# LU03.A02 - Identität und Gleichheit

Sie können zwei Objekte vergleichen:



- die Gleichheit der bzw. Zugehörigkeit zur Klasse
- 2. die Gleichheit der Objekt-Referenzen (eben die Identität)
- 3. die Gleicheit der Inhalte von Objekten

### Ziel

Sie kennen den Unterschied zwischen Gleichheit (gleiche Klasse) und Identität (gleiche Referenz). Zudem können Sie den Inhalt zweier Objekte vergleichen.

# **Auftrag**

- Ergänzen Sie den Code der Klasse ObjectIdentity gemäss der Erklärung im Quellcode der Methode eq .
- 2. Führen Sie den Unit-Test aus. Er muss fehlerfrei ablaufen. Sollten Fehler auftreten, begrenzen sich diese auf die von Ihnen implementierte Methode \_\_\_eq\_\_\_.
- 3. Ergänzen Sie den Code in main. Beachten Sie insbesondere den letzten Teilauftrag, bei dem es um die Frage des "Inhalts" der Objekte geht. Instanzieren Sie daher hier die beiden Objekte mit den gleichen Zeichenketten.

## Vorgehen

- 1. Akzeptieren Sie das Assignment im GitHub Classroom.
- 2. Klonen Sie ihr Repository in die Entwicklungsumgebung.
- 3. Führen Sie einen Commit und einen Push durch.

#### **Dauer**

30 Minuten + Hausaufgabenzeit

### Wichtig zu Wissen

In dieser Aufgabe werden 3 verschiedene Arten von "Gleichheit" angewendet.

1. die Gleichheit der bzw. Zugehörigkeit zur Klasse

- 2. die Gleichheit der Objekt-Referenzen (eben die Identität)
- 3. die Gleicheit der Inhalte von Objekten

Die ersten beiden Fälle sind durch Keywords eindeutig erkennbar und in allen Fällen durch den Interpreter zu verstehen. Die Klassenzugehörigkeit wird durch isinstance geregelt, während die Identität durch is überprüft wird.

Bei der inhaltlichen Gleichheit wird das ganze etwas komplexer. Bei einfachen Datentypen (int, float, char, boolean) kann der Interpreter die Anweisung

```
value = 7
...
if value == 77:
...
```

umsetzen, da klar ist, um welchen Datentyp es sich handelt.

Das gilt auch für einzelne Zeichen...

```
if sign == 'a':
```

... und den Vergleich von Attributen (bzw. Attributswerte) die einen einfachen DAtentyp repräsentieren.

```
val1 = 99
val2 = 33
...
if val1 > val2:
...
```

Bei einer komplexen Klasse mit verschiedenen Attributen kann der Interpreter den ==-Befehl aber nicht mehr ausführen. Hier braucht es eine (implizite) Methode, die den Vergleich korrekt bewerkstelligt.

Jede Klasse verfügt über eine (verborgene) Methode \_\_\_eq\_\_ (für equal). Diese muss im Sinne des Vergleichs der Attribute entsprechend programmiert werden. Die Methode muss dabei zwei Aspekte abdecken:

Es muss

- 1. die (inhaltliche) Gleichheit zweier Objekte sowie
- 2. ein Vergleich mit einem konstanten Wert

sichergestellt sein.

Ein Beispiel dazu.

Eine Klasse Person weist die Attribute Name und Vorname auf. Es wird festgelegt, dass bei einem Vergleich mit == auch eine Konstante der Art "Vorname Nachname" als gleich erkannt werden soll. Der Einfachheit halber deklarieren wir hier eine Datenklasse.

https://wiki.bzz.ch/ Printed on 2025/12/18 01:38

```
from dataclasses import dataclass
from dataclasses import dataclass
@dataclass
class Person:
   # die Attribute der Datenklasse
   first name: str = None
   last name: str = None
   # die Methode die den ==-Befehl korrekt umsetzt.
   def eq (self, other object):
       # zuerst prüfen, ob das Objekt überhaupt "passt".
       if isinstance(other_object, Person):
            return self.first_name == other_object.first_name and
self.last name == other object.first name
       # danach prüfen, ob das evtl. ein konstanter Wert vorliegt, der zu
prüfen ist.
        elif isinstance(other object, str):
           return self.first_name + ' ' + self.last_name == other_object
        else:
           return False
if __name__ == '__main__':
    carla = Person('Carla', 'Meier')
   theo = Person('Theo', 'Meier')
   if carla == theo:
     print(f'Objekt carla und theo haben den gleichen Inhalt')
   else:
     print(f'Objekt carla und theo haben unterschiedlichen Inhalt')
   if carla == "Carla Meier":
      print(f'Objekt carla weist den Inhalt "Carla Meier" auf')
```

⇒ GitHub Repo für externe Besucher

GitHub Repository https://github.com/templates-python/m319-lu10-a02-reader-module

Lernende am BZZ müssen den Link zum GitHub Classroom Assignment verwenden

#### M320-LU03



René Probst, bearbeitet durch Marcel Suter

Last

update: 2024/08/13 modul:m320\_2024:learningunits:lu03:aufgaben:gleichheit https://wiki.bzz.ch/modul/m320\_2024/learningunits/lu03/aufgaben/gleichheit?rev=1723527539 07:38

From:

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

Permanent link:

https://wiki.bzz.ch/modul/m320\_2024/learningunits/lu03/aufgaben/gleichheit?rev=1723527539

Last update: 2024/08/13 07:38



https://wiki.bzz.ch/ Printed on 2025/12/18 01:38