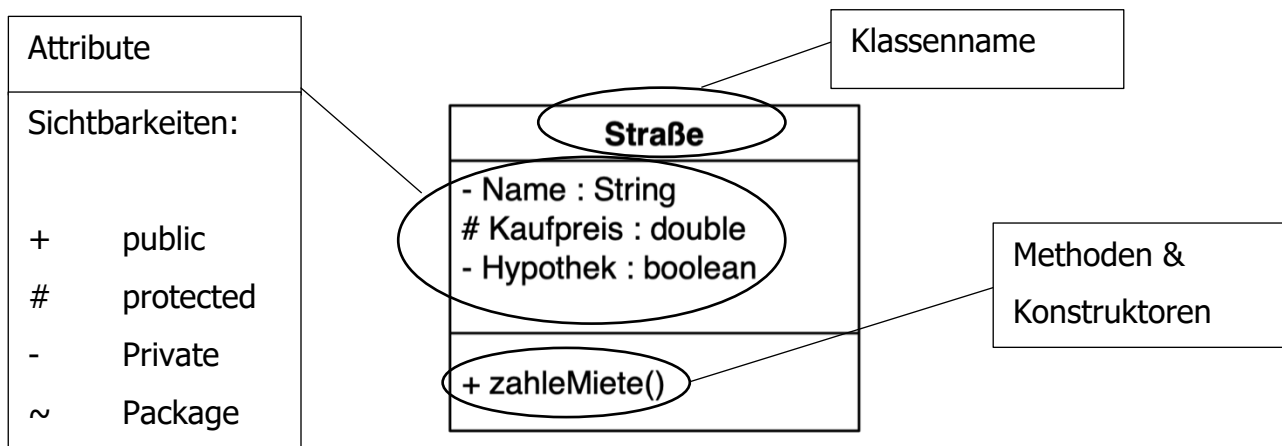
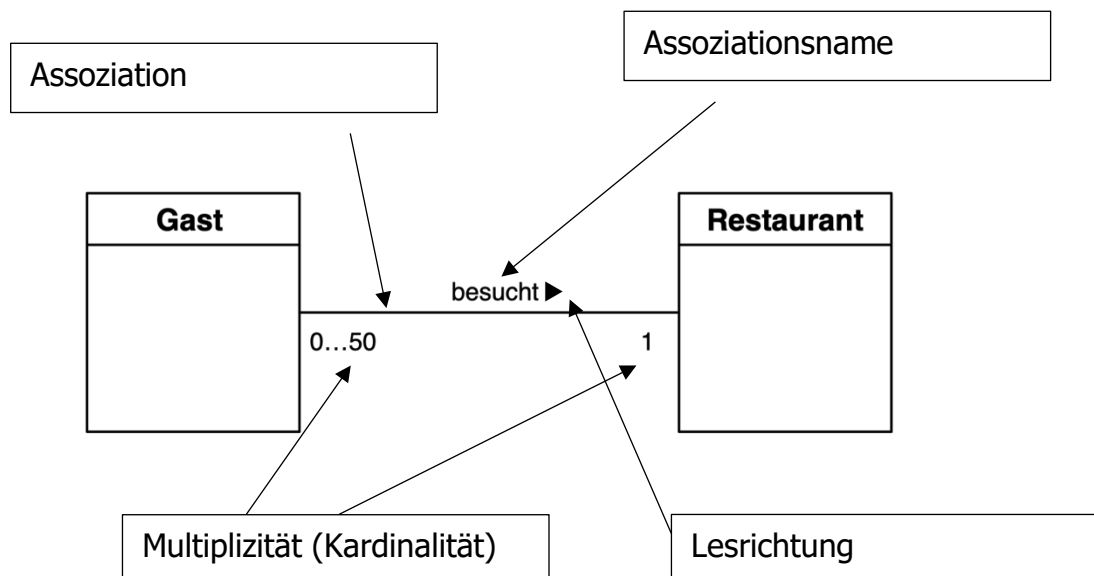


Das UML-Klassendiagramm – Assoziationen

Das Klassendiagramm modelliert die statischen Elemente eines Systems, ihre Eigenschaften und Beziehungen zueinander. Klassendiagramme stellen das zentrale Konzept der UML dar und sind aus der heutigen objektorientierten Softwareentwicklung nicht mehr wegzudenken. Sie zeigen die statischen Bestandteile und Attribute von Systemen und welche Beziehungen sie untereinander einnehmen können. Bis auf die Definition von Operationen (Methoden) der Klassen werden jegliche dynamischen Aspekte außer Acht gelassen. Klassendiagramme werden üblicherweise in den ersten beiden Phasen Analyse und Entwurf erstellt, können jedoch auch während der Phasen Implementierung, Test und Einsatz gewinnbringend verwendet werden.



Beziehungen zwischen den Klassen werden über sogenannte Assoziationen realisiert. Mit Hilfe von Assoziationen wird angezeigt, ob zwischen den Elementen eine referentielle Beziehung besteht.

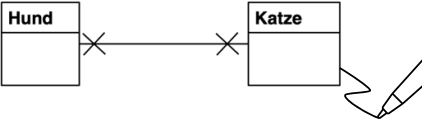
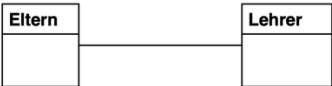
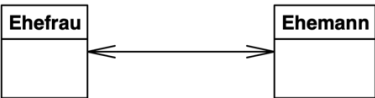
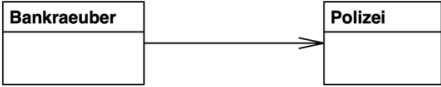
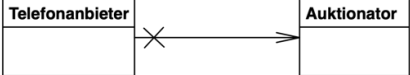




Arbeitsauftrag

Assoziationen können unterschieden werden in **bidirektionale Navigierbarkeit**, **Verbot der Navigierbarkeit**, **unspezifische Navigierbarkeit**, **Unidirektionale Navigierbarkeit** und **teilweise spezifizierte Navigierbarkeit**.

Ordnen Sie die oben aufgeführten Begriffe den Abbildungen zu.

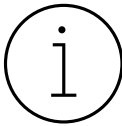
Abbildung	Erklärung
 <p>Verbot der Navigierbarkeit</p>	Untersagt jegliche Kenntnis der beteiligten Klassen voneinander. Kein der Instanzen einer Klasse kann damit eine Operation der Instanzen ihres Assoziationspartner aufrufen.
 <p>unspezifische Navigierbarkeit</p>	Hierbei verzichtet man darauf einen Zwang oder ein Verbot der Navigierbarkeit auszusprechen. Die Navigation von einer zu anderen Klasse ist erlaubt, aber nicht zwingend notwendig.
 <p>bidirektionale Navigierbarkeit</p>	Erlaubt, dass sich beide Klassen kennen dürfen. Die Instanzen beider Klassen können damit in beide Richtungen miteinander interagieren (z.B. gegenseitig Operationen aufrufen)
 <p>Teilweise spezifizierte Navigierbarkeit</p>	Hierbei ist die Navigation nur teilweise spezifiziert.
 <p>unidirektionale Navigierbarkeit</p>	Hierbei wird die Navigation in nur eine Richtung erlaubt.

Lösung zum Vergleichen:

<https://kurzelinks.de/6btu>



Aggregation und Komposition

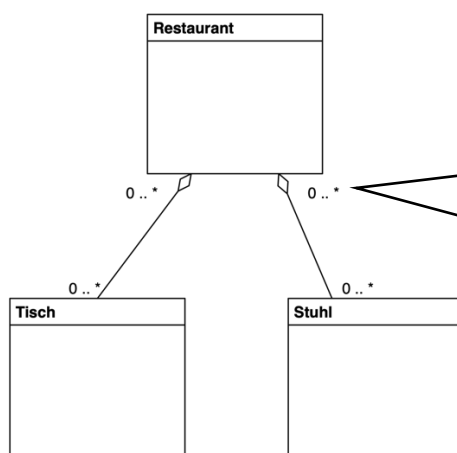


Zu Beginn ein Beispiel ...

Wir bleiben im Restaurant. In diesem befinden sich für gewöhnlich Tische und Stühle. Die Anzahl von Tischen und Stühlen im Restaurant kann sich stetig verändern. So müssen für größere Feste (z.B. Hochzeiten) mehr Tische und Stühle in den Raum gestellt werden, um die große Anzahl an Gästen unterzubringen. Demgegenüber werden evtl. in der Weihnachtszeit, wenn noch ein Weihnachtsbaum im Raum steht, ein Tisch mit Stühlen rausgenommen, um Platz für den Baum zu haben.

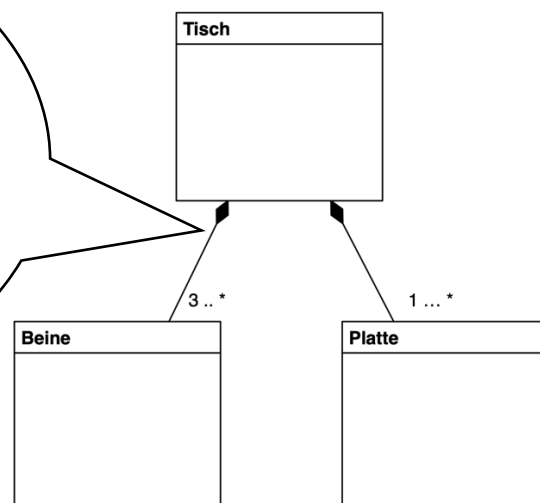
Betrachtet man nun den Tisch etwas näher so wird klar, dass dieser aus Tischbeinen (mind. 3) und einer Platte besteht. Diese Teile sind zwingend erforderlicher, sonst wäre es kein Tisch.

Diese einfachen Beispiele, angelehnt an die reale Welt, zeigen Aggregation und Komposition.



Immer dann, wenn **nach** der Instanziierung des Ganzen, nachträglich noch Teilelemente **hinzugefügt** werden können, handelt es sich um eine **Aggregation**

Immer dann, wenn nach der Instanziierung des Ganzen, nachträglich **keine** Teilelemente **hinzugefügt** werden können, handelt es sich um eine **Komposition**.





Arbeitsauftrag

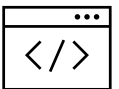
Bei Monopoly gibt es verschiedene Straßen. Jede Straße kann von einem Spieler erworben werden. Es liegt ein vereinfachter Ausschnitt der Klasse Straße und Spieler wie folgt vor:

Spieler
<ul style="list-style-type: none"> - Spielername : string - Konto : decimal
<ul style="list-style-type: none"> + Spieler (Spielername : string, Konto : decimal) + getName() : string

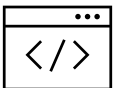
Straße
<ul style="list-style-type: none"> - Strassenname : string - Miete : decimal - Besitzer : Spieler
<ul style="list-style-type: none"> + Strasse (Strassenname : string, Miete : decimal) + addBesitzer(Besitzer : Spieler) : void + removeBesitzer() : void + getBesitzer() : Spieler



- Gehen Sie davon aus, dass die Straße zu Beginn keinen Besitzer hat, aber jeder Zeit ein Besitzer hinzugefügt werden bzw. entfernt werden kann. Handelt es sich hierbei um eine Aggregation oder Komposition? **Vervollständigen** Sie das obere Klassendiagramm und **programmieren** Sie in einer **objektorientierten Sprache Ihrer Wahl** die Klasse **Straße** und **Spieler** gemäß der Klasse von oben aus.
- Programmieren** Sie die Klasse Straße so um, dass bei der **Instanziierung immer ein Besitzer vorliegen muss**, welcher auch nicht geändert werden kann. Zeichnen Sie auch das Klassendiagramm oben neu. Handelt es sich hierbei um eine Aggregation oder Komposition?



aggregation



Lösungshinweise Beispiel Python:

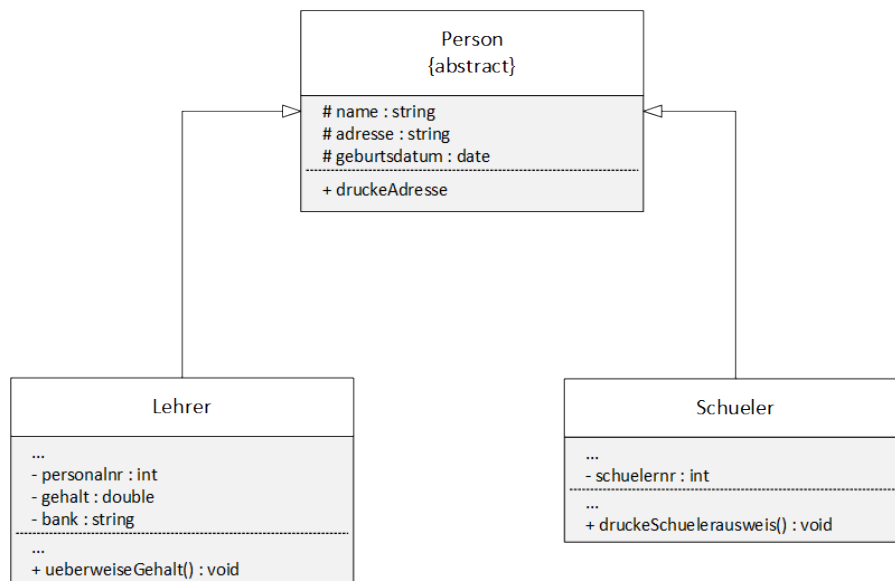
<https://t1p.de/3gsdj>





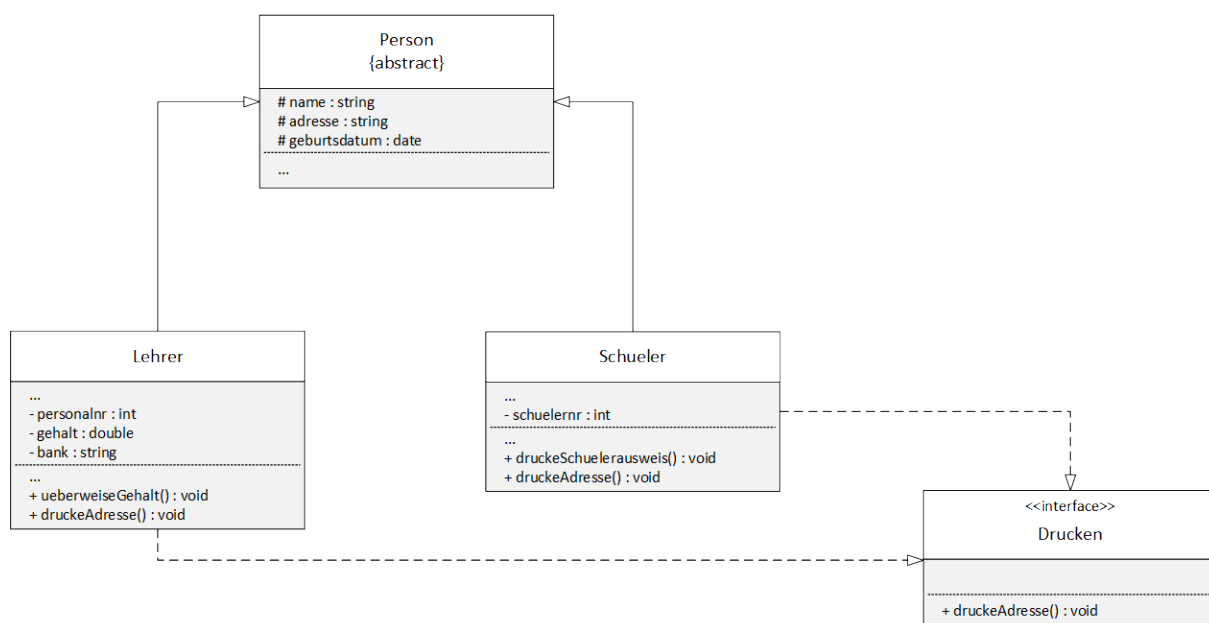
Abstrakte Klassen und Interfaces

In manchen Fällen kann es nötig sein, dass eine Klasse nicht instanziiert werden kann (bedeutet: es soll keine Objekte dieser Klassen geben).



Erst durch eine Vervollständigung und Konkretisierung der abstrakten Klasse unter Einsatz der Spezialisierung wird es möglich, einen für die Erzeugung von Objekten (Bsp. Lehrer und Schüler) geeigneten Bauplan zu spezifizieren.

Eine weitere Besonderheit ist die Schnittstelle oder auch Interface. Diese können nicht instanziiert werden, weil sie selbst und alle ihre Operationen immer als abstract angesehen werden. Sie ähneln konzeptuell den abstrakten Klassen und werden wie diese von anderen Klassen realisiert. Eine Schnittstelle enthält u.a. die Spezifikation der Aufrufe von Operationen, macht aber keine Aussage zur Operation selbst. Eine Schnittstelle kann von beliebig vielen Klassen realisiert werden bzw. implementiert werden, wie auch eine Klasse beliebig viele Schnittstellen realisieren kann.

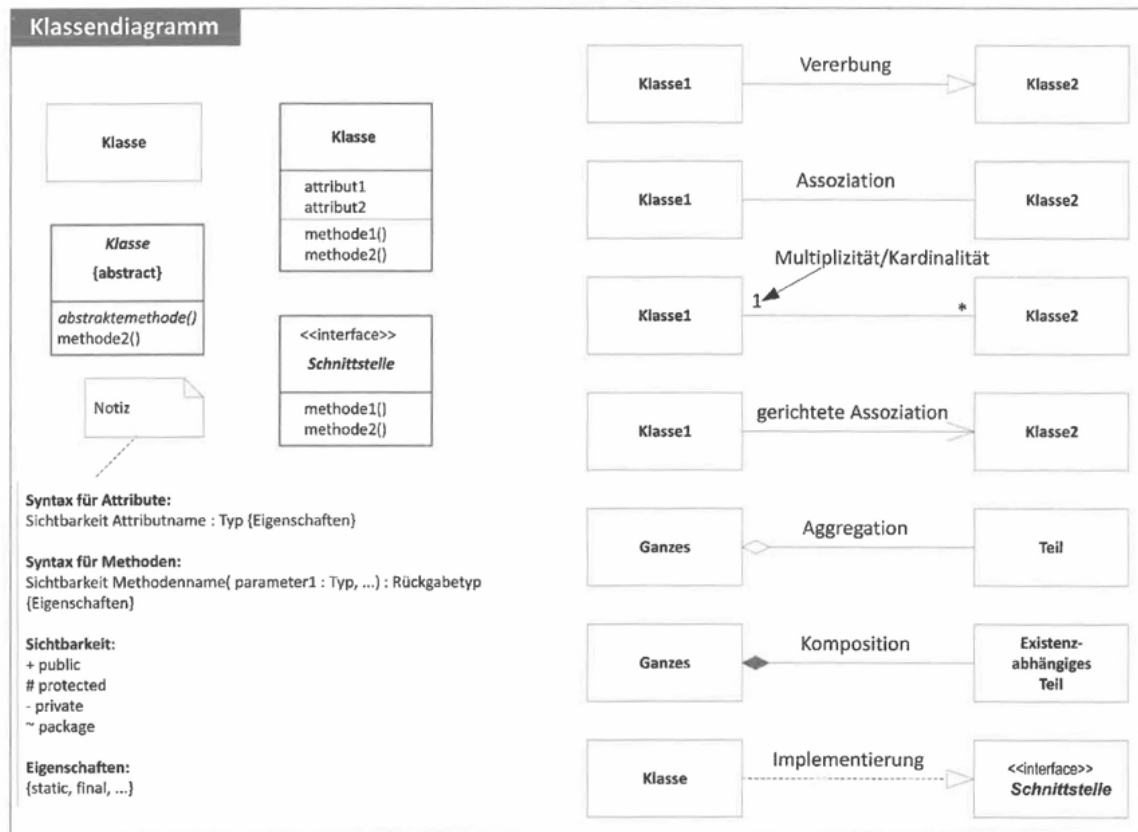




UML Belegsatz IHK

In der Abschlussprüfung bekommen Sie einen Belegsatz mit den Notationen zu den jeweiligen UML-Diagrammen. Hier sehen Sie den Auszug zum Klassendiagramm gemäß IHK:

UML-Klassendiagramm



Die Notation entspricht in der aktuellen Version des Belegsatzes der IHK auch dem UML-Standard. Ältere Prüfungsaufgaben, die auch in diesem Modul zu Übungszwecken bearbeitet werden sollen, haben eine abweichende Notation (z.B. in Bezug auf abstrakte Klassen). Im weiteren Verlauf halten wir uns an den aktuellen Standard (s. Auszug Belegsatz oben).

Wenn Sie sich mit den aktuellen Regeln zu UML 2.5 informieren wollen, schauen Sie unter folgendem Link nach:

	https://t1p.de/f63yb	
--	---	--

Stand: Januar 2025