

Informatik Q1 Abels



Beziehungen von Klassen im UML

UML

Assoziation



Aggregation



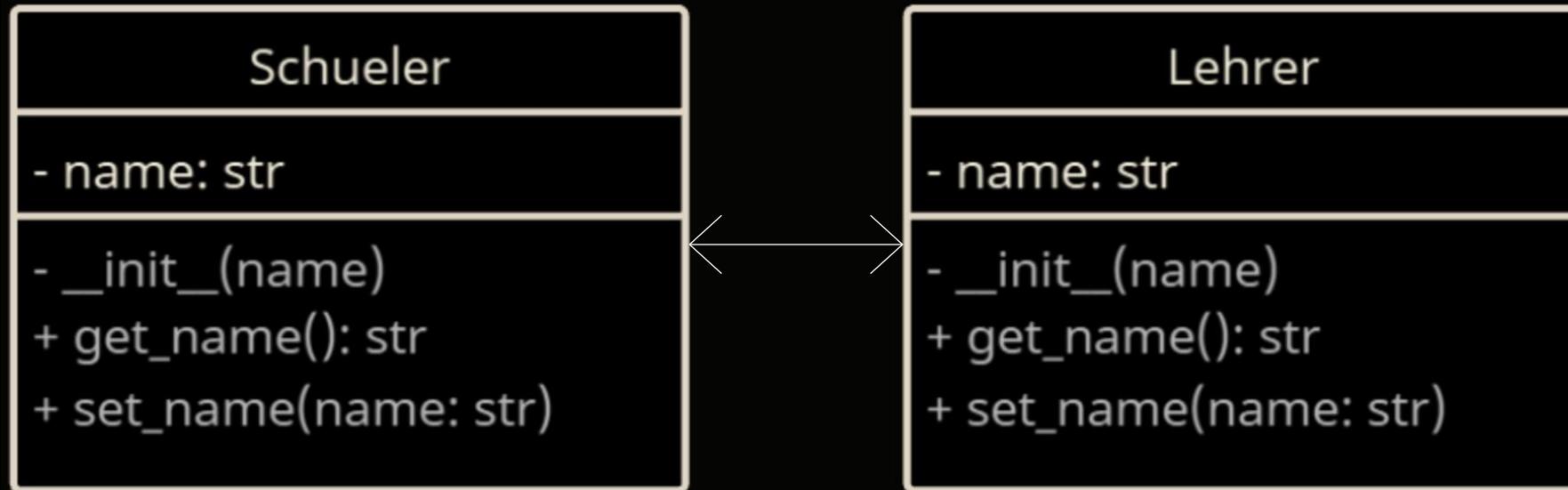
Komposition



Assoziation Aggregation Komposition

“kennt” – Beziehung

Definition: Eine Assoziation beschreibt die Beziehung zwischen zwei Klassen, bei der jedes Objekt einer Klasse mit Objekten einer anderen Klasse in Verbindung steht.

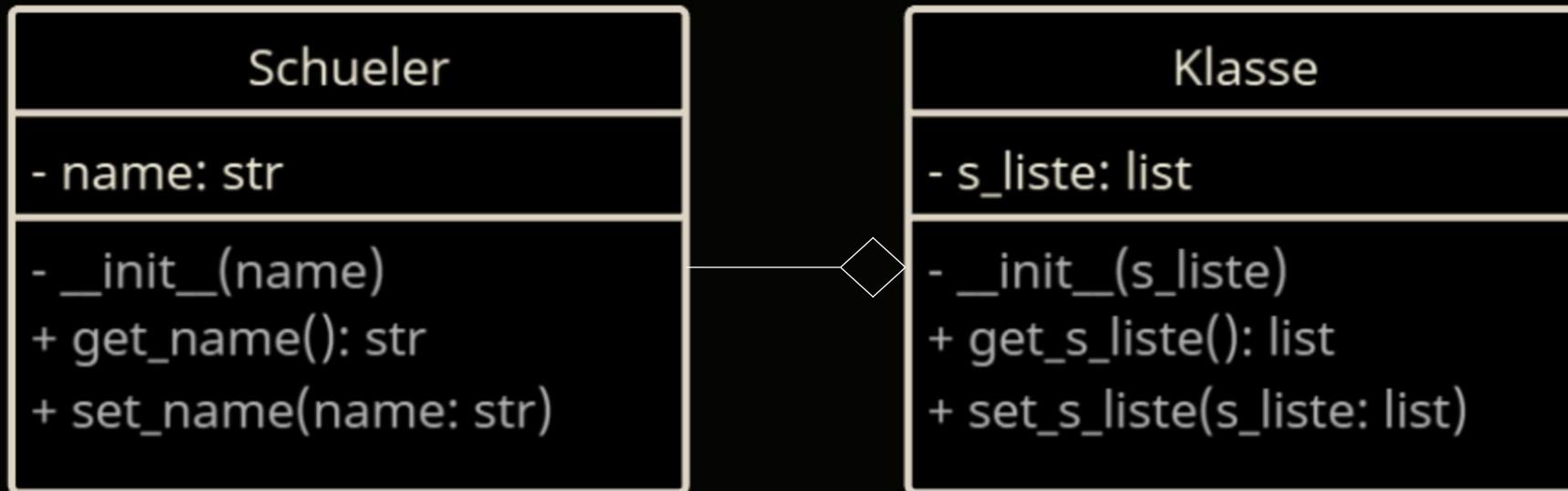


Beispiel: Die Klassen **Schueler** und **Lehrer** können in einer Assoziation stehen, da Schüler Beziehungen zu Lehrern haben. Es gibt jedoch keine direkte Abhängigkeit, und beide Klassen **können unabhängig voneinander existieren**.

Assoziation Aggregation Komposition

“ist-Teil-von” – Beziehung

Definition: Eine Aggregation ist eine spezielle Form der Assoziation, bei der eine Klasse (Container) eine Sammlung von Objekten einer anderen Klasse besitzt, und diese Objekte **können unabhängig voneinander existieren**.

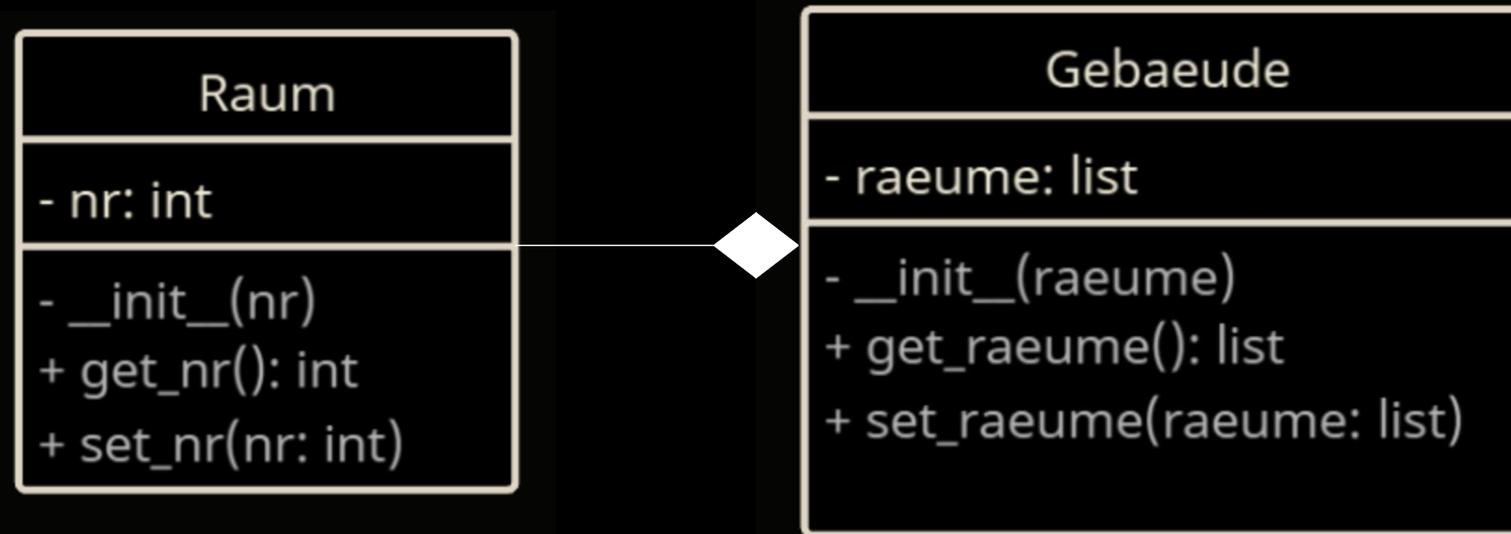


Beispiel: Die Klasse **Klasse** kann eine Aggregation mit der Klasse **Schueler** haben, da eine Klasse eine Sammlung von Schülern enthält. Die Schüler können jedoch auch ohne die Klasse existieren.

Assoziation Aggregation **Komposition**

“ist-Teil-von” – Beziehung

Definition: Eine Komposition ist eine strenge Form der Aggregation, bei der die Lebensdauer der enthaltenen Objekte von der Lebensdauer des übergeordneten Objekts abhängt. **Wenn das übergeordnete Objekt zerstört wird, werden auch die enthaltenen Objekte zerstört.**



Beispiel: Die Klasse **Raum** kann eine Komposition mit der Klasse **Gebaeude** haben, da der Raum nur im Kontext des Gebäudes existiert. Wird das Schulgebäude zerstört, endet auch die Existenz des Raums.



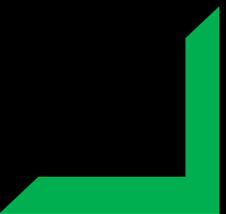
Übung 1

Wir betrachten einen **Buchladen**.

Die Klasse **Autor** wird durch die Attribute **name**, **synonym** und **gage** gekennzeichnet. Ein Autor wird mit seinem Namen erzeugt, Alle Attribute können gelesen werden, Synonym und Gage können auch gesetzt werden. Die Klasse Autor besitzt eine Methode **stelle_vor()**.

Die Klasse **Buch** besitzt die Attribute **autor: Autor**, **titel**, **isbn** und **preis**. Ein Buch wird mit seinem Autor, der ISBN und seinem Titel erzeugt. Alle Attribute sollen gelesen werden können, nur der Preis soll geändert werden können. Die Klasse besitzt eine Methode **stelle_vor()**.

- a) Modelliere ein UML-Klassendiagramm mit beiden Klassen und deren Beziehung.
- b) Implementiere das Programm.
- c) Teste das Programm, indem du zwei Autoren erzeugst, die jeweils drei Bücher geschrieben haben. Zuerst sollen beide Autoren und dann alle Bücher vorgestellt werden.





Übung 1



```
Buchladen.py
1 class Autor:
2     def __init__(self, name):
3         self._name: str = name
4
5     def get_name(self) -> str:
6         return self._name
7
8     def get_synonym(self) -> str:
9         return self._synonym
10
11    def get_gage(self) -> float:
12        return self._gage
13
14    def set_synonym(self, synonym: str):
15        self._synonym = synonym
16
17    def set_gage(self, gage: float):
18        self._gage = gage
19
20    def stelle_vor(self):
21        print(f"Hi, ich bin {self._name}. Manche nennen mich auch '{self._synonym}'. Ich nehme {self._gage:.2f} € pro Buch.")
22
23 class Buch:
24    def __init__(self, autor, titel, isbn):
25        self._autor: Autor = autor
26        self._titel: str = titel
27        self._isbn: str = isbn
28
29    def get_autor(self) -> Autor:
30        return self._autor
31
32    def get_titel(self) -> str:
33        return self._titel
34
35    def get_isbn(self) -> str:
36        return self._isbn
37
38    def get_preis(self) -> float:
39        return self._preis
40
41    def set_preis(self, preis: float):
42        self._preis = preis
43
44    def stelle_vor(self):
45        print(f'''
46            {20*" "}
47            Autor: {self._autor._name} ({self._autor._synonym})
48            Titel: {self._titel}
49            ISBN: {self._isbn}
50            Preis: {self._preis:.2f} €
51            {20*" "}
52            ''')
53
```

```
54 # Autor 1: J. K. Rowling
55 autor_1 = Autor("J. K. Rowling")
56 autor_1.set_synonym("Robert Galbraith")
57 autor_1.set_gage(2000000.0)
58
59 buch_1 = Buch(autor_1, "Harry Potter und der Stein der Weisen", "978-3-551-55681-7")
60 buch_1.set_preis(20.00)
61
62 buch_2 = Buch(autor_1, "Harry Potter und die Kammer des Schreckens", "978-3-551-55682-4")
63 buch_2.set_preis(22.00)
64
65 buch_3 = Buch(autor_1, "Harry Potter und der Gefangene von Askaban", "978-3-551-55683-1")
66 buch_3.set_preis(24.00)
67
68
69 # Autor 2: George R. R. Martin
70 autor_2 = Autor("George R. R. Martin")
71 autor_2.set_synonym("GRRM")
72 autor_2.set_gage(2500000.0)
73
74 buch_4 = Buch(autor_2, "A Game of Thrones", "978-0-553-89784-5")
75 buch_4.set_preis(18.50)
76
77 buch_5 = Buch(autor_2, "A Clash of Kings", "978-0-553-80150-7")
78 buch_5.set_preis(19.50)
79
80 buch_6 = Buch(autor_2, "A Storm of Swords", "978-0-553-10663-3")
81 buch_6.set_preis(21.00)
82
83 # Autoren vorstellen
84 autor_1.stelle_vor()
85 autor_2.stelle_vor()
86
87 # Bücher vorstellen
88 buch_1.stelle_vor()
89 buch_2.stelle_vor()
90 buch_3.stelle_vor()
91 buch_4.stelle_vor()
92 buch_5.stelle_vor()
93 buch_6.stelle_vor()
94
```



Übung 1



```
Terminal

Hi, ich bin J. K. Rowling. Manche nennen mich auch 'Robert Galbraith'. Ich nehme 2000000.00 € pro Buch.
Hi, ich bin George R. R. Martin. Manche nennen mich auch 'GRRM'. Ich nehme 2500000.00 € pro Buch.

-----

Autor: J. K. Rowling (Robert Galbraith)
Titel: Harry Potter und der Stein der Weisen
ISBN: 978-3-551-55681-7
Preis: 20.00 €

-----

Autor: J. K. Rowling (Robert Galbraith)
Titel: Harry Potter und die Kammer des Schreckens
ISBN: 978-3-551-55682-4
Preis: 22.00 €

-----

Autor: J. K. Rowling (Robert Galbraith)
Titel: Harry Potter und der Gefangene von Askaban
ISBN: 978-3-551-55683-1
Preis: 24.00 €

-----

Autor: George R. R. Martin (GRRM)
Titel: A Game of Thrones
ISBN: 978-0-553-89784-5
Preis: 18.50 €

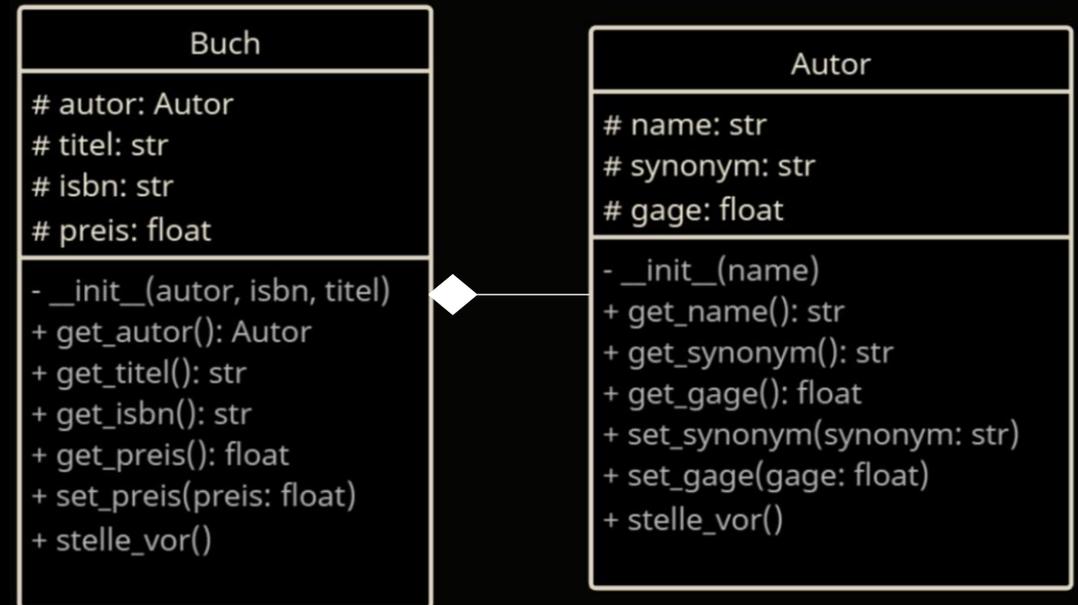
-----

Autor: George R. R. Martin (GRRM)
Titel: A Clash of Kings
ISBN: 978-0-553-80150-7
Preis: 19.50 €

-----

Autor: George R. R. Martin (GRRM)
Titel: A Storm of Swords
ISBN: 978-0-553-10663-3
Preis: 21.00 €

-----
```





Übung 2

Modelliere die folgenden Situationen als UML Klassendiagramm inkl. Beziehungen.

- a) Zwischen einem Musiker und einem Musik-Label besteht in der Regel eine Verbindung, die als „Vertrag“ bezeichnet wird. Musiker können bei einem oder auch keinem Label unter Vertrag stehen.
- b) Ein Zug besteht aus genau 4 Waggons und einer Lok, wobei Waggons und Loks auch ohne Zug existieren können.
- c) In einem Buch gibt es Seiten, Absätze und Wörter. Die Absätze können dabei über mehrere Seiten gehen. Auch ein allein stehendes Wort bildet schon einen Absatz. Jede Seite kann herausgerissen werden. Wenn das Buch verbrennt, verbrennen auch alle seine Seiten.
- d) Eine Bank hat viele Kunden. Jeder Kunde besitzt einen Namen und kann über mehrere Konten verfügen. Zu jedem Konto gehören eine Kontonummer, der Kontostand und die vielen, mit dem Konto verbundenen Einzahlungen und Auszahlungen. Ein- und Auszahlungen bestehen jeweils aus einem Betrag und einem Datum.





Tagebucheintrag

Assoziation

Aggregation

Komposition



Wochenübung

- a) Erstelle ein UML-Klassendiagramm zum Thema **Fitnessstudio** mit den Klassen **Mitglied**, **Trainer**, **Kurs**, wobei ein Trainer selbst auch Mitglied ist. Dabei wird ein Kurs von einem Trainer geleitet und von mehreren Mitgliedern bestritten.
- b) Implementiere entsprechende Klassen und teste sie.

