

Klassenbeziehungen (mit Material von Siegfried Spolwig)

Wir haben gesehen, dass Objekte Eigenschaften haben und ein Verhalten, welches durch ihre Methoden ausgedrückt ist. In der Realität gibt es natürlich auch **Beziehungen** zwischen Objekten, die im Softwaresystem entsprechend abgebildet werden sollen.

Eine prinzipielle OO-Sichtweise beschreibt das Verhältnis zwischen zwei aktiven Objekten. Wenn ein Objekt irgendeine Aktion nicht aus dem eigenen Verhaltensrepertoire erfüllen kann, dann ruft es die Methode eines anderen Objekts auf und fordert beispielsweise eine Leistung an. Die beiden Objekte kommunizieren über ihre Methoden miteinander. Dazu müssen sich beide Objekte kennen und eine Beziehung zueinander haben. Es gibt drei wichtige Beziehungsmuster in der OO-Modellierung, die eine Beziehungssituation abbilden:

Vererbung, Aggregation und Assoziation.

1. Referenz- oder Nutzungsbeziehungen

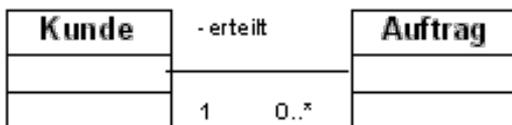
1.1 Assoziation

Sie drückt das Verhältnis von zwei völlig selbständigen Objekten aus,



die auf der gleichen Abstraktionsebene stehen und eigentlich nichts miteinander zu tun haben, aber unter bestimmten Gesichtspunkten in eine lose „kennt“-Beziehung ('KNOWS') oder auch „hat“-Beziehung gebracht werden können.

Beispiel: *Kunde Meier* (ein mögliches Objekt der Klasse *Kunde*) erteilt *Auftrag1* (ein mögliches Objekt der Klasse *Auftrag*) zur Softwareerstellung

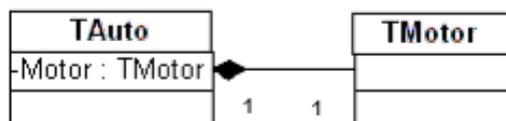


Kunde Meier kann einen, mehrere oder keinen Auftrag erteilt haben. Die Verbindung wird also erst dann aufgebaut, wenn der Auftrag erteilt wird und kann gelöscht werden, wenn das Produkt geliefert wurde.

1.2 Die Aggregation

Die Aggregation ist der Sonderfall einer Assoziation (s.o.). Sie drückt ein starkes semantisches Verhältnis von zwei an sich selbständigen Objekten aus, von denen eines Teil des anderen ist ('IS-PART-OF').

Beispiel: *Der Motor ist Teil eines Autos*



Motor als komplexes Objekt wird dem Autoobjekt hinzugefügt (aggregiert) und damit zu einem Attribut von TAuto. Es wird daher als eingebundenes Komponentenobjekt bezeichnet.

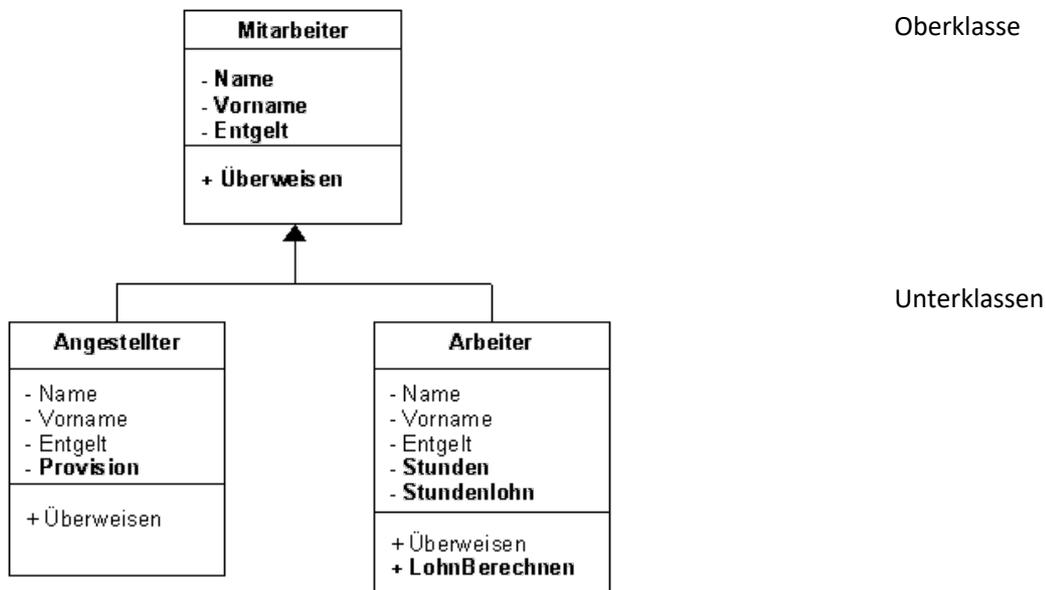
Man kann man zwei Arten der Aggregation finden:

- physische Aggregation: Das Analyseobjekt besteht aus Teilen. Der Motor ist Teil eines Autos (Komposition).
- logische Aggregation: Begriffliche Objekte werden als ein Teil angesehen. Die Adresse gehört zu einer Person.

2. Vererbung

Kinder haben Eltern. Sie erben von den Eltern Eigenschaften wie Haarfarbe, Hautfarbe und auch oft Verhaltensweisen. Diese Beziehung lässt sich abbilden und ist ein besonderes, wichtiges Merkmal der OOP.

Beispiel: *In einem Betrieb gibt es verschiedene Arten von Mitarbeitern.*



Alle drei haben die gleichen Personenattribute wie Name, Vorname usw., die sich quasi überdecken. Man spricht daher auch von einer Überdeckungsbeziehung. Ein Arbeiter ist eine Art von Mitarbeiter ('IS-KIND-OF' oder 'IS-A'), der aber eine besondere Form der Lohnberechnung hat.

Sie haben aber auch unterschiedliche, spezielle Attribute wie Stundenlohn, Arbeitsstunden beim Arbeiter und unterschiedliche Methoden der Gehaltsberechnung. Die Vererbungsbeziehung ist damit eine *Spezialisierung vom Allgemeinen (Gemeinsamen) zum Besonderen*, wie Sie auch bei der wissenschaftlichen Klassifikation angewandt wird.

Arbeiter erbt alle Attribute und Methoden von Mitarbeiter (Name, Vorname, Entgelt, Zeigen usw.) und fügt Stunden und LohnBerechnen als spezielle hinzu. Die geerbten sind sofort verfügbar und werden nicht noch einmal aufgeführt. (Wenn man es doch tut, werden die geerbten 'überdeckt', d.h. außer Kraft gesetzt, und es gelten dann die eigenen!)

Je nachdem, in welcher Richtung man Ober- und Unterklassen untersucht, spricht man von

Spezialisierung: ausgehend von der Oberklasse werden in der Unterklasse Spezialisierungen definiert (s.o.)

oder

Verallgemeinerung: Gemeinsamkeiten von Unterklassen werden extrahiert und in einer Oberklasse zusammengefasst.