

## 16. OOP Konzepte (2) Datenkapselung

### 16.1. Datenkapselung

- auf öffentliche (public) Attribute von Instanzen kann von außen zugegriffen werden
- das entspricht nicht dem OOP-Prinzip der Datenkapselung, wonach Attribute vor dem unkontrollierten Zugriff von außen zu schützen sind
- deshalb sollte der Zugriff auf die Attribute immer geeignete Methoden erfolgen
- nur in diesen Methoden kann geprüft werden, ob der Zugriff korrekt ist
- diese Methoden werden Set-Methoden genannt
- das Gegenstück sind die Get-Methoden
- im Vergleich zu den öffentlichen Instanzattributen ist die Verwendung der Get- und Set-Methoden für private Instanzattribute wesentlich aufwendiger
  - daher gibt es die **Properties**
  - Properties verhalten sich im Zugriff so einfach wie öffentliche Attribute
  - sie rufen aber im Hintergrund die entsprechenden Get- und Set-Methoden auf
  - Properties werden so wie das Attribut benannt, allerdings mit Großbuchstaben
- bei der Definition der Get-Methoden muss geprüft werden, ob der zurückgegebene Wert vom Typ *veränderlich* (mutable) oder *unveränderlich* (immutable) ist
- bei veränderlichen Typen kann die Rückgabe dazu führen, dass über diesen Verweis Änderungen an dem Attribut vorgenommen werden, ohne eine Set-Methode zu nutzen
- dies kann verhindert werden, indem die Methode nicht das Attribut selbst, sondern eine Kopie zurückgibt  
`return self._attr.copy()`

- Beispiel:

```
class Person:
    def __init__(self):
        self.__name = "leer"
        self.__telefon = "..."

    def set_name(self, n):
        if n.isalpha() == True:
            self.__name = n

    def get_name(self):
        return self.__name

    def set_telefon(self, t):
        self.__telefon = t

    def get_telefon(self):
        # return self.__telefon
        return self.__telefon.copy()

    NAME = property(get_name, set_name)

p = Person()
#p.set_name("Müller")
#print(p.get_name())
p.NAME = "Müller"
print(p.NAME)

p.set_telefon(["0157", "0175"])
telefonnr = p.get_telefon()
telefonnr.append("176")
print("Verweis ", end=" ")
print(telefonnr)
print("Attribut", end=" ")
print(p.get_telefon())
```

```
Müller
Verweis ['0157', '0175', '176']
Attribut ['0157', '0175'].5 10
```

## Aufgabe

### 1. Remote Control

Die Klasse RemoteControl ist so umzubauen, dass die Get-Methoden eine Kopie der Attribute zurückgibt und nicht das Attribut selbst.

Für die privaten Attribute ist der vereinfachte Zugriff über Properties zu ergänzen.