2025/11/20 09:27 1/2 LU09a - Mehrfachbeziehung

## LU09a - Mehrfachbeziehung

Aus der realen Welt sind Mehrfachbeziehungen bestens bekannt. So weist eine Schulklasse viele Lernende auf, eine Feriendestination kann von vielen Personen gebucht werden und umgekehrt kann eine Person viele Destinationen besuchen.

Diese beiden Fälle können wie folgt in UML dargestellt werden.

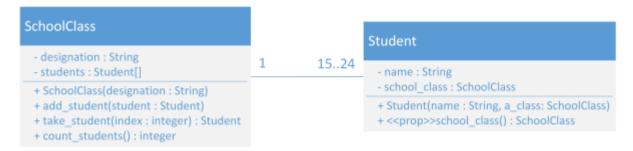


Abb: 1:n Beziehung

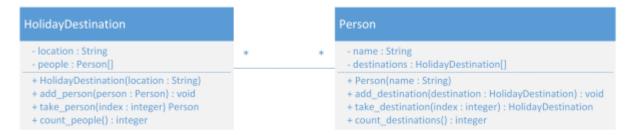


Abb: n:n Beziehung

## Kardinalität

Die Kardinalität gibt an, wie mengenmässig eine Beziehung aussieht. Vereinfacht kann man von

- 1:1 Beziehung (ein Objekt kennt genau ein anderes Objekt)
- 1:n Beziehung (ein Objekt kenntviele andere Objekte)
- n:n Beziehung (ein Objekt kennt viele andere Objekte und umgekehrt)

sprechen. Dabei steht **n** für den Begriff 'viele'. Dies wird in der UML mit einem Stern (\*) wiedergegeben. Der Stern steht für **0 bis unendliche viele** Beziehungen. In der UML ist es aber auch möglich, die Kardinalität ganz genau zu spezifizieren. So ist in der Abbildung 5.9 klar erkennbar, dass eine Schulklasse mindesten 15 aber maximal 24 Studenten haben kann.

## **Umsetzung in Python**

Kennt ein Objekt viele andere, gleichartige Objekte, werden die Referenzen in einer Liste festgehalten. In der Regel wird die Methode für das **Zufügen** als **add**-Methode bezeichnet. Gegenüber dem Begriff **set** signalisiert **add**, dass eben mehrere Werte gesetzt werden können. Für die oben dargestellte Klasse SchoolClass würde das dann wie folgt aussehen (unter Einhaltung der maximalen Grösse der Klasse)

 $update: \\ 2024/09/18 \\ modul: m320\_2024: learning units: lu09: mehr fach beziehung https://wiki.bzz.ch/modul/m320\_2024/learning units/lu09/mehr fach beziehung? rev=1726660970$ 

```
class SchoolClass:
   def init__(self, designation):
       self. designation = designation
       self. students = [] # hier wird eine leere Liste bereitgestellt, die
dann die Refeenzen der Lernenden hält.
   def add student(a student):
       if len(self. students) < 24:</pre>
          self. students.append(a student)
```

Weitere wichtige Methoden dienen

- dem **Abfragen** der Grösse der Liste (Anzahl der gespeicherten Objekte)
- dem Abrufen einer Referenz über einen Index. Dieser muss aber gegen die size der Liste geprüft werden, damit kein unerlaubter Zugriff erfolgt!
- dem Entfernen einer Referenz aus der Liste

```
@property
def count students(self):
   return len(self._students)
def take student(self, index):
  if index < len(self. students):</pre>
     return self. students[index]
  else:
     raise StudentIndexError('search')
def remove student(self, index):
  if index < len(self. students):</pre>
     self._students.remove(index)
  else:
    raise StudentIndexError('remove')
```

## M320-LU09



BY NO 56 René Probst, bearbeitet durch Marcel Suter

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

Permanent link:

https://wiki.bzz.ch/modul/m320 2024/learningunits/lu09/mehrfachbeziehung?rev=172666097

Last update: 2024/09/18 14:02



https://wiki.bzz.ch/ Printed on 2025/11/20 09:27