

LU03b - Gleichheit und Identität

In der objektorientierten Programmierung muss klar unterschieden werden, ob zwei Objekte

- gleichartig,
- gleich oder
- identisch

sind.

Gleichartig

Objekte werden dann als gleichartig betrachtet, wenn sie vom gleichen Typ (der gleichen Klasse) sind.

Die Gleichartigkeit wird mit der Funktion `isinstance` überprüft. Unter der Annahme, dass eine Klasse `Point` existiert, wird folgender Code ausgeführt:

```
if __name__ == "main":
    obj1 = Point(10,10)
    obj2 = Point(5,5)
    obj3 = obj1
    #
    if isinstance(obj1, Point):
        # hier kann nun Code folgen, der das Objekt nutzt
    else:
        # hier könnte eine Fehlermeldung folgen
```

Da `obj1` vom Typ `Point` ist, wird der Code nach dem `if`-Befehl ausgeführt. In diesem Beispiel liefert `isinstance` für alle 3 Objekte (`obj1`, `obj2` und `obj3`) den Wert `True`.

Gleichheit

Zwei Objekte sind gleich, wenn sie

- vom gleichen Typ (Klasse) **und**
- alle Attribute die gleichen Werte enthalten.

`__eq__`

Um zwei Objekte auf Gleichheit zu prüfen, muss die Klasse die Methode `def eq(self, other):` enthalten. In dieser Methode codieren wir die Logik um die Gleichheit zweier Objekte zu prüfen. Hat die Klasse keine Methode `eq`, so verhält sich der Vergleich mittels `==` wie der Vergleich mittels `is` (siehe unten).

```
class Point:
    def __init__(self, xposition, yposition):
        self._xposition = xposition
        self._yposition = yposition

    def __eq__(self, other):
        if not isinstance(other, Point): # don't attempt to compare against
unrelated types
            return NotImplemented
        if self._xposition == other.xposition and self._yposition ==
other.yposition:
            return true
        return false

if __name__ == "main":
    obj1 = Point(10,10)
    obj2 = Point(5,5)
    obj3 = Point(10,10)
    if obj1 == obj2:
        # bei Identität von obj1 und obj2 wird dieser Code ausgeführt
    else:
        print('obj1 und obj2 sind nicht gleich')
# - -
    if obj1 == obj3:
        print('obj1 und obj3 sind gleich')
    else:
        # sind obj1 und obj3 nicht gleich, wird dieser Code ausgeführt
```

In diesem Beispiel gelten obj1 und obj3 als gleich, da alle Attributswerte identisch sind.

Identität

Objektreferenzen werden dann als identisch betrachtet, wenn sie auf ein und das selbe Objekt verweisen. Die Identität einer Referenz wird über den Operator `is` ermittelt.

```
if __name__ == "main":
    obj1 = Point(10,10)
    obj2 = Point(5,5)
    obj3 = obj1
    if obj1 is obj2:
        # bei Identität von obj1 und obj2 wird dieser Code ausgeführt
    else:
        print('obj1 und obj2 sind nicht identisch')
# - -
    if obj1 is obj3:
        print('obj1 und obj3 sind identisch')
    else:
        # sind obj1 und obj3 nicht identisch, wird dieser Code ausgeführt
```

Der erste Vergleich liefert False, da die Referenzen obj1 und obj2 auf unterschiedliche Objekte zeigen. Beim zweiten Vergleich wird True geliefert, da obj1 und obj3 auf ein und das selbe Objekt zeigen.

M320-LU03

 René Probst, bearbeitet durch Marcel Suter

From:

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

Permanent link:

https://wiki.bzz.ch/de/modul/m320_2024/learningunits/lu03/identitaet

Last update: **2026/08/30 14:05**

