UML Klassendiagramme

Eine gute Möglichkeit Klassen und ihre Beziehungen grafisch darzustellen sind UML Klassendiagramme. UML ist eine Abkürzung für "Unified Modeling Language".

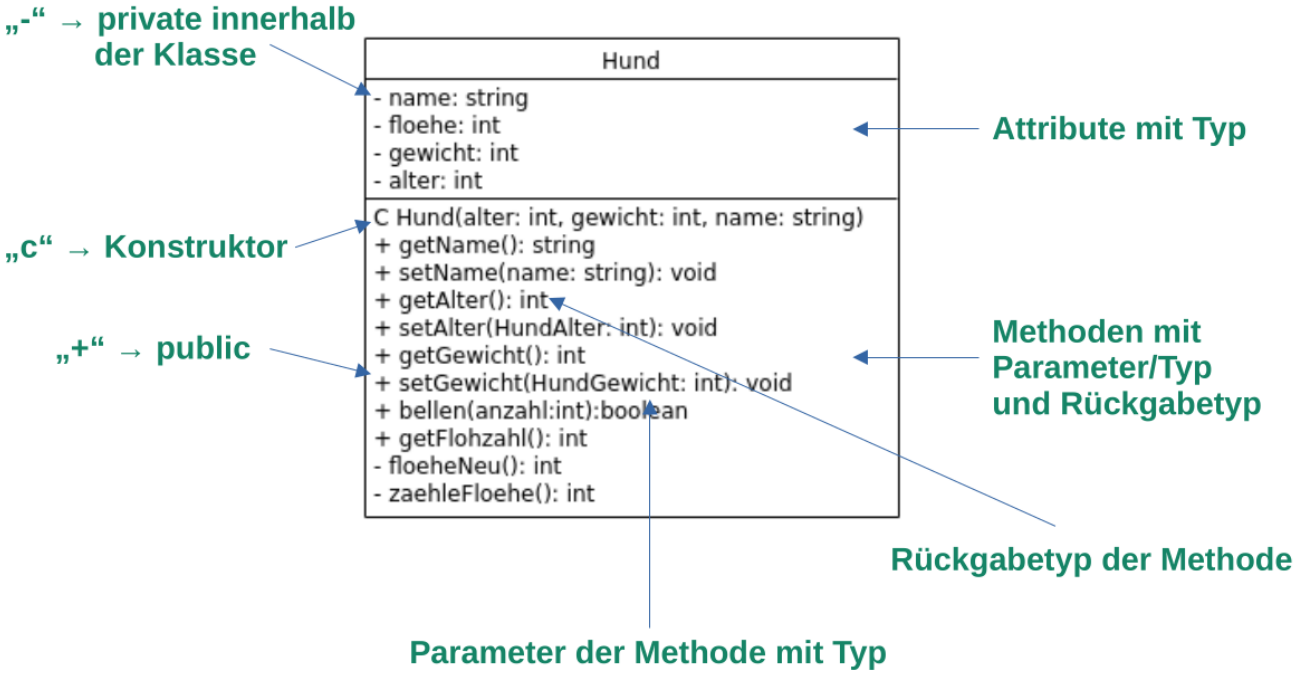
Klassen bestehen drei Teilen:

- Klassenname
- Eigenschaften (auch Attribute oder Instanzvariablen)
- Methoden

Aus einer Klasse können wir **Objekte** erzeugen, man spricht von "instanziieren". Ein bestimmtes Objekt ist eine Instanz einer Klasse. Aus der unten abgebildeten Klasse "Hund" kann man also eine Hunde-Objekt mit dem Namen "Higgs", dem Gewicht "20kg", und der Farbe "weiß" erzeugen.

- Die **Attribute** einer Klasse beschreiben den *Zustand* eines Objekts, wie z.B. Name und Gewicht eines Hundes.
- Die **Methoden** einer Klasse definieren das *Verhalten* eines Objekts - die Methoden geben dem Objekt Fähigkeiten, unser Hund kann beispielsweise bellen.

In einem UML Klassendiagramm werden die drei Bestandteile durch waagerechte Striche getrennt. Für das Hunde-Beispiel sieht das Klassendiagramm so aus:



- Im oberen Feld steht der Klassenname (Hund).
- Darunter folgen die Klassen-Attribute. Den Datentyp der Attribute schreibt man durch einen Doppelpunkt getrennt hinter den jeweiligen Attributs-Namen.
- Die Methoden werden mit ihrer Parameterliste und ihrem Rückgabewert im unteren Feld des UML Diagramms angegeben. Der Datentyp des Rückgabewerts einer Methode steht dabei hinter dem Doppelpunkt.

Neben der Methode "bellen" enthält unsere Klasse vor allem "getter-" und "setter-"Methoden für die Attribute. Diese sind nötig, weil die Attribute "private" deklariert wurden, und somit ein direkter Zugriff auf die Werte der Instanzvariablen nicht möglich ist - das ist aus Sicherheitsgrünen auch gut so. Um dem Hund nun einen neuen Namen zu geben, würde man ein Objekt der Klasse Hund erzeugen und könnte der Instanzvariablen dann mit der "setter"-Methode für den Namen einen neuen Wert zuweisen. Ebenso funktioniert das Auslesen der Werte der Instanzvariablen mit den "getter"-Methoden:

```
myDog = new Hund(5,36,"Emil")
myDog->geName() // "Emil"
myDog->setName("Higgs")
myDog->getName() // "Higgs"
myDog->name // FEHLER
myDog->name = "Higgs" // FEHLER
```

From:
https://www.info-bw.de/ -

Permanent link:
https://www.info-bw.de/faecher:informatik:oberstufe:java:objektorientierung:uml:start?rev=1620294721

Last update: **06.05.2021 09:52**