

LU12a - Eigene Datentypen

Einführung

Problematik

Sie kennen bereits einzelne Variablen und Listen um Daten in einem Programm zu speichern. Wenn ihr Programm mit vielen Daten arbeiten, kann dies aber umständlich werden. Angenommen Sie möchten die Mitglieder eines Clubs verwalten. Für jedes Clubmitglied speichern Sie diese Angaben (**Attribute**):

- Vorname
- Nachname
- Adresse
- Ort
- Postleitzahl
- Eintrittsjahr
- Geburtsjahr
- Ehrenmitgliedschaft (Ja/Nein)

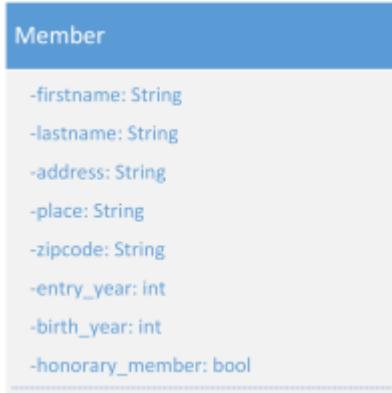
Um diese Daten mit Hilfe von Listen zu verarbeiten, müssten Sie für jedes Attribut eine Liste erstellen.

```
def main():
    firstname_list = []
    lastname_list = []
    address_list = []
    ...
```

Ausserdem müssen Sie sicherstellen, dass die Angaben eines Mitglieds in jeder dieser Listen an der genau gleichen Position steht.

Klassen als Lösung

In vielen modernen Programmiersprachen können wir einen eigenen Datentyp definieren, der aus mehreren bekannten Datentypen besteht. Der Fachbegriff für einen solchen Datentyp ist **Klasse**. Indem wir eine Klasse Member definieren, können wir alle Attribute eines Clubmitglieds zusammenfassen.



Diese Grafik zeigt ein UML Klassendiagramm mit den Attributen und deren Datentyp.

[m319-C1F](#), [m319-C1E](#)



Marcel Suter

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

Permanent link:
<https://wiki.bzz.ch/de/modul/m319/learningunits/lu12/datentypen>

Last update: **2025/06/23 19:31**

