## Lösung 1 - einseitige Beziehung

```
class House:
    1.1.1
    Die Klasse gibt einzig den Typ eines Hauses zurück.
    def __init__(self, name):
        Legt den Typ des Hauses fest
        self. type = name
    @property
    def type(self):
        Liefert den Typ des Hauses
        :return: Typ des Hauses
        1.1.1
        return self. type
class HomeOwner:
    Ein Hausbesitzer. Er wird durch seinen Namen benannt und kann den Typ
seines Hauses bekannt geben
    1.1.1
    def __init__(self, name, house):
        Erzeugt ein HomeOwnerObjekt mit einem Namen und der Referenz auf
sein Haus
        :param name: Name des Hausbesitzer
        :param house: Referenz zum Haus
        1.1.1
        self. name = name
        self. my house = house # Der owner erhält die Referenz im
Konstruktor geliefert und weist diese zu.
    def print(self):
        \mathbf{I} \cdot \mathbf{I} \cdot \mathbf{I}
        gibt die Infos zum Hausbesitzer und seinem Haus aus.
          print(f'{self._name} besitzt ein {self._my_house.type}') # Der
owner kan die Referenz nutzen.
    @property
    def name(self):
        Liefert den Namen des Hausbesitzers
        :return: Name des Hausbesitzers
```

 $up a a te: \\ 2024/03/28 \\ modul: m 320: learning units: lu 05: loes ungen: lu 03-aufgabe\_6 \\ https://wiki.bzz.ch/modul/m 320/learning units/lu 05/loes ungen/lu 05/loes ung$ 

```
1.1.1
        return self._name
if __name__ == '__main__':
    house = House('Landhaus')
    owner = HomeOwner('Ron', house) # Dem owner wird die Referenz zum
Haus mitgegeben.
    owner.print()
```



https://wiki.bzz.ch/ - BZZ - Modulwiki

Permanent link:

https://wiki.bzz.ch/modul/m320/learningunits/lu05/loesungen/lu03-aufgabe\_6

Last update: 2024/03/28 14:07

https://wiki.bzz.ch/ Printed on 2025/11/20 02:47