# Konstruktoren

Wir haben bereits die ersten Grundbegriffe kennen gelernt und einen ersten Konstruktor erstellt. Nun sehen wir uns an, wie Objekte auf unterschiedliche Arten erzeugt werden können.

Ein Konstruktor ist eine spezielle Methode, mit der sich Objekte einer Klasse erzeugen lassen. Die Methode hat bei ihrem Aufruf denselben Namen wie die korrespondierende Klasse und wird im Gegensatz zu allen anderen Methoden groß geschrieben, z.B. Konto().

Wir haben immer zwei Möglichkeiten, ein Objekt zu initialisieren.

1. Entweder wir deklarieren ein Objekt und weisen seinen Attributen später einen Wert zu.

2. Oder wir erstellen einen Konstruktor mit Parameter(n) und legen diese(n) bereits beim Aufruf des Konstruktors fest.

## Der Leerkonstruktor

Die einfachste Art, Objekte zu erzeugen, ist mit dem Leerkonstruktor. Die Attribute werden hier zwar definiert, deren Werte müssen wir dann aber in weiteren Schritten festlegen.

In der Fachklasse lässt sich der Leerkonstruktor als Methode \_\_init\_\_ (man erkennt diese an den zwei Unterstrichen, jeweils vor und nach dem Schlüsselwort *init*) durch den folgenden Code konkret angeben. Der Parameter self stellt einen Verweis auf das aktuelle Objekt dar. Dies ist das gerade aktive Objekt. Hinweis: Der Parameter *self* erscheint nur bei der Definition einer Methode. Beim Aufruf wird er nicht angegeben.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | # Fachklasse  class Konto:  # zuerst definieren wir die Attribute  def \_\_init\_\_(self):  self.inhaber = ""  self.kontonummer = 0  self.kontostand = 0  # Hauptprogramm  k = Konto()  k.inhaber = "Peter Schmidt"  k.kontonummer = 66124457  k.kontostand = 450.00 |
|  | **Datei:** L1\_5a\_Info.py |

Zeile 4: Attribute werden in Python in einem Konstruktor definiert. Dessen Definition startet mit einem einleitenden *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.

Zeile 5: Das eigene Objekt *self* wird aufgerufen. Dem Attribut inhaber wird der Wert „“, also eine leere Zeichenkette zugewiesen.

Zeilen 6-7: Nach dem gleichen Prinzip wie in Zeile 4 wird den Attributen *kontonummer* und *kontostand* der Wert 0 zugewiesen.

Zeilen 10-13: Das Objekt k wird erzeugt. Die Attributwerte können anschließend gesetzt werden.

## Konstruktor, der die Angabe eines Wertes erzwingt

Die Methode \_\_init\_\_ lässt sich in der Fachklasse so manipulieren, dass man gezwungen ist, einen oder mehrere Parameter anzugeben. Indem die Zuweisung eines Standardwertes ausbleibt, muss der Nutzer beim Aufruf des Konstruktors direkt einen eigenen Wert angeben. Hinweis: Der Parameter *self* erscheint nur bei der Definition einer Methode. Beim Aufruf wird er nicht angegeben.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | # Fachklasse  class Konto:  def \_\_init\_\_(self, p\_inhaber, p\_kontonummer, p\_kontostand):  self.inhaber = str(p\_inhaber)  self.kontonummer = int(p\_kontonummer)  self.kontostand = float(p\_kontostand)  # Hauptprogramm  k = Konto("Peter Schmidt", 66124457, 450.00) |
|  | **Datei:** L1\_5b\_Info.py |

## Zeilen 4-6: Bei der Zuweisung der Attributwerte legen wir der Übersicht halber auch den Datentyp fest. Mögliche Datentypen sind: str: Zeichenkette, int: Ganzzahl, float: Dezimalzahl, bool: Wahrheitswert