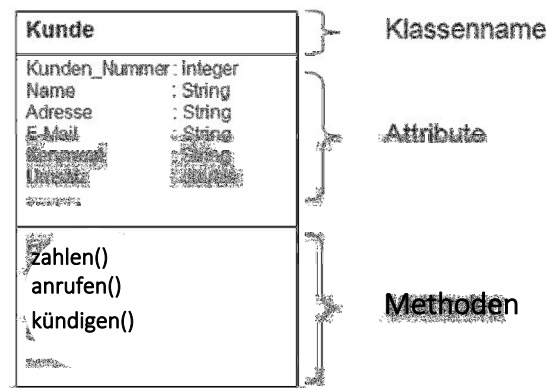
	Skript: UML – Unified Modelling Language	Sommer (SR)
	Das Klassendiagramm	Datum:

7 Das Klassendiagramm

Für die Analyse und das Design von Softwaresystemen ist das Klassendiagramm das wichtigste Diagramm. Hier werden Klassen und deren Beziehungen untereinander statisch dargestellt.

7.1 Die Darstellung einer Klasse

Eine Klasse besteht aus drei wesentlichen Teilen: dem Klassennamen, den Attributen und den Methoden:




7.1.1 Attribute

Attribute haben unterschiedliche **Sichtbarkeiten**. Sie können privat oder öffentlich sein. Dazu bietet das Klassendiagramm folgende Symbolik:

Symbol	Bedeutung/Sichtbarkeit
-	privates Attribut (private) - Ist gegen den Zugriff von außen geschützt und kann nur über entsprechende Methoden angesprochen werden.
+	öffentliches Attribut (public) - Kann direkt angesprochen werden und wird nur selten verwendet.
#	geschütztes Attribut – spielt bei der Vererbung eine Rolle (protected) - Verhält sich nach außen wie ein privates Attribut. Innerhalb der Vererbungshierarchie aber wie ein öffentliches. Eine Klasse, die von einer anderen erbt, kann auf deren geschützte Attribute zugreifen als wäre es ihre eigenen.

Soll ein Attribut spezifiziert werden, können **Datentyp** und **Multiplizität** angegeben werden. Die Multiplizität gibt an, wie viele Exemplare von diesem Attribut angelegt werden sollen. Das entspricht einem Array in der Programmierung.

	Skript: UML – Unified Modelling Language	Sommer (SR)
	Das Klassendiagramm	Datum:

Beispiel:

<i>Werte:</i>	<i>Integer [1..10]</i>
---------------	------------------------

Attribute können initialisiert werden. Das wird durch die Zuweisung eines Wertes dargestellt.

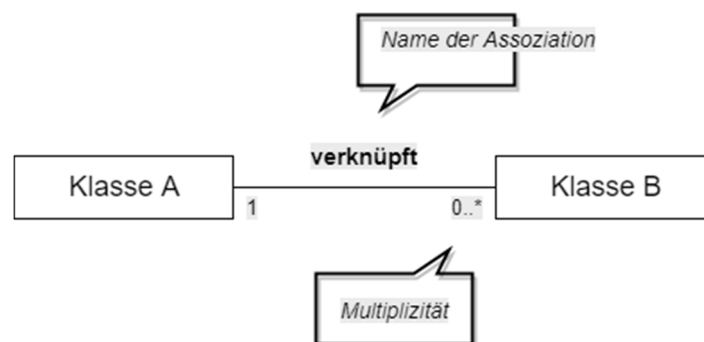
Beispiel:

<i>Provision:</i>	<i>Double = 0.5</i>
-------------------	---------------------

7.2 Beziehungen zwischen Klassen

7.2.1 Die Assoziation

Die Assoziation ist eine Beziehung zwischen zwei Klassen.



7.2.2 Die Aggregation


Bei der Aggregation beschreiben die verknüpften Klassen eine **Ganzes-Teile-Beziehung**. Eine Klasse verkörpert das Ganze und die zweite einen Teil davon. Die Aggregation wird mit einer Raute an dem Ganzen dargestellt.



7.2.3 Die Komposition

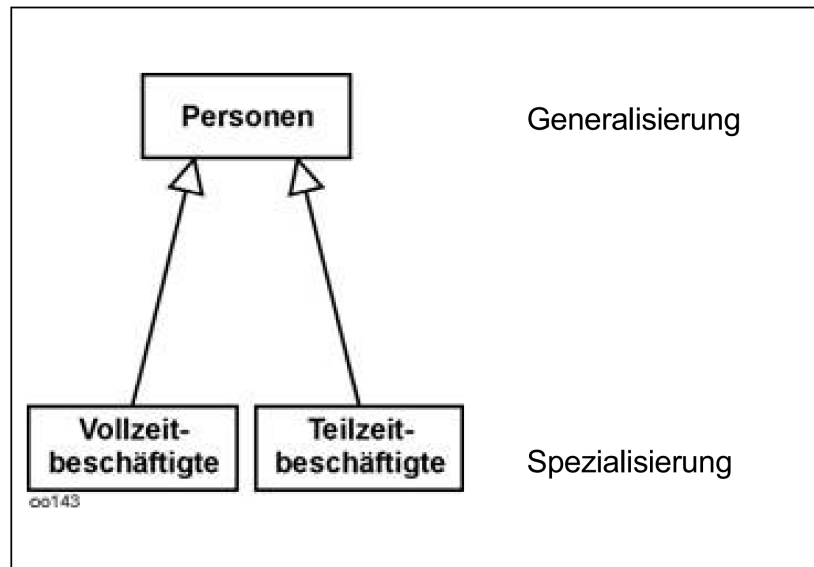
Die Komposition ist eine Aggregation mit einer besonderen Eigenschaft. Hier besteht zwischen dem Ganzen und den Teilen eine so starke Abhängigkeit, dass die Teile nicht ohne das Ganze existieren können. Die Raute ist in der Darstellung ausgefüllt



	Skript: UML – Unified Modelling Language	Sommer (SR)
	Das Klassendiagramm	Datum:

7.3 Generalisierung und Spezialisierung (Vererbung)

Das Konzept der Vererbung ist ein zentrales Thema in der OOP. Es erlaubt, Situationen aus der realen Welt besser abzubilden. Daneben hat dieses Konzept auch den Vorteil, dass bereits existierender Programmcode sehr gut wiederverwendet werden kann.

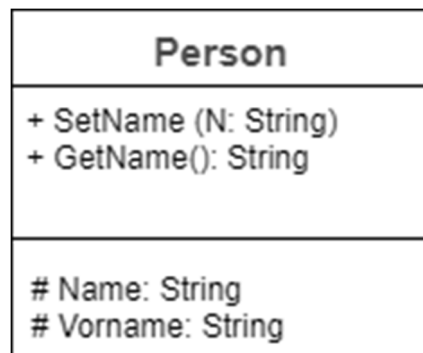



(Die Pfeile zeigen immer in Richtung Basisklasse!)

7.3.1 Sichtbarkeit von Attributen

Alle öffentlichen Elemente der Basisklasse können in der spezialisierten (vererbten) Klasse genutzt werden. Auf private Elemente kann nur über Methodenaufrufe zugegriffen werden. *Protected* ermöglicht es, Elemente nach außen hin zu schützen, aber den direkten Zugriff von spezialisierten Klassen zuzulassen.

Beispiel:



	Skript: UML – Unified Modelling Language	Sommer (SR)
	Das Klassendiagramm	Datum:

7.3.2 Mehrfachgeneralisierung

Bei der Mehrfachgeneralisierung (Mehrfachvererbung) generalisiert eine Klasse mehrere Klassen. Das ist häufig noch realitätsnäher als eine einfache Vererbung, aber deutlich komplexer. Aus diesem Grund gibt es in einigen Programmiersprachen wie z.B. Java diese Technik nicht.

7.3.3 Abstrakte Basisklassen

Eine Abstrakte Klasse dient als Grundlage für die weiteren Klassen einer Vererbungshierarchie. Von ihr können keine Objekte erzeugt (instanziert) werden. In einer abstrakten Basisklasse können Methoden als abstrakt deklariert werden. Sie sind hier noch nicht implementiert. Das geschieht erst zwingend in erben den Klassen.

Im Klassendiagramm werden abstrakte Methoden kursiv dargestellt.

