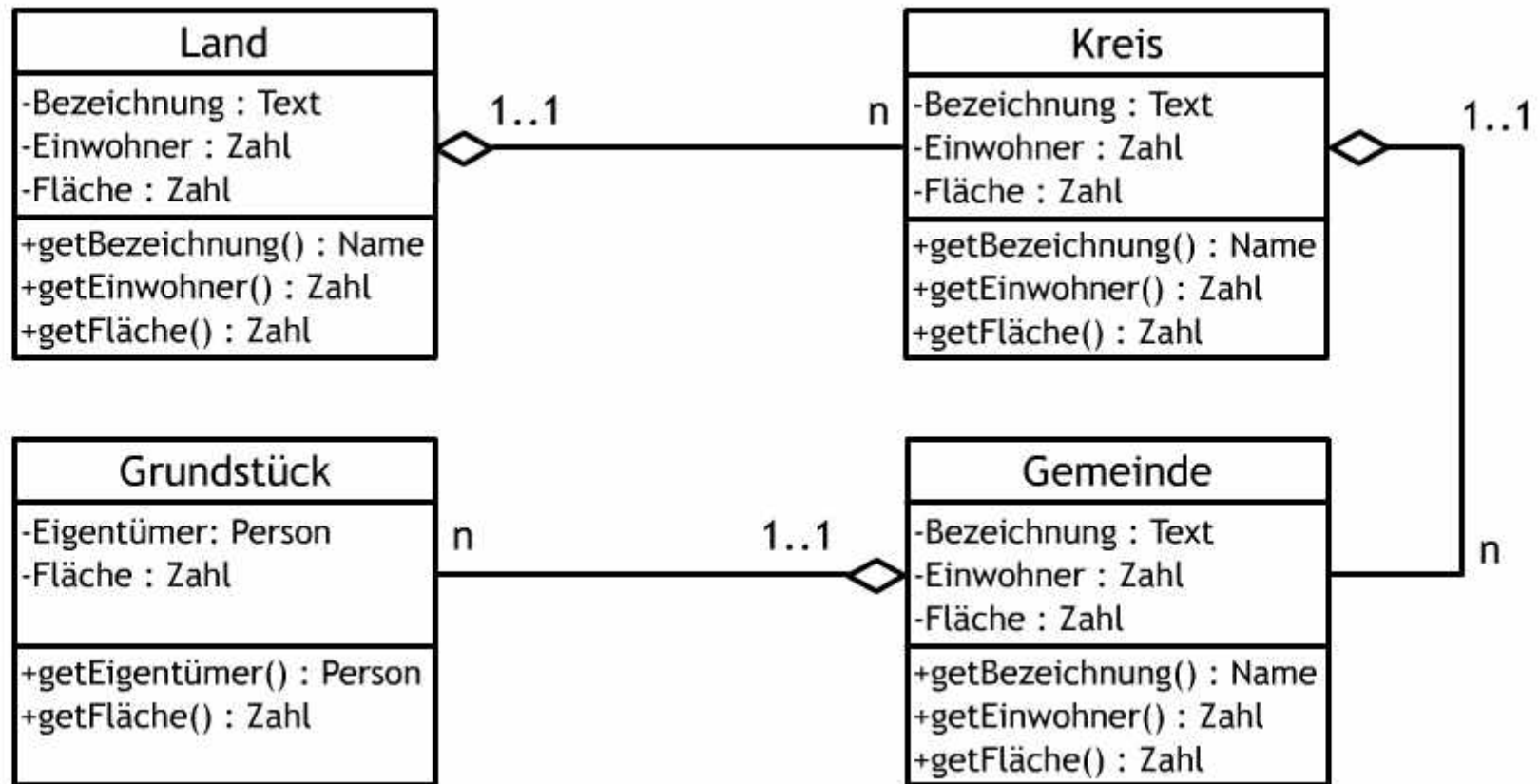
- **Gleichartige Methoden**, die auf Objekten verschiedener Klassen auszuführen sind, können **gleich benannt** werden
- Beim Aufruf einer solchen Methode wird jeweils die objektspezifische aktiviert
- **Vorteil** der Polymorphie: Man muss sich weniger Namen merken



Beispiel 2 Übung mit Raumbezug



Beispiel 2 Land - Kreis - Gemeinde - Grundstück



Beispiel 2 Grundstück - Flurstück - Polygon

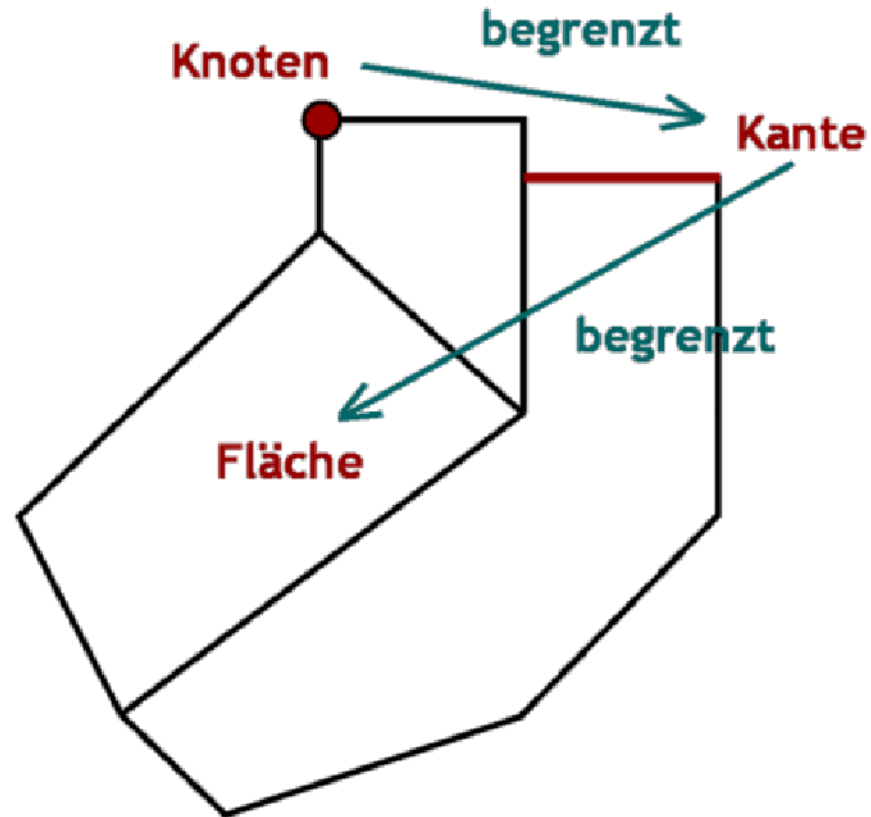
Läßt sich diese Unterteilung des Raumes über das Objekt "Grundstück" hinaus fortsetzen?

Zunächst:

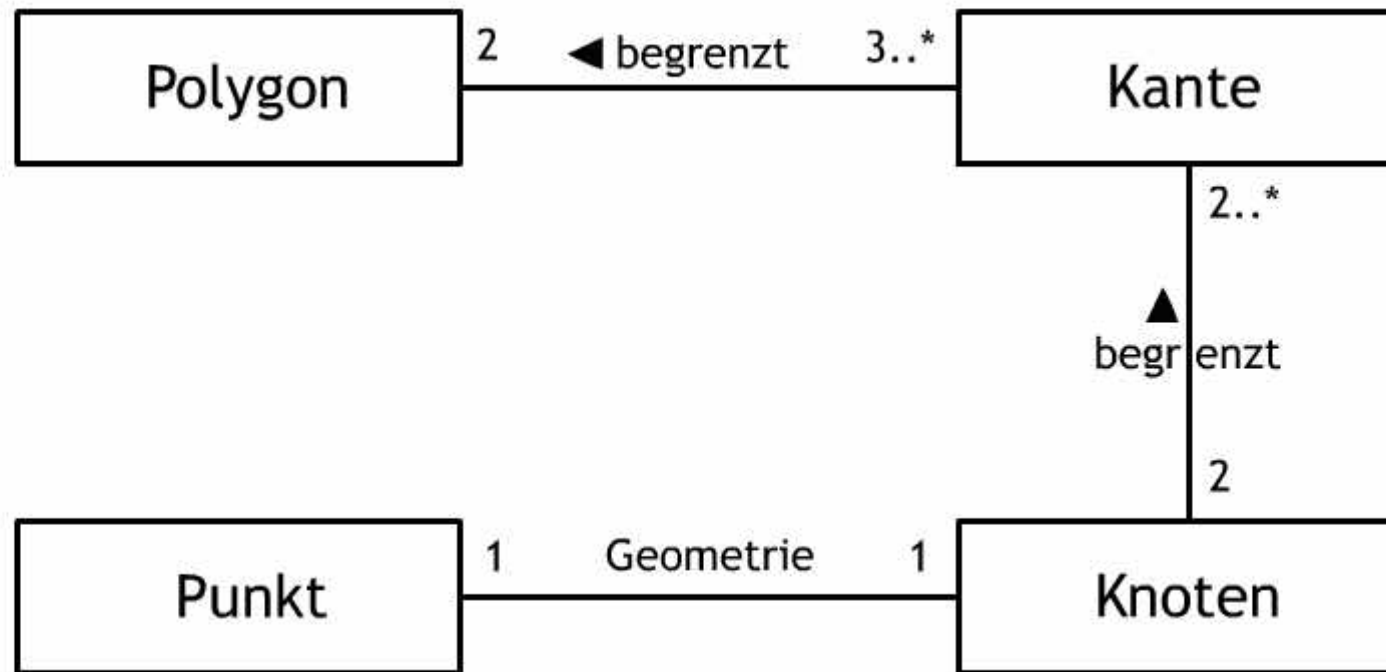
Übergang vom rechtlichen Objekt "Grundstück" auf das geometrische Objekt "Polygon"



Beispiel 2 Topologische Beziehungen von Polygonen



Beispiel 2 Polygon - Kante - Knoten - Punkt



Beispiel 2 **Klasse Punkt**

Punkt
-Länge : Grad -Breite : Grad -X : Zahl -Y : Zahl -Projektion : Text -Erfassung : Text = GPS -Erfassungsdatum : Datum
+getLänge() : Grad +getBreite() : Grad +setLänge(L : Grad) +setBreite(B : Grad) +getX() : Zahl +getY() : Zahl +...



Datenmodellierung **Literatur**

Balzert, Heide: Lehrbuch der Objektmodellierung. Akademischer Verlag, 1999

Oestereich, Bernd: Objektorientierte Softwareentwicklung: Analyse und Design mit der Unified Modeling Language. 4. Auflage - Oldenbourg, München Wien, 1998

