# Klassen erstellen

Wir haben bereits die ersten Grundbegriffe kennen gelernt und mit Klassen und Objekten gearbeitet. Nun erstellen wir unsere erste eigene Klasse.

## Modellierung eines UML-Klassendiagramms

Bei der objektorientierten Programmierung werden Objekte mit gleichem Aufbau in einer   
(Fach-)Klasse zusammengefasst. Fachklassen können z.B. sein: Rechteck, Kreis (greifbare Dinge), Schüler, Lehrer (Rollen von Personen), Konto (abstrakte Dinge).

In den Fachklassen wird der allgemeine Aufbau der Objekte durch Attribute und Methoden festgelegt.

Um einen Überblick über den Aufbau der verwendeten Klassen zu haben, verwenden wir das sogenannte UML-Klassendiagramm (UML = Unified Modeling Language). In diesem sind die Eigenschaften (Attribute) und das Verhalten (Methoden) gekennzeichnet.

Steht das Modell fest, entscheidet man sich für geeignete Datentypen (Zeichenkette: str, Ganzzahl: int, Dezimalzahl: float, Wahrheitswert: bool) Außerdem schreibt man vor den Attributnamen ein „+“ um zu kennzeichnen, dass die Attribute sichtbar sind (mehr zur Sichtbarkeit in *„L1.6 Sichtbarkeit von Attributen“*). Um das UML-Klassendiagramm zu zeichnen kann ein Programm (z. B. Violet UML Editor) verwendet werden.

**Beispiel Kontoverwaltung (*L1\_3\_Info.py*)**

|  |  |
| --- | --- |
|  |  |
| Fachklasse Konto | Konto-Objekt k  Erstellung im Hauptprogramm |

## Quellcode in Python

Nun erläutern wir den Quellcode in Python, um mit Klassen und Objekten zu arbeiten.

### Aufbau einer Fachklasse

Die Klasse Konto wird als Fachklasse bezeichnet, weil in ihr das eigentliche Fachproblem gelöst werden soll.

In Python wird der gesamte Code, der zu einer Klasse gehört, mit einer Einrückung (Tabulatortaste) gekennzeichnet. Alles, was wir hinter dem #-Zeichen schreiben, ist ein Kommentar und wird vom Interpreter ignoriert.

|  |  |
| --- | --- |
| 1  2  3  4 | class Konto:  # zuerst definieren wir die Attribute  # anschließend die Methoden |

### Festlegen der Attribute mit dem Konstruktor

Anschließend werden über einen sogenannten Konstruktor die Attribute definiert. Um die Datentypen müssen wir uns nicht kümmern. Dies regelt Python selbst. Wir benötigen die Attribute *inhaber*, *kontonummer* und *kontostand*. Ziel ist es, dass der Anwender später im Hauptprogramm mit dem Befehl

k = Konto()

ein eigenes Konto mit bereits festgelegten Werten erstellt.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | class Konto:  # zuerst definieren wir die Attribute  def \_\_init\_\_(self):  self.inhaber = ""  self.kontonummer = 0  self.kontostand = 0  # anschließend die Methoden |

Zeile 3: Attribute werden in Python in einem Konstruktor definiert. Seine Definition startet mit einem *def*, gefolgt von dem Namen dieser speziellen Methode *\_\_init\_\_* (bitte beachten: Hier werden jeweils zwei Unterstriche geschrieben). Der Parameter *self* bedeutet, dass mit diesem Konstruktor das eigene Objekt erstellt wird. Er ist ein besonderer Parameter, da er nur in der Deklaration eines Konstruktors auftaucht (hier: *def \_\_init\_\_(****self****):*), aber beim Aufruf des Konstruktors (hier: *Konto()*) nicht.

Zeile 4: Das eigene Objekt *self* wird aufgerufen. Dem Attribut inhaber wird der Wert „“, also eine leere Zeichenkette zugewiesen.

Zeilen 5-6: Nach dem gleichen Prinzip wie in Zeile 4 wird den Attributen *kontonummer* und *kontostand* der Wert 0 zugewiesen.

### Arbeiten mit den Objekten im Hauptprogramm

Das Hauptprogramm ist dadurch zu erkennen, dass der Quellcode nicht eingerückt ist. Es regelt den grundsätzlichen Programmablauf (von oben nach unten).

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | # Hauptprogramm  k = Konto()  k.inhaber = "Peter Schmidt"  k.kontonummer = 66124457  k.kontostand = 450.00  print("Inhaber:", k.inhaber)  print("Kontonummer:",k.kontonummer)  print("Kontostand:",k.kontostand,"€") |

Zeile 2: Das Objekt k der Klasse Konto wird deklariert und initialisiert. Die Methode Konto() ruft den Konstruktor \_\_self\_\_() des Objektes k auf.

Zeilen 3-5: Den Attributen inhaber, kontonummer und kontostand des Objektes k werden Werte zugewiesen.

Zeilen 7-9: Mit dem print-Befehl werden Ausgaben in der Konsole gemacht.