

# LU08a - Einseitige Beziehungen

Damit ein Objekt A ein anderes Objekt B ansprechen kann, muss es dessen Referenz kennen. Dazu muss die Referenz über den Konstruktor oder eine Methode übergeben werden.

Vom grundsätzlich Verhalten her unterscheiden wir in

- einseitige Beziehungen
- zweiseitige Beziehungen

Bei einer einseitigen Beziehung kennt ein Objekt „A“ ein anderes Objekt „B“ aber nicht umgekehrt.

## Beispiel: Geldbeutel und Besitzer

### Klassendiagramm

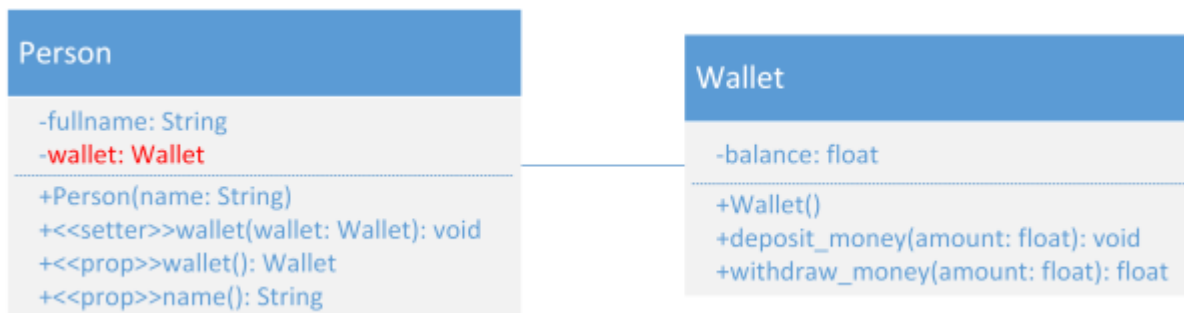


Abb: Klassen in einer einseitigen Beziehung

Eine Person kann auch keinen Geldbeutel besitzen. Daher wird hier die Referenz über eine Methode gesetzt. So sind die Erzeugung des Personen-Objekts und die Zuweisung der Referenz zeitlich unabhängig.

### Sequenzdiagramm

Dies zeigt das zugehörige Sequenz-Diagramm eindeutig auf.

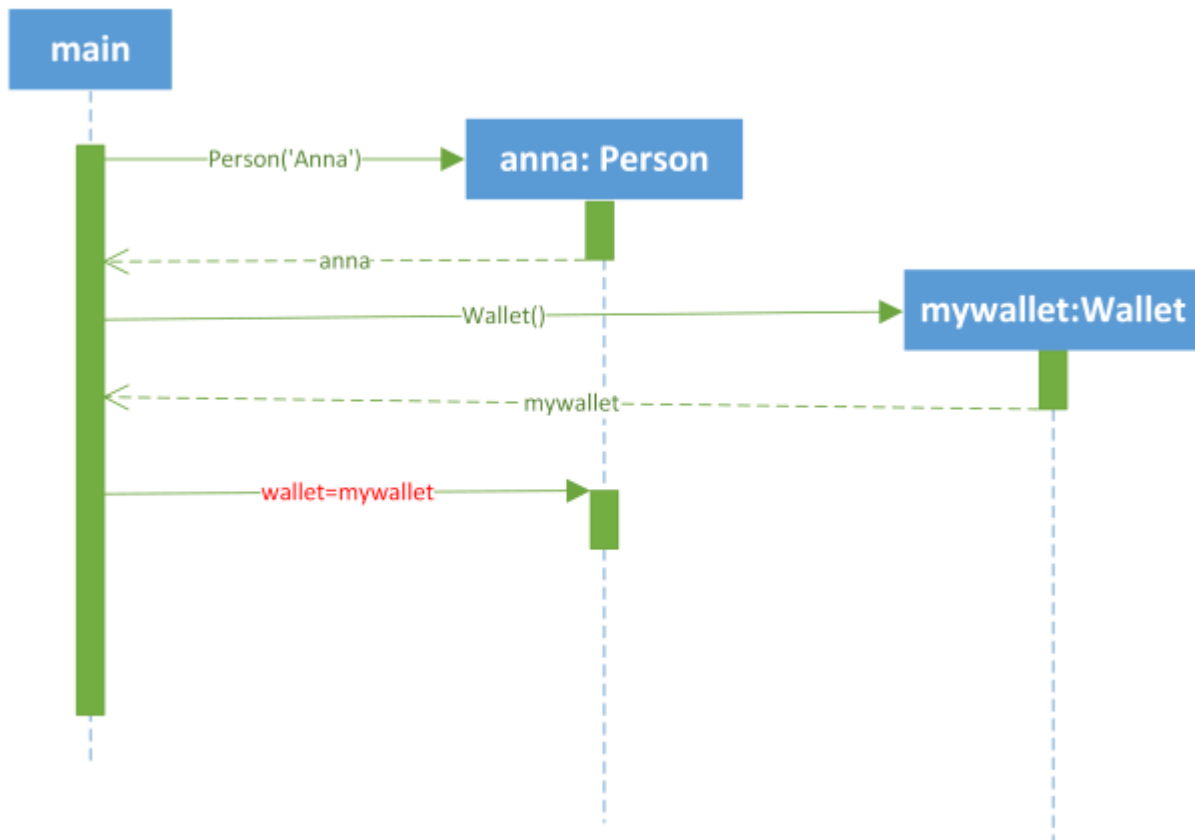


Abb: Sequenz-Diagramm der Zuweisung einer einseitigen Beziehung über eine Methode

## Sourcecode

```
class Person:
    def __init__(self, name):
        self._name = name
        self._wallet = None # die Referenz wird erst später zugewiesen

    @property
    def name(self):
        return self._name

    @property
    def wallet(self):
        return self._wallet

    @wallet.setter
    def wallet(self, wallet):
        self._wallet = wallet

class Wallet:
    def __init__(self):
        pass
```

```

def deposit_money(self, amount):
    pass

def withdraw_money(self, amount):
    pass

if __name__ == '__main__':
    person = Person('Max')
    wallet = Wallet()
    person.wallet = wallet

```

## Beispiel: Stromschalter und Wippe (Aktuator)

### Klassendiagramm

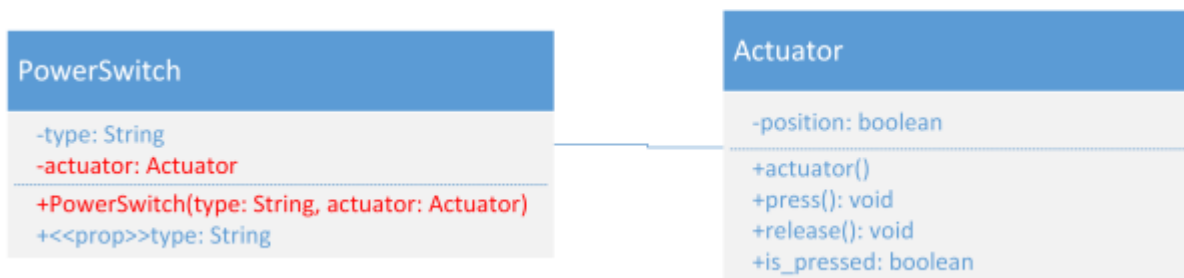


Abb: Klassen in einer einseitigen Beziehung

Ein Stromschalter braucht immer eine Wippe, sonst kann er ja nicht benutzt werden. Daher muss sichergestellt sein, dass mit der Erzeugung des Stromschalter-Objektes auch die Referenz zu einer Wippe gesetzt wird. Es ist also wichtig, dass die Referenz mit dem Konstruktor geliefert wird. In diesem Fall ist es zwingend, dass die Wippe zeitlich zuerst erzeugt wird, so dass ihre Referenz verfügbar ist.

### Sequenzdiagramm

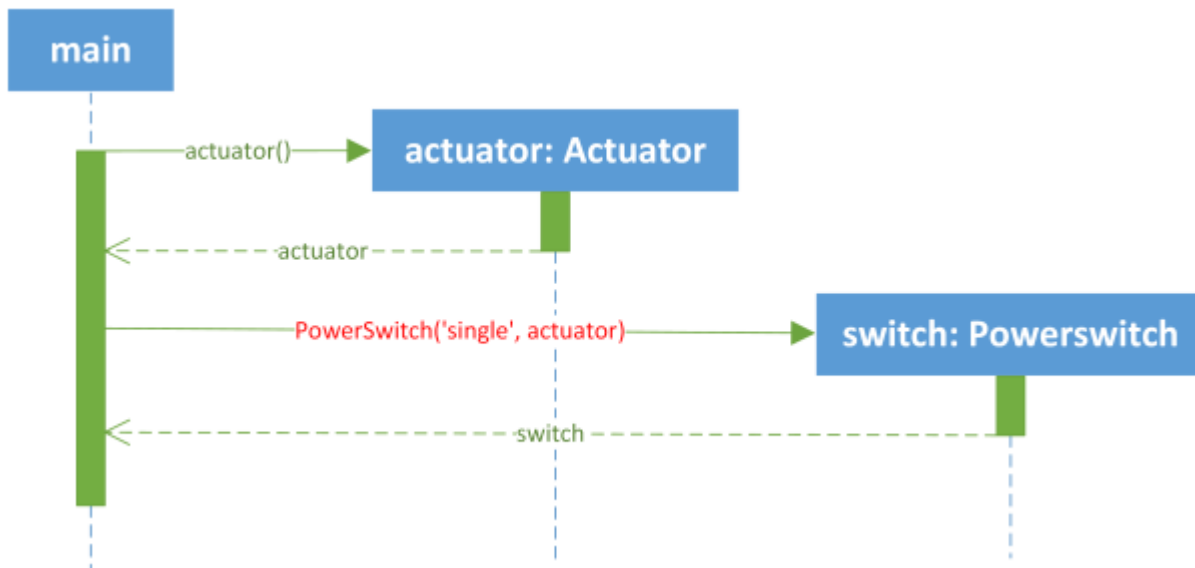


Abb: Sequenz-Diagramm der Zuweisung einer einseitigen Beziehung über den Konstruktor

## Sourcecode

```
class PowerSwitch:

    def __init__(self, type, seesaw):
        self._type = type
        self._actuator = actuator

    @property
    def type(self):
        return self._type

class Actuator:

    def __init__(self):
        self._position = False

    def press(self):
        pass

    def release(self):
        pass

    def is_pressed(self):
        return self._position

if __name__ == '__main__':
    actuator = Actuator()
    switch = PowerSwitch('single', actuator)
```

## M320-LU08



René Probst, bearbeitet durch Marcel Suter

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

[https://wiki.bzz.ch/modul/m320\\_2024/learningunits/lu08/einseitigebeziehung?rev=1726465747](https://wiki.bzz.ch/modul/m320_2024/learningunits/lu08/einseitigebeziehung?rev=1726465747)

Last update: **2024/09/16 07:49**

