

UML Mini Cheat Sheet

Wir fokussieren uns in P1 auf UML *Klassendiagramme*, um die Inhalte von Klassen und deren Beziehungen zu visualisieren.

In einem UML Klassendiagramm wird jede Klasse in einem Rechteck mit drei Bereichen repräsentiert. Der erste Bereich enthält den Klassennamen, der zweite Bereich enthält die Variablen und der dritte Bereich die Methoden der Klasse. In Abb. 1 ist ein Klassendiagramm gezeigt.

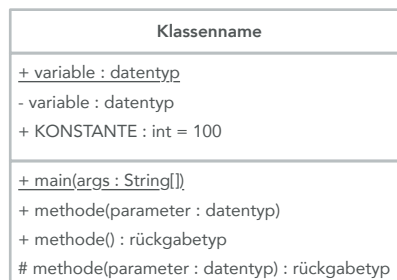


Abbildung 1: Ein UML Klassendiagramm.

UML wurde nicht speziell für Java entwickelt (UML ist eine sprachunabhängige Notationsmöglichkeit). Die Syntax in UML Diagrammen ist deshalb nicht identisch mit der Syntax von Java.

- Die Sichtbarkeiten der Variablen und Methoden werden mit einem Plus (+) für **public**, einem Minus (–) für **private** und mit einem Hash (#) für **protected** visualisiert.
- Nach dem Bezeichner einer Variablen folgt nach einem Doppelpunkt der zugehörige Datentyp.
- Optional kann man nach einem Gleichheitszeichen noch einen initialen Wert der Variablen oder der Konstanten definieren.
- Die Parameter einer Methode werden in Klammern nach dem Bezeichner der Methode ebenfalls nach dem Muster (*Parameter-Bezeichner : Datentyp*) aufgelistet.
- Nach den Parametern in Klammern wird nach einem Doppelpunkt der Rückgabotyp der Methode angegeben.
- Den Typ **void** gibt es in UML nicht. Das heisst, bei Methoden ohne Rückgabe wird in UML der Rückgabotyp einfach weggelassen.
- Bezeichner für statische Variablen und statische Methoden werden unterstrichen dargestellt.

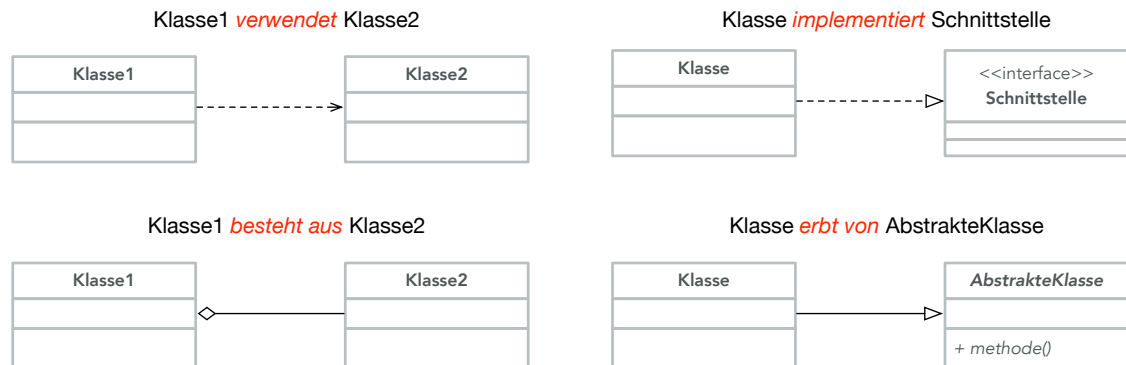


Abbildung 2: Verschiedene Beziehungstypen in einem UML Klassendiagramm.

Der Pfeil, der die zwei Klassen miteinander verbindet, zeigt an, dass eine Beziehung zwischen diesen beiden Klassen existiert. Jeder Beziehungstyp wird in UML mit anderen Typen von Pfeilen visualisiert (Siehe Abb. 2):

- Eine gestrichelte Linie mit offenem Pfeilkopf bedeutet, dass die eine Klasse die andere Klasse *verwendet*.
- Eine Beziehung *besteht aus* nennt man *Aggregation* und wird in UML mit einer ausgezogenene Linie mit leerer Raute dargestellt¹. Die Raute befindet sich auf der Seite des aggregierten Objektes.
- In UML wird eine Schnittstelle wie eine Klasse dargestellt, mit dem Unterschied, dass <<Interface>> oberhalb des Bezeichners der Schnittstelle steht. Eine gestrichelte Linie mit einer geschlossenen Pfeilspitze wird von der Klasse zur Schnittstelle gezeichnet und bedeutet *implementiert*.
- Mit einem ausgezogenen Pfeil mit weissem Kopf zeigt man Vererbung an (eine Subklasse *erbt von* einer Superklasse). Beachten Sie, dass in UML die Bezeichner von abstrakten Klassen und abstrakten Methoden kursiv dargestellt werden.

¹Die Aggregationen aus **String** Objekten macht man nicht explizit sichtbar. **String** Objekte sind so fundamental, dass diese in UML wie primitive Datentypen behandelt werden.