

Kapitel 11.2: Klassen – Glossar

Klasse

Klassen sind Vorlagen (oder auch Baupläne), nach denen Objekte zur Laufzeit des Programms erzeugt werden. In einer Klasse ist definiert, welche **Attribute (Variablen)** ein Objekt dieser Klasse hat und welche **Methoden (Funktionen)** für das Objekt zur Verfügung stehen. Ein Auto hat z. B.

- die Attribute `marke`, `modell`, `ps-zahl`, `datum_erstzulassung`, ...
- und die Methoden `fahren`, `bremsen`, `kuppeln`, `tuere_oeffnen`, ...

Wichtig ist, dass man eine Klasse nicht mit einem Objekt verwechselt. Wenn ein Auto gebaut wird, dann wird es nach einem bestimmten Bauplan gebaut. Jedes Auto ist aber trotzdem ein eigenständiges Objekt. Es stehen ja nicht alle Autos am gleiche Platz, haben die gleiche Farbe, die gleiche PS-Zahl, usw.

Wir definieren Klassen in einer eigenen Datei, die genauso heißt wie die Klasse. Der **Dateiname wird kleingeschrieben**. Der **Klassenname** hingegen **beginnt mit einem Großbuchstaben** – die Klasse `Auto` würde also in einer Datei namens `auto.py` definiert. Das ist zwar nicht zwingend notwendig, aber grundsätzlich sinnvoll, weshalb man sich das gleich so angewöhnen sollte.

Klassen-Definition

Klassen werden wie folgt definiert:

```
class Kreis:
    # alles, was zur Klasse gehört, wird eingerückt
```

self

Jedes Objekt kennt sich selbst und kann auf seine eigenen Attribute und Methoden zugreifen – über sich selbst: **self**. Jede Methode der Klasse hat als ersten Parameter die Variable `self`. Immer! Diese Variable wird nicht übergeben, sondern von Python automatisch hinzugefügt. Im Aufruf der Methode ist sie nicht zu sehen. Mit `self.x` kann man z. B. auf das Attribut `x` des aktiven Objekts zugreifen. Mit `self.draw()` auf die `draw` Methode des Objekts.

init Methode

Zu jeder Klasse gehört immer eine `__init__` Methode. Diese hat zwei Unterstriche vorne und zwei hinten! Auch diese Methode erhält als ersten Parameter immer `self`. Nach dem Parameter `self` können beliebig viele weitere Parameter übergeben werden – wie in den uns bereits bekannten Funktionen. Die `__init__`-Methode wird immer aufgerufen, wenn ein Objekt dieser Klasse erzeugt wird. Die Parameter, die übergeben werden, dienen dazu, ein gültiges Objekt der Klasse zu erzeugen. Mithilfe dieser Parameter muss es also möglich sein, allen Attributen der Klasse einen sinnvollen Wert zu geben. Hier sollten – und das ist guter Stil – auch wirklich alle Attribute der Klasse einmal mit einem Wert initialisiert werden.

Klassen importieren

Um Objekte einer Klasse im Hauptteil (`main.py`) oder in Objekten einer anderen Klasse nutzen zu können, muss man sie erst importieren. Das geht so ähnlich, wie wir das schon von Modulen kennen. Im Grunde sind auch das wieder nur Klassen. Allerdings geben wir für unsere Klassen explizit an, dass wir diese Klasse aus ihrer Datei importieren möchten – das erleichtert nachher den Aufruf. Es gibt auch noch andere Möglichkeiten, aber wir machen das erstmal so. `from klasse import Klasse`

Objekt

Ein Objekt wird aus den Vorgaben einer Klasse erzeugt. Es kann beliebig viele Objekte einer Klasse geben, die alle unabhängig voneinander verwaltet werden. In unserem Beispiel hat jeder Kreis eine andere Position, Farbe, Geschwindigkeit, Bewegungsradius, usw. Denken wir nochmal an die `type`-Funktion aus der Teil 1 – ist lange her, aber da war doch was! Diese `type`-Funktion funktioniert nicht nur für Integer oder Strings, sondern für alle Objekte. Mit anderen Worten: Auch Integer, Strings, Listen, Booleans usw. sind Objekte. **Python arbeitet immer mit Objekten!**

Ein Objekt erzeugen

Um ein Objekt aus einer Klasse zu erzeugen, muss man zunächst einmal wissen, welche Parameter dafür benötigt werden, also welche Parameter zusätzlich zu `self` noch an die `__init__` Methode übergeben werden müssen.

Beispiel

```
class Kreis:
    def __init__(self, x, y):
        self.x = x
        self.y = y
        self.radius = 20
```

Zum Erzeugen eines Objektes der Klasse `Kreis` wird also die Position des Kreises benötigt. Der Radius hingegen wird in diesem Beispiel immer zunächst auf 20 Pixel gesetzt. Man weist also einer Variablen ein Objekt zu. Die Variable ist dann vom Typ der Klasse des Objekts.

Methode

Eine Methode ist eine Funktion in einer Klasse. Der erste Parameter einer jeden Methode ist `self` – also das Objekt selbst. Dieser Parameter wird beim Aufruf der Methode nicht übergeben, sondern von Python wird das Objekt, auf dem die Methode aufgerufen wird, als `self` automatisch übergeben.

Wenn wir z. B. 2 Kreise erzeugen:

```
k1 = Kreis(15, 27)
k2 = Kreis(0, 0)
```

und dann für `k2` eine Methode aufrufen:

```
k2.move(5)
```

dann wird k2 als Parameter `self` and die move-Methode weitergeben:

```
def move(self, dist)
    self.x += dist
```

Attribut

Die Variablen, die einem Objekt einer Klasse zur Verfügung stehen, werden als Attribute (Eigenschaften) der Klasse bzw. der darauf erzeugten Objekte bezeichnet. Die Attribute einer Klasse werden immer über das Objekt selbst referenziert: `self.x` Auf die Attribute eines Objekts greift grundsätzlich nur das Objekt selbst zu. Andere Objekte oder Funktionen verwenden grundsätzlich nur die Methoden des Objekts.