



Tag 6a - Python OOP

Falls Sie heute lieber noch Aufgaben von den vorherigen Übungszetteln bearbeiten möchten, dürfen Sie das natürlich auch gerne tun.

Aufgabe 1: Objekte erzeugen

Gegeben ist folgende Python-Klasse:

```
class Person:

    # Attribute (Eigenschaften)
    name="noch kein Name"
    age=0

    # Methoden (um Attribute ggf. zu aendern)
    def setName(self,x):
        self.name = x

    def setAge(self,x):
        self.age = x

    def haveBirthday(self):
        self.age = self.age + 1

    def talk(self):
        print("Ich heiße", self.name, "und ich bin", self.age, "Jahre alt")
```

- Speichere zuerst den Code in einer Datei namens `Person.py` führe diese in der interaktiven python-Shell (z.B in IDLE) aus.
- Führe unmittelbar nach Teil (a) folgendes in der python-Shell aus:
 - Erzeuge ein Objekt der Klasse `Person`. Speichere dabei das Objekt in einer Variablen `p`.
 - Setze den Namen des Objekts auf `Alice`.
 - Setze das Alter des Objekts auf 22.
 - Lasse die Person sprechen. Was ist die Ausgabe?
 - Lasse die Person ihren Geburtstag feiern. Überprüfe mit der Methode `talk()`, ob das Alter um eins erhöht wurde.
- Ändere die Datei `Person.py` wie folgt:
 - Erweitere die Klasse um ein Attribut `eyeColor`.
 - Schreibe eine Methode `setEyeColor()`, mit der du die Augenfarbe setzen kannst.
 - Passe die Methode `talk()` so an, dass die Person auch ihre Augenfarbe nennt.
 - Speichere die Datei und öffne sie wieder.
- Erzeuge nun unmittelbar nach (c) eine Person namens Bob. Bob soll 20 Jahre alt sein und blaue Augen haben. Lasse Bob sprechen.

Aufgabe 2: Entwurf einer Klasse

In dieser Aufgabe sollst du nun eine Klasse selbst schreiben.

- (a) Schreibe eine Klasse namens `Bestellung` mit:
1. Attributen: `speise`, `getraenk`
 2. Methoden: `setSpeise()`, `setGetraenk()`, `bestellungVorlesen()`
- Speichere die Klasse in einer Datei namens `Bestellung.py` ab und führe diese in der interaktiven python-Shell aus.
- (b) Führe unmittelbar nach Teil (a) folgendes in der python-Shell aus.
1. Erzeuge ein Objekt der Klasse `Bestellung`. Speichere das Objekt in einer Variablen `B`.
 2. Du möchtest Pizza und Cola bestellen. Verwende die entsprechenden set-Methoden.
 3. Überprüfe mit der Methode `bestellungVorlesen()`, ob deine Bestellung richtig war.
- (c) Um diese Bestellung zu dokumentieren (und um diese morgen wieder auszuführen), schreibe ein Programm, das unter Verwendung von `Bestellung.py` die Bestellung durchführt.
Hinweis: Benutze `import`.

Aufgabe 3: Konstruktor verwenden

Meistens möchten wir beim Erzeugen eines Objekts auch gleich die Werte einiger Attribute festlegen. Dies können wir mit dem Konstruktor `__init__()` erreichen. Beachte die *zwei* Unterstriche vor und nach `init`.

```
class Box:
```

```
    # Konstruktor (zum Initialisieren der Attribute)
    def __init__(self,x,y,z,f):
        self.laenge = x
        self.breite = y
        self.hoehe = z
        self.farbe = f
        print("Du hast gerade eine Box erzeugt.")
```

- (a) Speichere die Klasse in einer Datei namens `Box.py` ab und führe diese in der interaktiven python-Shell aus.
- (b) Erzeuge unmittelbar nach (a) ein Objekt der Klasse `Box`. Das Objekt soll die Länge 3, Breite 2 und Höhe 5 besitzen und rot sein. Verwende die python-Shell.
- (c) Erweitere die Klasse um eine Methode `bestimmeVolumen()`. Diese Methode soll das Volumen ausgeben.
- (d) Erweitere die Klasse um eine Methode `bestimmeFarbe()`. Diese Methode soll die Farbe der Box ausgeben. Speichere die Datei und führe sie aus.
- (e) Erzeuge nun mit der erweiterten Klasse die Box aus (b). Lasse dann das Volumen und die Farbe deiner Box im Terminal ausgeben.

Viel Erfolg!