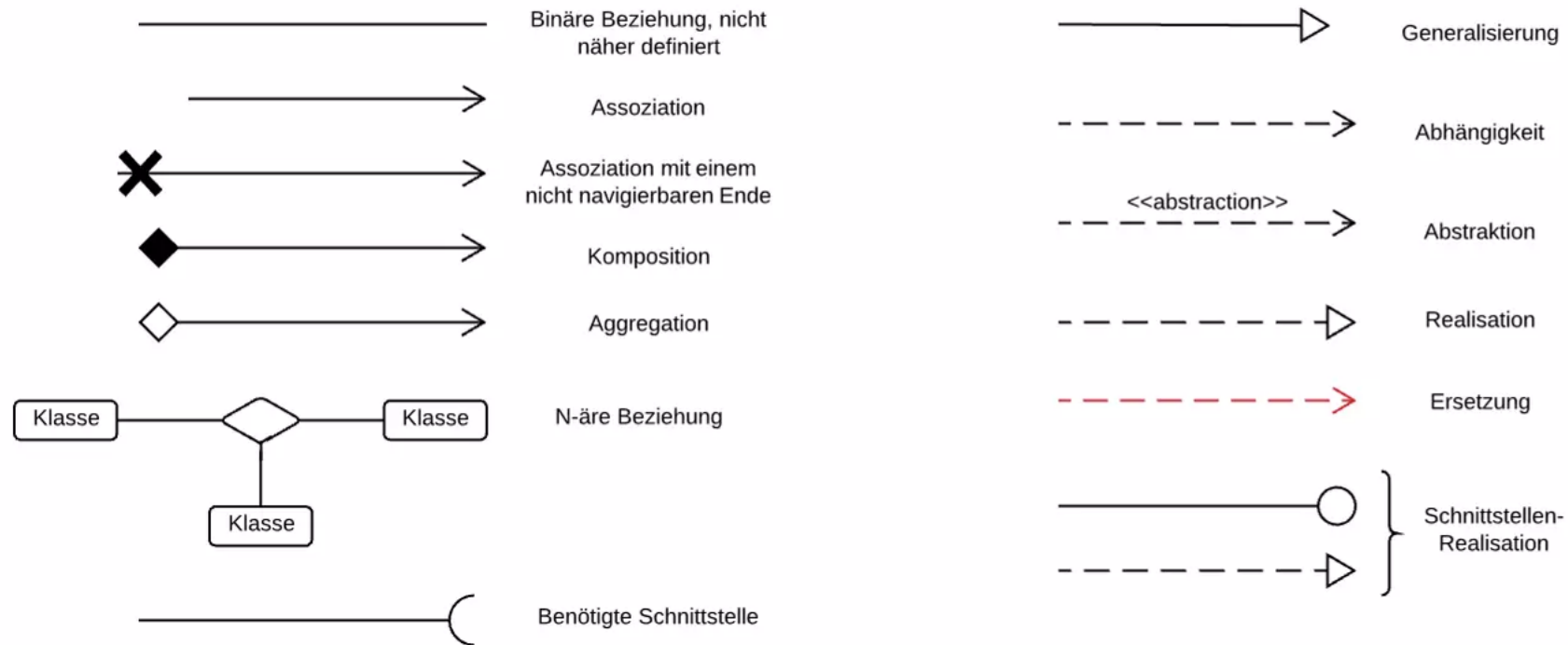


# UML-Klassendiagramme

**Beziehungen zwischen Klassen**

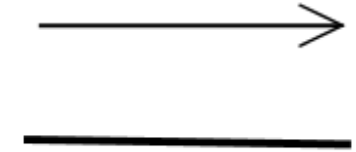
## Beziehungen



# Warum überhaupt?

- Warum benötigen wir eine Beziehung zwischen den Klassen?
  - Modellieren, wie Klassen miteinander interagieren
  - Strukturierte Softwarearchitektur
  - Erleichtert Wartung und Erweiterung

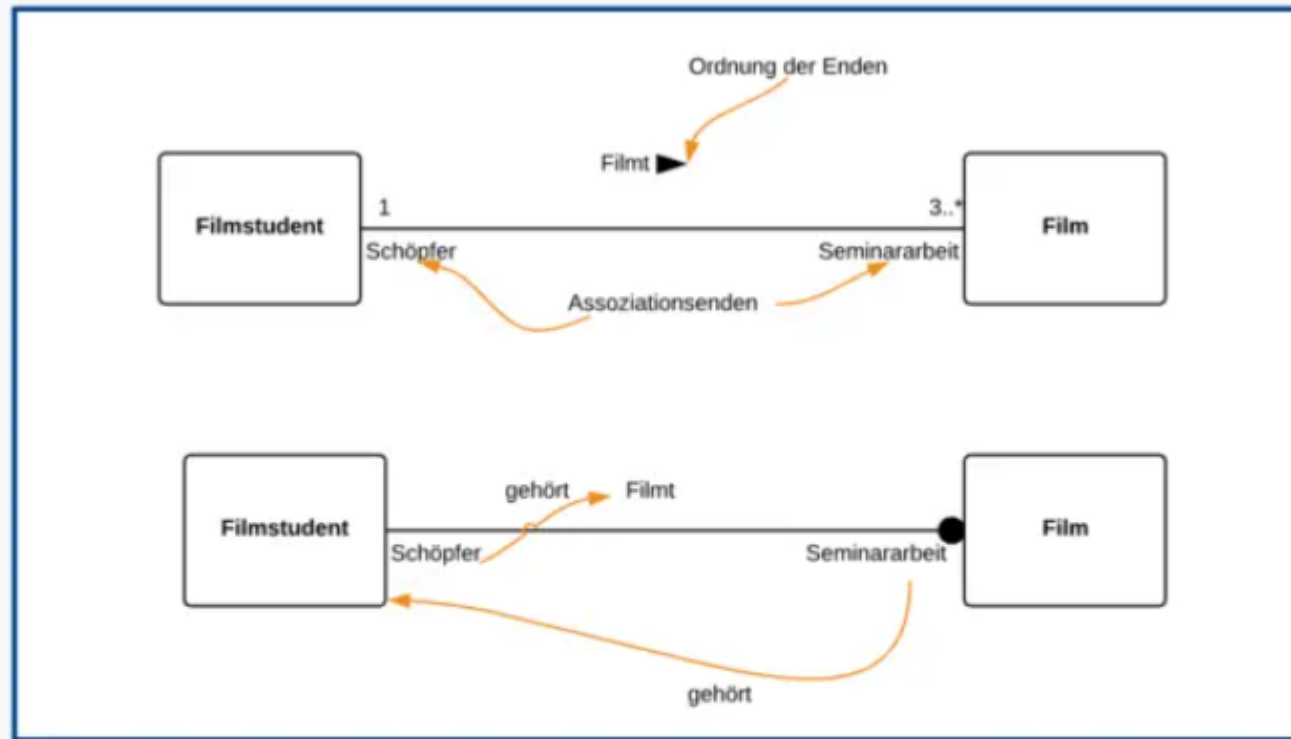
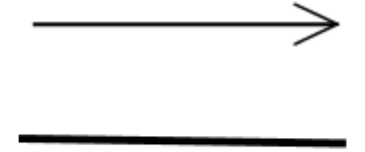
# Arten von Beziehungen:Assoziation



**Assoziation** (einfache Verbindung zwischen Klassen)

- Die Assoziation stellt die **Kommunikation zwischen zwei Klassen** im Diagramm dar
- Die Klassen werden mit einer einfachen Linie verbunden.
  - Mithilfe eines Pfeils kann man eine gerichtete Assoziation modellieren.
- Jede Assoziation kann mit einem Namen versehen werden, der die Beziehung näher beschreibt
- Der Name kann zusätzlich mit einer Leserichtung – einem kleinen ausgefüllten Dreieck – versehen werden
- Umgangssprachlich wird diese Beziehung manchmal als "**kennt**"-**Beziehung** bezeichnet
- Die Beziehung kann mit einer Kardinalität versehen werden (1:1, 1:\*, 0..1, usw.)

# Arten von Beziehungen: Assoziation



IONOS

Die Instanz „Filmstudent“ hat die Rolle „Schöpfer“. Sie ist verbunden mit der Instanz „Film“ über die Assoziation „Filmt“. Über der Assoziation kann eine Multiplizität eingetragen werden, wie im oberen Diagramm zu sehen.

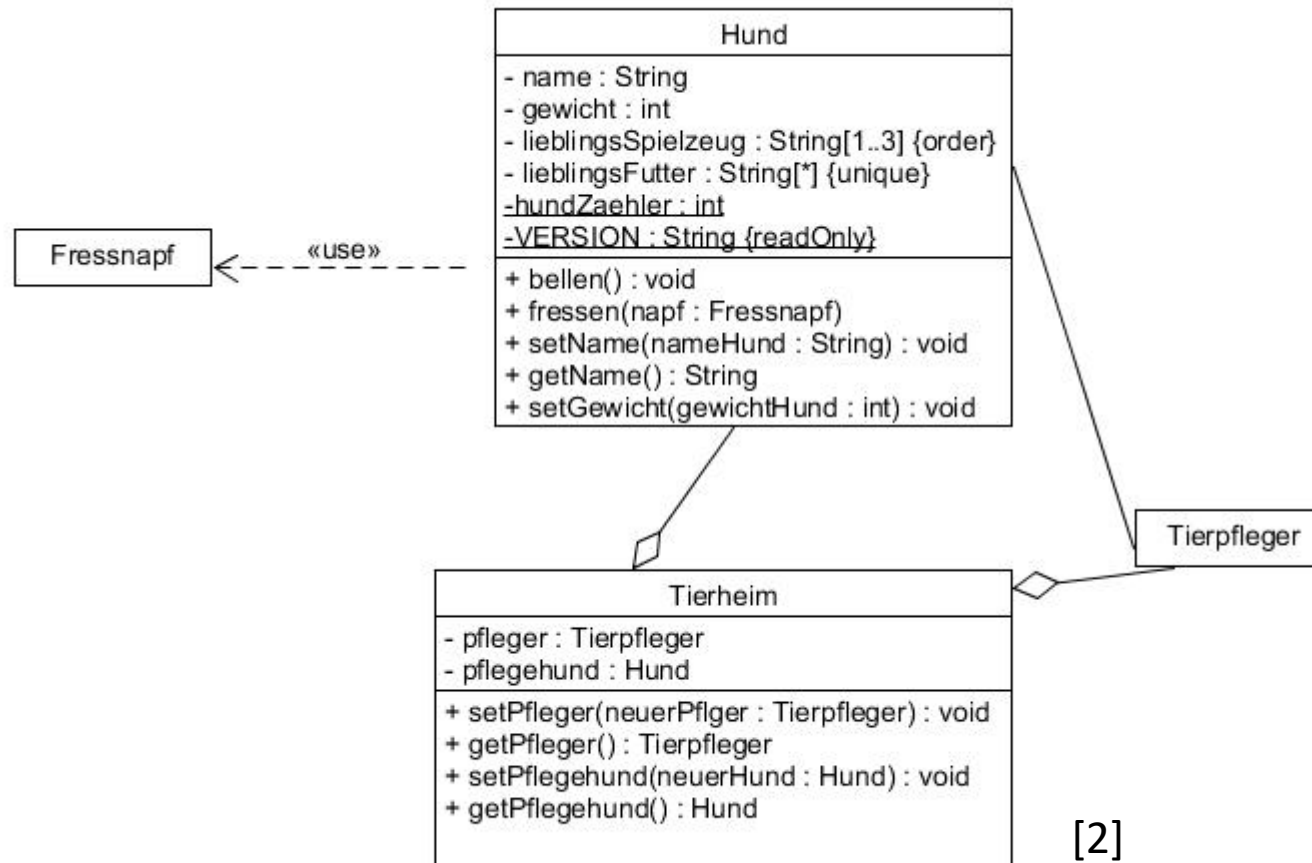
# Arten von Beziehungen: Aggregation



**Aggregation** (schwache Beziehung, Teil kann unabhängig existieren)

- Sie wird statt einer Assoziation verwendet, wenn zwischen zwei Klassen eine "**Teil-Ganzes-Beziehung**" besteht, das heißt, dass sich ein Objekt dieser Klasse unter anderem aus einer bestimmten Anzahl von Objekten der anderen Klasse zusammensetzt
- Beispiel: „Ein Team besteht aus mehreren Spielern.“

# Arten von Beziehungen: Aggregation

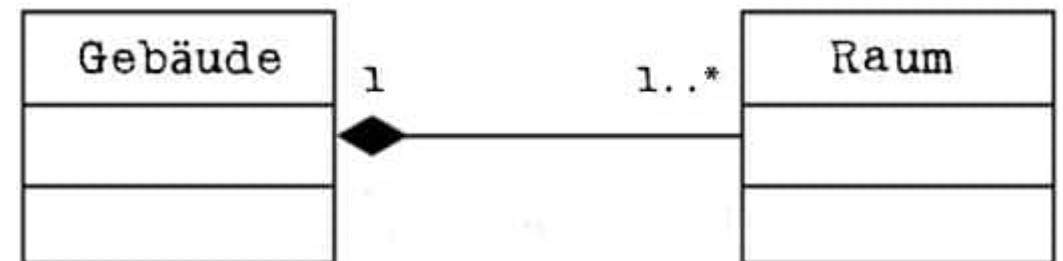


# Arten von Beziehungen:Komposition



**Komposition** (starke Beziehung, Teil existiert nur mit dem Ganzen)

- Bei einer Komposition kann ein Teil nie größer als das Ganze sein
- Eine Aggregation wird wie eine Assoziation als Linie zwischen zwei Klassen dargestellt und zusätzlich mit einer kleinen Raute versehen. Die Raute steht auf der Seite des Aggregats, also des Ganzen. Sie symbolisiert gewissermaßen das Behälterobjekt, in dem die Einzelteile gesammelt sind.
- Ein Gebäude besteht aus Räumen. Das Gebäude ist das Ganze, die Räume sind Teile des Ganzen.



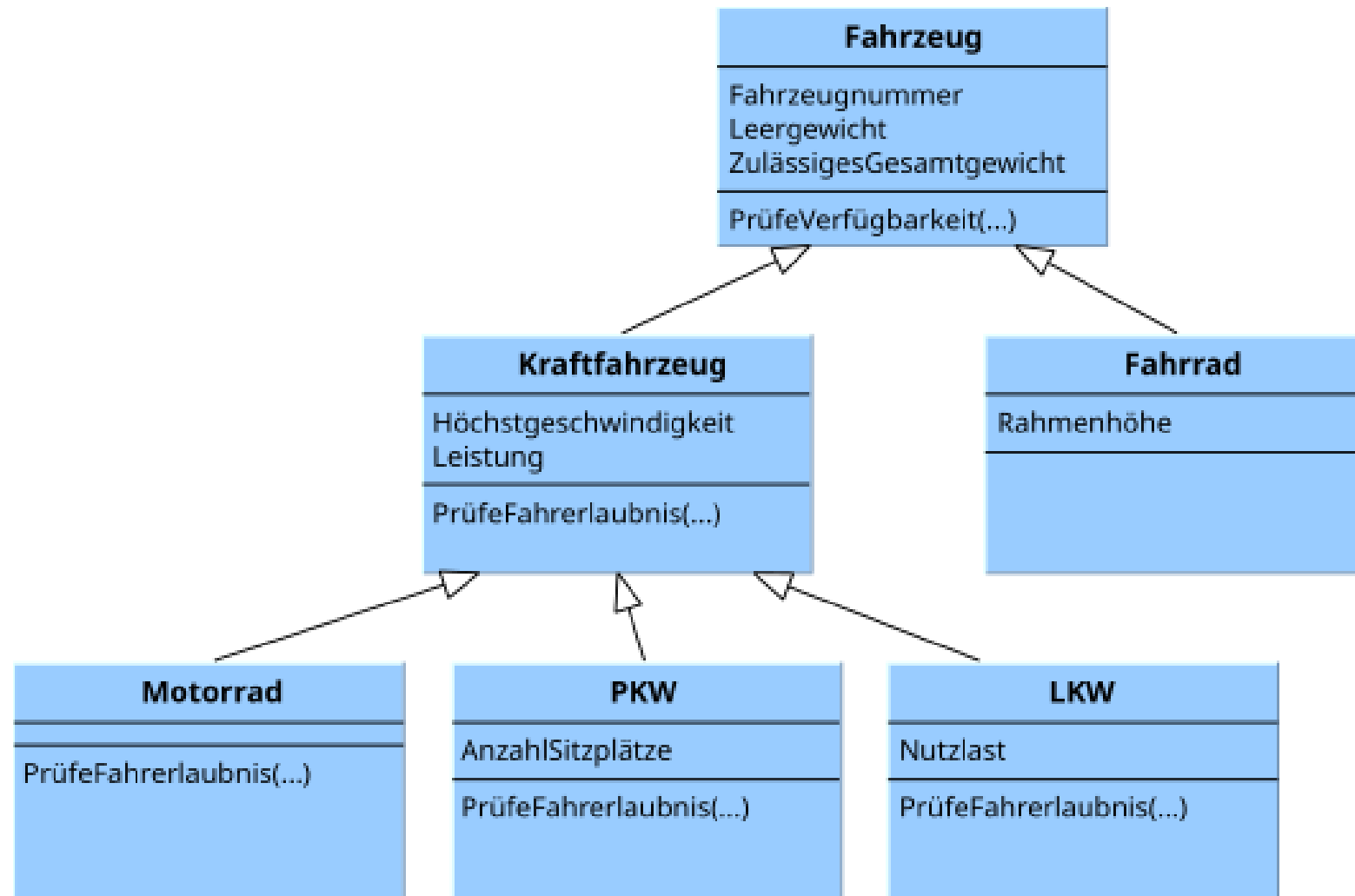


# Arten von Beziehungen: Generalisierung

**Generalisierung (Vererbung)** (eine Klasse erbt von einer anderen)

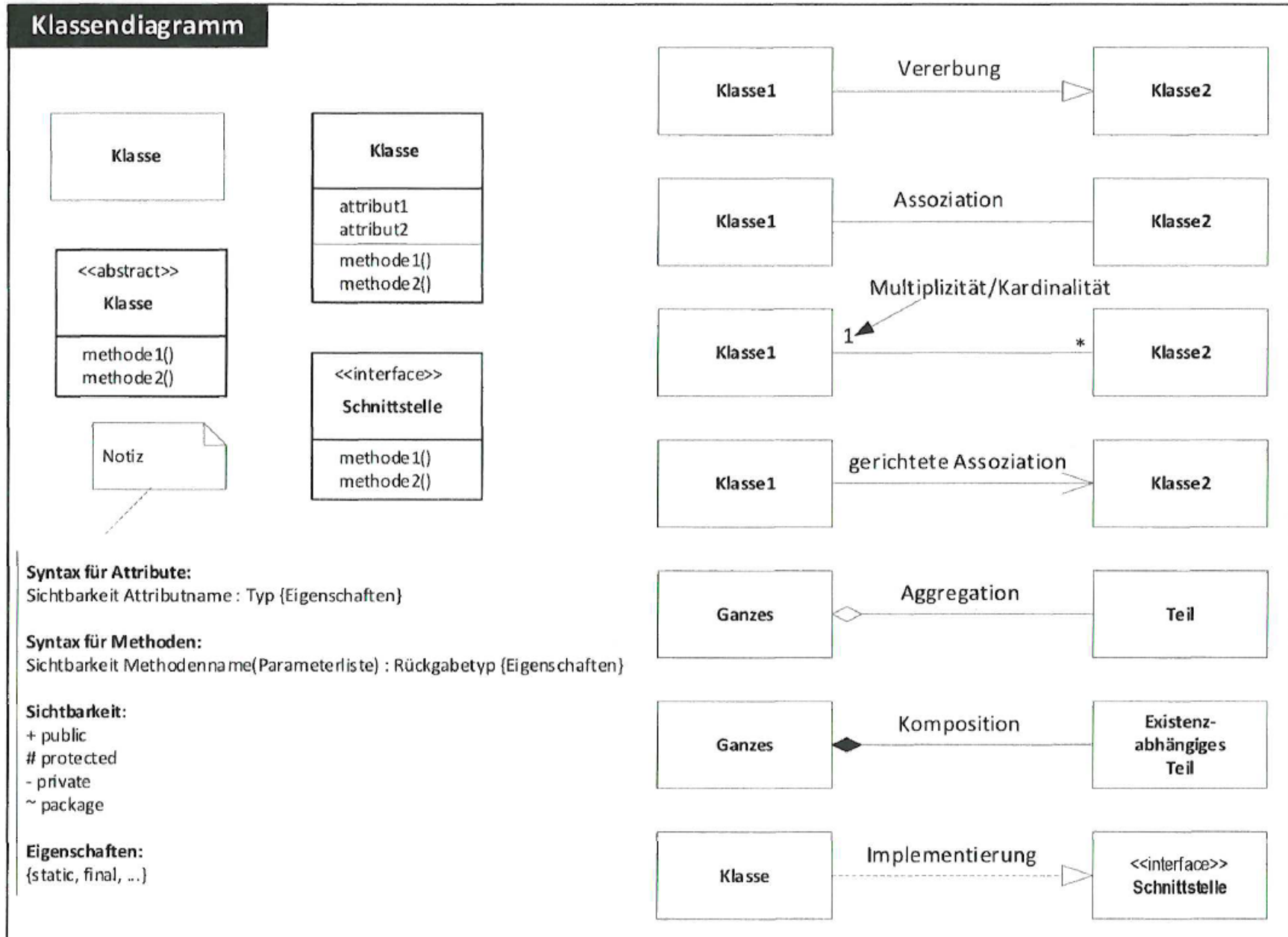
- Eine Entität basiert auf der anderen und erbt die Attribute, Operationen und Beziehungen der übergeordneten Entität.
- Das übergeordnete Modell kann viele untergeordnete Klassen haben und ebenso kann eine untergeordnete Klasse mehrere übergeordnete Modelle haben
- Oberklasse enthält gemeinsame Attribute/Methoden
- Unterklassen erben Eigenschaften der Oberklasse
- Beispiel: "Fahrzeug" als Oberklasse mit „Auto“ und „Fahrrad“ als Unterklassen

# Arten von Beziehungen: Generalisierung



# IHK

Auszug aus IHK-  
Abschlussprüfung  
Winter 2023/24



# Übung: Beziehungen erkennen

**Aufgabe:** Identifiziere die Beziehungen in folgendem Szenario:

"Ein Bibliothekssystem hat Bücher, Autoren und Mitglieder. Jedes Buch hat genau einen Autor, ein Mitglied kann mehrere Bücher ausleihen."

- Welche Klassen gibt es?
- Welche Beziehungen bestehen zwischen ihnen?
- Skizziere ein UML-Klassendiagramm

# Quellen

- [1] <https://www.ionos.at/digitalguide/websites/web-entwicklung/klassendiagramme-mit-uml-erstellen/#c147252>
- [2] <https://www.codeadventurer.de/?p=2428>
- [3] <https://t2informatik.de/wissen-kompakt/uml-komposition/>
- [4] [https://de.wikipedia.org/wiki/Vererbung\\_%28Programmierung%29](https://de.wikipedia.org/wiki/Vererbung_%28Programmierung%29)