

OOP - Klassen I

1.1 Was ist eine Klasse?

- Klassen sind eine Art Container um daran Variablen und Methoden zu sammeln.
- Klassen beschreiben ein Objekt, z.B. ein Viereck und dessen Eigenschaften. So ist ein Viereck durch seine Breite, Höhe und Farbe eindeutig definiert.
- Eine Klasse ist nur der Bauplan für ein Objekt. Man muss so genannte Instanzen(Objekte) davon erstellen, die alle Eigenschaften der Klasse bekommen.

1.2 Klassen in Python

- Klassen beinhalten Funktionen. Jede Funktion **muss** als ersten Parameter immer den Parameter `self` haben. Allerdings muss dieser beim Aufruf **nicht** übergeben werden!
- Es gibt eine Spezial-Methode (`__init__`) , den so gennaten Konstruktor. Hier werden alle Variablen definiert, die zur Klasse gehören sollen. Jede solche Variable wird mit `self.variablenName = wert` erstellt.
- Variablen, die zu `self` gehören müssen innerhalb der Klasse immer mit `self.variablenName` verwendet werden. Außerhalb sind diese über die Klassen-Instanz zugreifbar

```
class Rechteck:
    def __init__(self, breite, farbe):
        self.breite = breite
        self.hoehe = hoehe
        self.farbe = farbe

    def flaeche(self):
        return self.breite * self.hoehe

# ruft die __init__ Methode auf
meinRechteck = Rechteck(100, 200, "rot")

# Ruft die Methode flaeche der Instanz auf
flaeche = meinRechteck.flaeche()
print("Die Fläche des Rechtecks ist: ", flaeche)

# Instanzvariable (definiert mit self.variablenName in
  ↳ __init__) abfragen
print("Breite: ", meinRechteck.breite)
```

1.2.1 Klassen-Instanzen

- Man kann beliebig viele Instanzen einer Klasse erstellen
- Um eine Methode/Variable auf einer Instanz aufzurufen muss man immer den ‘Variablennamen’ der Instanz mit Punkt verbunden davor schreiben, z.B. `meinRechteck.flaeche()`. Sonst weiß das Programm nicht zu welcher Instanz diese Methode/Variable gehört und es gibt einen Fehler.

```
# ...

# Zwei verschiedene Instanzen der Klasse Rechteck
r1 = Rechteck(100, 200, "rot")
r2 = Rechteck(100, 100, "blau")

# Methode flaeche() auf den Instanzen aufrufen
flaeche1 = r1.flaeche()
flaeche2 = r2.flaeche()

print("Flächen:", flaeche1, flaeche2)
```

1.2.2 Spezielle Klassenmethoden

- Es gibt verschiedene *Sondermethoden*. Diese beginnen und enden wie schon `__init__` mit zwei `__`.
- Eine nützliche Sondermethode ist `__str__`. Diese Methode dient dazu um Information über dein Objekt als Text auszugeben.
- Die `print(..)`-Funktion ruft intern die `__str__` eines Objektes auf und gibt das Ergebnis aus.

```
class Rechteck:
    # ...

    def __str__(self):
        return "Rechteck: " + str(self.breite) + ", " +
        ↪ str(self.hoehe)

recht = Rechteck(100, 100, "gruen")
print(recht)
```