

Informatik Q1 Abels



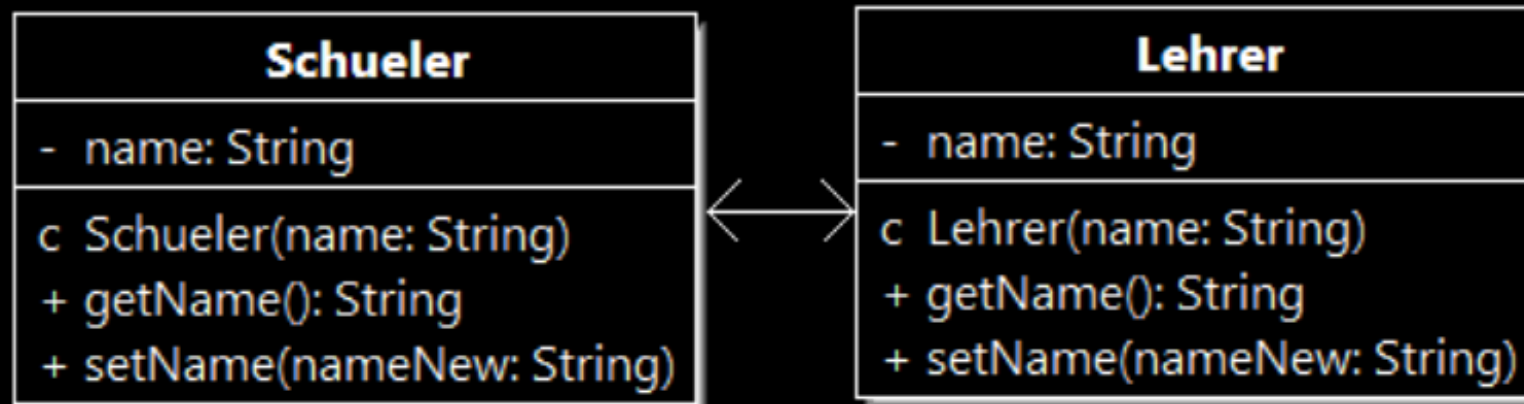
Beziehungen von Klassen im UML

- Assoziation
- Aggregation
- Komposition

Assoziation Aggregation Komposition

“kennt”-Beziehung

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

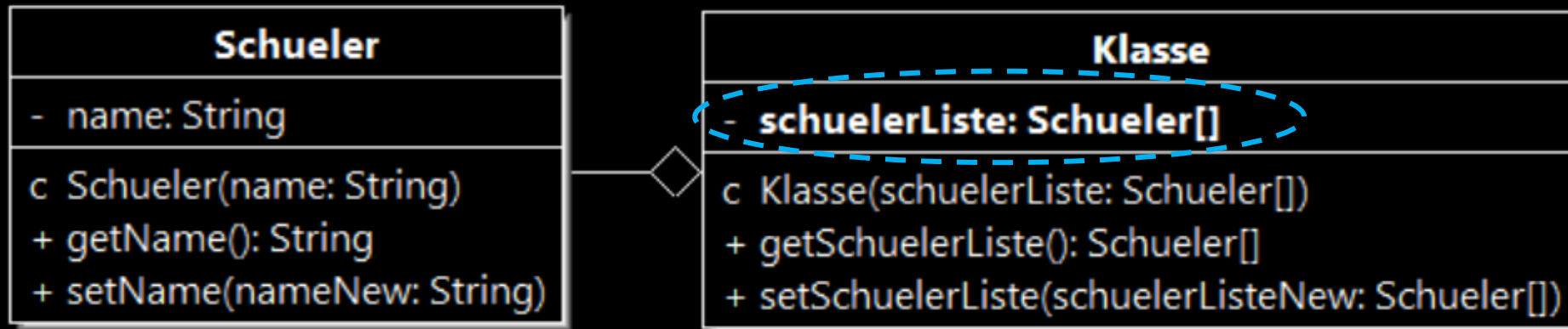


Beispiel: Die Klassen “Schüler” und “Lehrer” können in einer Assoziation stehen, da Schüler Beziehungen zu Lehrern haben. Es gibt jedoch keine direkte Abhängigkeit, und beide Klassen können unabhängig voneinander existieren.

Assoziation Aggregation Komposition

“ist-Teil-von”-Beziehung

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**.

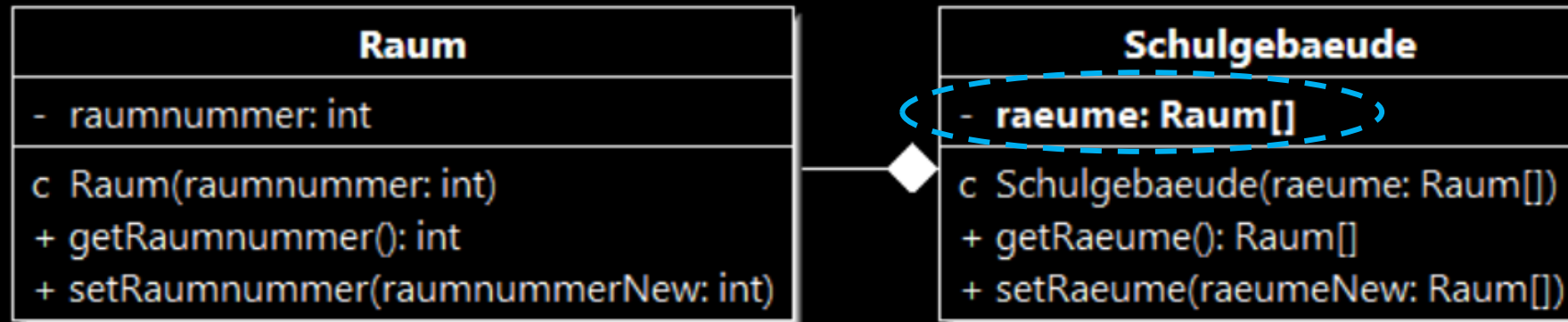


Beispiel: Die Klasse "**Klasse**" kann eine Aggregation mit der Klasse "**Schueler**" haben, da eine Klasse eine Sammlung von Schülern enthält. Die Schüler können jedoch auch ohne die Klasse existieren.

Assoziation Aggregation **Komposition**

“ist-Teil-von”-Beziehung

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 „**Schulgebaeude**“ haben, da der Raum nur im Kontext des Schulgebäudes existiert. Wird das Schulgebäude zerstört, endet auch die Existenz des Raums.



Übung 1

Wir betrachten einen **Buchladen**.

Die Klasse **Buch** besitzt die Attribute **Titel**, **ISBN** und **Preis**. Ein Buch wird mit seiner ISBN und seinem Titel erzeugt. Alle Attribute sollen gelesen werden können, nur der Preis soll geändert werden können. Die Klasse besitzt eine Methode **toString**, welche die Buchdaten zurückgibt.

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 **toString**, welche die Daten des Autors zurückgibt.

Zu jedem **Buch** soll ein **Autor** gespeichert werden.

- a) Modelliere ein UML-Klassendiagramm der Klassen **Buch** und **Autor** mit deren Beziehung.
- b) Beschreibe, wie der **Autor** in der Klasse **Buch** durch Attribute und Methoden eingebunden wird.
- c) Implementiere die Klassen **Buch** und **Autor** in Java.
- d) Implementiere eine **Main**, welches drei Bücher mit ihren Autoren erzeugt und über die Methode **toString** die Daten ausgibt.



Tagebucheintrag



Assoziation

Aggregation

Komposition



Wochenübung

- Erstelle ein UML-Klassendiagramm zum Thema **Fitnessstudio**.
- Implementiere entsprechende Klassen sowie eine Console in Java.

