



Übungszettel 6a - Python OOP

Falls du heute lieber Aufgaben von den vorherigen Übungszetteln bearbeiten möchtest, darfst du das natürlich auch gerne tun.

Aufgabe 1: Objekte erzeugen

Hinweis: Die Datei kann auf der Vorkurshomepage heruntergeladen werden

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` und führe diese in der interaktiven python-Shell (z. B in IDLE) aus. Du kannst den Code abtippen, oder von der Vorkurshomepage herunterladen.
- 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.

Solution:

- (a) 1. idle starten.
2. Editor öffnen: File → New File
3. Code für die Klasse kopieren und in der Datei `Person.py` speichern.
Bei herunterladen der Datei: 1. Datei auf Rechner speichern 2. Datei in Editor öffnen: File → open
3. Laufen lassen: Run → Run Module, oder F5 drücken

- (b) 1. `p=Person()`
2. `p.setName('Alice')`
3. `p.setAge(22)`
4. `p.talk()` ⇒ Ausgabe: Ich heiße Alice und ich bin 22 Jahre alt
5. `p.haveBirthday()`
`p.talk()` ⇒ Ausgabe: Ich heiße Alice und ich bin 23 Jahre alt

- (c) class Person:

```
# Konstruktor (Eigenschaften)
def __init__(self):
self.name = "noch kein Name"
self.age = 0
self.eyeColor = "noch keine Farbe"

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

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

def setEyeColor(self,x):
    self.eyeColor = x

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

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

- (d) `>>> p = Person()`
`>>> p.setName("Bob")`
`>>> p.setAge(20)`
`>>> p.setEyeColor("blau")`
`>>> p.talk()`
Ich heiße Bob und ich bin 20 Jahre alt.
Meine Augfarbe ist blau

Aufgabe 2: Entwurf einer Klasse

In dieser Aufgabe sollst du nun eine Klasse selbst schreiben.

- (a) Schreibe eine Klasse namens `Bestellung` mit:
- Attributen: `speise`, `getraenk`
 - 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`.

Solution:

```
(a) class Bestellung:
    2
    3     def __init__(self):
    4         self.speise = "Noch nichts"
    5         self.getraenk = "Noch nichts"
    6
    7     # Methoden
    8     def setSpeise(self,x):
    9         self.speise = x
   10
   11     def setGetraenk(self,x):
   12         self.getraenk = x
   13
   14     def bestellungVorlesen(self):
   15         print("Sie haben", self.speise, "mit", self.getraenk, "bestellt
        .")

(b) >>> b = Bestellung()
    2 >>> b.setSpeise("Pizza")
    3 >>> b.setGetraenk("Cola")
    4 >>> b.bestellungVorlesen()
    5 Sie haben Pizza mit Cola bestellt.

(c) import bestellung
    2
    3 B = bestellung.Bestellung()
    4
    5 B.setSpeise("Pizza")
    6 B.setGetraenk("Cola")
    7 B.bestellungVorlesen()
```

Aufgabe 3: Konstruktor verwenden

Meistens möchten wir beim Erzeugen eines Objekts auch gleich die Werte einiger Attribute festlegen. Dies können wir mit der Init-Methode `__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.

Solution:

- (a) 1. IDLE ausführen
2. File → New Window
3. Code für die Klasse kopieren und in der Datei `Box.py` speichern.
- (b)

```
>>> meineBox = Box(3,2,5,"rot")
Du hast gerade eine Box erzeugt.
```
- (c)

```
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.")

    def bestimmeVolumen(self):
        print("Mein Volumen betraegt:", self.laenge*self.breite*self.hoehe)

    def bestimmeFarbe(self):
        print("Meine Farbe ist", self.farbe)
```
- (d) siehe (c)
- (e)

```
>>> meineBox = Box(3,2,5,"rot")
Du hast gerade eine Box erzeugt.
>>> meineBox.bestimmeVolumen()
Mein Volumen betraegt: 30
>>> meineBox.bestimmeFarbe()
Meine Farbe ist rot
```

Viel Erfolg!