

## 22.8 Besondere Methoden in Python

- Konstruktor-Methode, die ein Objekt bei der Instanziierung mit Anfangswerten initialisiert  
wird mit dem Instanzieren automatisch aufgerufen `__init__()`
- Methode, die die Eigenschaften (Attribute) eines Objektes ausgibt  
bereitet die Zeichenkette zur Ausgabe der Attribute eines Objektes auf  
z.B. mit `print(...)` ausgeben `__str__()`

- Beispiel:

```
# Definition der Klasse Fahrzeug mit
# der Konstruktor-Methode __init__() und
# der Methode zur Ausgabe der Attribute (Beschreibung des Objektes)
class Fahrzeug:
    def __init__(self, h, m, g):
        self.hersteller = h
        self.modell = m
        self.geschwindigkeit = 0
    def beschleunigen(self, wert):
        self.geschwindigkeit = wert
def ausgeben(self):
    print(f"{self.hersteller} {self.modell}: " \
          f"Geschwindigkeit: {self.geschwindigkeit:3} km/h")
    def __str__(self):
        return f"{self.hersteller} {self.modell}: " \
               f"Geschwindigkeit: {self.geschwindigkeit:3} km/h"

# das Objekt mit dem Namen f1 instanziiieren
# durch die Konstruktor-Methode __init__() werden die Attribute
# initialisiert
f1 = Fahrzeug("VW", "Golf", 0)

# aufrufen der Methode ausgeben
# zur formatierten Ausgabe der aktuellen Geschwindigkeit
f1.ausgeben()
# damit wird die Methode __str__() zur Ausgabe der Attribute
# aufgerufen
print(f1)
# aufrufen der Methode beschleunigen
# zur Veränderung der Geschwindigkeit - neuer Wert = 50
f1.beschleunigen(50)

# aufrufen der Methode ausgeben
# zur formatierten Ausgabe der aktuellen Geschwindigkeit
f1.ausgeben()
print(f1)
```

```
VW Golf: Geschwindigkeit: 0 km/h
VW Golf: Geschwindigkeit: 50 km/h
```

## 22.9 Klassendefinition auslagern

- die Klassendefinition sollten zur besseren Übersicht in eigene Python-Module ausgelagert werden
- für den Modulname kann der Name der Klasse mit der Dateiendung .py verwendet werden
- im Hauptmodul wird diese Klassendefinition in der Form

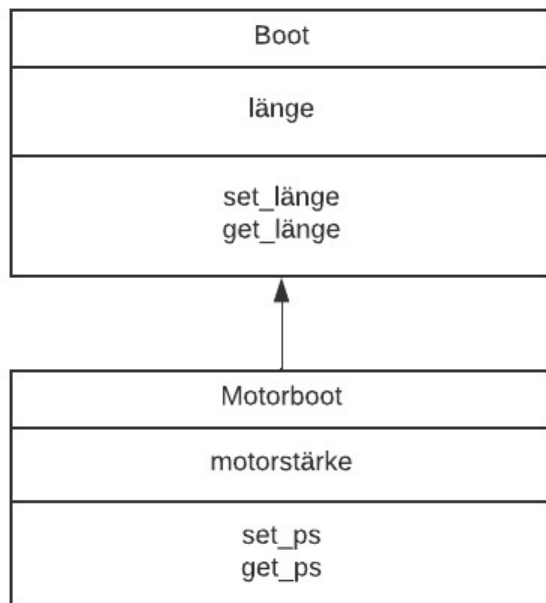
**from** modulename **import** klassenname

eingebunden

- Modulname ist der Name der Pythondatei ohne die Dateiendung .py
- Klassenname ist der Name der Klasse oder \* für alle Klassen in der Pythondatei

## 22.10 Vererbung

- Grundprinzip der OOP:
  - aus einer übergeordneten Klasse (Elternklasse) kann an
  - eine speziellere Klasse (Kindklasse) abgeleitet werden
  - in der abgeleiteten Klasse müssen nur noch Attribute und Eigenschaften definiert werden, in der Elternklasse nicht vorhanden sind oder angepasst werden müssen
- Beispiel:



- die Klasse Boot hat das Attribut *länge* und die Methoden *set\_länge* und *get\_länge*
- die Klasse Motorboot erbt von der Klasse Boot alle Attribute und Methoden und hat zusätzlich die Attribute *motorstärke* und *set\_ps* und *get\_ps*
- Definition der Elternklasse Boot wie gehabt, siehe *Definition eines Objektes (Instanz) in Python*
- Definition der Kindklasse Motorboot mit dem Verweis auf die Elternklasse  
**class kindklasse(elternklasse)**
- in der Kindklasse lassen sich die Attribute und Methoden der Elternklasse ansprechen über **super()......**

- Beispiel:

```
# Definition der Elternklasse Boot mit
#   der Konstruktor-Methode __init__() und
#   den Methode: Konstruktor-Methode, set_laenge, get_laenge und
#   der Methode zur Ausgabe der Attribute (Beschreibung des Objektes)
class Boot:
    def __init__(self, l):
        self.__laenge = 0
        self.set_laenge(l)

    def set_laenge(self, l):
        if l > 0:
            self.__laenge = l

    def get_laenge(self):
        return self.__laenge

    def __str__(self):
        return f"Boot mit {str(self.get_laenge())} m Länge"

# Definition der Kindklasse Motorboot mit Verweis auf die Elternklasse
#   Boot und
#   der Konstruktor-Methode __init__() und
#   den Methode: Konstruktor-Methode, set_ps, get_ps und
#   der Methode zur Ausgabe der Attribute (Beschreibung des Objektes)
#   (mit super(). wird auf die Elternklasse verwiesen)
class Motorboot(Boot):
    def __init__(self, l, p):
        self.__ps = 0
        self.set_ps(p)
        super().__init__(l)

    def set_ps(self, p):
        if p > 0:
            self.__ps = p

    def get_ps(self):
        return self.__ps

    def __str__(self):
        return f"Motorboot mit {str(self.get_laenge())} m Länge " \
            f"und {str(self.get_ps())} PS"

# Objekt Boot erzeugen
b = Boot(10)
# Objekt Motorboot erzeugen
m = Motorboot(15, 120)
# Attribute von Boot ausgeben
print(b)
# Attribute von Motorboot ausgeben
print(m)
```

```
Boot mit 10 m Länge
Motorboot mit 15 m Länge und 120 PS
```

## Aufgabe

1. Die Klasse Konto ist zu erstellen bzw. die Klassendefinition von Aufgabe auf der Seite 6 zu verwenden.
  - Die Klasse hat die Eigenschaften Kontonummer *ktoNr*, Kontoinhaber *inhaber* und den aktuellen Kontostand *ktoStand*.
  - Die Methoden zum Einzahlen *einzahlen*, zum Auszahlen *auszahlen* und zur Ausgabe des aktuellen Kontostandes *ktoStand\_abfragen* sind zu definieren:
    - *einzahlen*                                      Betrag auf das Konto einzahlen  
Kontostand wird erhöht
    - *auszahlen*                                      Betrag vom Konto auszahlen  
Kontostand wird verringert
    - *ktoStand\_abfragen*                              Ausgeben des aktuellen Kontostandes
2. Von dieser Klasse Konto sind die beiden Klassen Girokonto und Sparkonto abzuleiten.
  - Die Klasse Girokonto besitzt zusätzlich die Eigenschaften *sollzinsen* und *habenzinsen*, sowie die Methode *überweisen* mit den Parametern *ziel\_ktoNr* und *betrag*
  - Die Klasse Sparkonto hat die Eigenschaft *zinssatz* und die Methode *ktoStand\_abfragen*  
(Hinweis: diese Methode überschreibt die Methode gleichen Namens der Elternklasse – es soll zusätzlich der Zinssatz ausgegeben werden)
3. Diese Klassendefinitionen sind wie folgt zu testen:
  - Es ist ein Girokonto anzulegen
  - Es ist ein Sparkonto anzulegen
  - Auf das Girokonto sind 500€ einzuzahlen
  - Auf das Sparkonto sind 100€ einzuzahlen
  - Vom Girokonto sind 200€ auszuzahlen
  - Vom Girokonto ist eine Überweisung von 50€ zugunsten des Sparkontos auszuführen
  - Die Kontodaten des Girokontos sind auszugeben
  - Die Kontodaten des Sparkontos sind auszugeben