

Technik

Lesen von Klassendiagrammen

Stichworte

Klasse, Attribut, Generalisierung, Assoziation, Aggregation

Motivation

Ein gegebenes (einfaches) Klassendiagramm muss gelesen und verstanden werden.

Ein Beispiel zum Lesen

Abbildung 1 zeigt ein Klassendiagramm mit den Klassen *Kunde*, *Ticket* und *Coupon*, ihren Attributen sowie ihren Assoziationen.

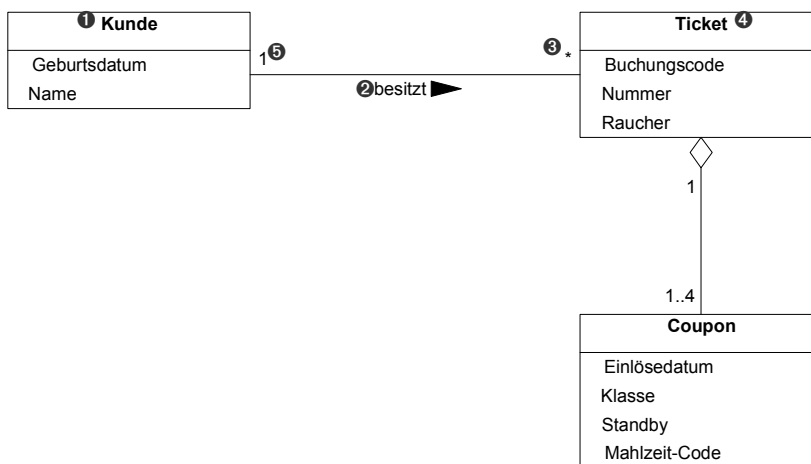


Abbildung 1 Ein Klassendiagramm mit Assoziationen

Liest man das Klassendiagramm in Abbildung 1, so kann man die Assoziation zwischen den Klassen *Kunde* und *Ticket* wie folgt lesen:

- Ein (der Satz beginnt immer mit „EIN“) Objekt der einen Klasse hat eine Assoziation zu einer Anzahl von Objekten der anderen Klasse.

In dieser ersten abstrakten und allgemein gültigen Formulierung muss man die entsprechenden Werte aus dem Diagramm einfügen. Der Name der einen Klasse ist *Kunde*¹, der Name der anderen Klasse ist *Ticket*⁴ und der Name der Assoziation ist *besitzt*²:

- Ein *Kunde*¹ *besitzt*² *³ *Ticket*⁴.

Wird der Stern durch seine Bedeutung ersetzt, entsteht der normale deutsche Satz:

- Ein *Kunde*¹ *besitzt*² null, 1 oder mehrere³ *Ticket(s)*⁴.

Da Assoziationen zwischen zwei Klassen normalerweise nicht gerichtet sind, d.h. in beide Richtungen laufen, hat unserer Assoziation auch in die andere Richtung eine Bedeutung:

- Ein *Ticket*⁴ ist im Besitz von² genau einem³ *Kunden*¹.

Das kleine Dreieck beim Namen der Assoziation² gibt an, in welche Richtung der Name der Assoziation gilt.

Auf diese Art und Weise lassen sich alle Assoziationen im Klassendiagramm lesen.

Die Angabe der Anzahl der Objekte der *anderen Klasse* (man beginnt immer mit einem Objekt der *einen Klasse*) nennt man *Multiplizität*. Dabei wird immer nach dem gleichen Schema vorgegangen:

Zuerst die Angabe der *Untergrenze* (mindestens so viel) gefolgt von zwei Punkten (..) und der Angabe der *Obergrenze* (höchstens so viel).

Abbildung 2 zeigt die gebräuchlichsten Möglichkeiten auf:

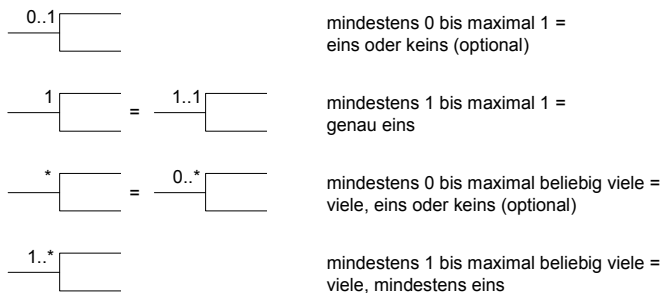


Abbildung 2 Multiplizitäten

In der UML ist es jedoch auch möglich, für die Ober- und Untergrenze beliebige Werte einzusetzen, z.B. 2..4 oder 6..*.

Der Name der Assoziation ist notwendig, um die *fachliche Bedeutung* der Assoziation zu verstehen. Im Gegensatz zur Assoziation selbst, die in beide Richtungen gilt, bezieht sich der Name immer nur in eine Richtung, die mit einem schwarzen Dreieck angezeigt wird. Ist die Assoziation nicht beschriftet, so muss ihre Bedeutung aus dem fachlichen Zusammenhang herausgefunden werden oder es wird eine allgemeine Bedeutungen wie *hat* oder *gehört zu* angenommen. Im Zweifelsfalle sollte man Assoziationen lieber zu viel als zuwenig beschriften. Viele Diagramme, die wir in der Praxis angetroffen haben, waren nicht verständlich, weil die Assoziationen nicht beschriftet waren.

Assoziationen können auch als die Umsetzung von statischen Geschäftsregeln betrachtet werden. Aussagen wie *ein Ticket gehört zu genau einem Kunden* werden mittels Assoziationen im Klassendiagramm dokumentiert.

Rollen sind eine weitere Möglichkeit in der UML, der Beziehung zwischen den Klassen eine fachliche Bedeutung zu geben. Dabei wird angegeben, welche Rolle ein Objekt der einen Klasse gegenüber den Objekten der anderen Klasse spielt.

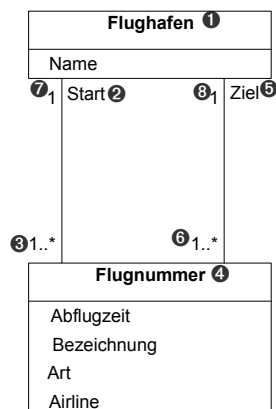


Abbildung 3 Ein Klassendiagramm mit Rollen

Liest man das Klassendiagramm in Abbildung 3 so kann die linke Assoziation mit Rollen zwischen den Klassen *Flugnummer* und *Flughafen* wie folgt gelesen werden:

- Ein *Flughafen*¹ ist *Start(ort)*² für *1 oder mehrere*³ *Flugnummern*⁴.

Zwischen den beiden Klassen *Flugnummer* und *Flughafen* gibt es noch eine zweite Assoziation:

- Ein *Flughafen*^① ist *Ziel(ort)*^⑤ für 1 oder mehrere^⑥ *Flugnummern*^④.

Auch diese beiden Assoziation haben eine Umkehrung, obwohl nur in die eine Richtung Rollen angegeben sind:

- Eine *Flugnummer*^④ hat als *Start(ort)*^② genau 1^⑦ *Flughafen*^①.
- Eine *Flugnummer*^④ hat als *Ziel(ort)*^⑤ genau 1^⑧ *Flughafen*^①.

Damit ist festgehalten, dass eine bestimmte Flugnummer einen Startflughafen und einen Zielflughafen hat. Ein Beispiel für eine Flugnummer ist LX 859, ein täglicher Flug der Crossair von Edinburgh nach Zürich.

Unter den vielen fachlichen Bedeutungen, die eine Assoziation haben kann, gibt es eine, die in der UML durch ein eigenes Symbol bezeichnet werden kann: die Ganzes-Teil-Beziehung oder eben Aggregation. Dieser Beziehungstyp wird immer dann verwendet, wenn Objekte einer Klasse Teile von Objekten einer anderen Klasse sind.

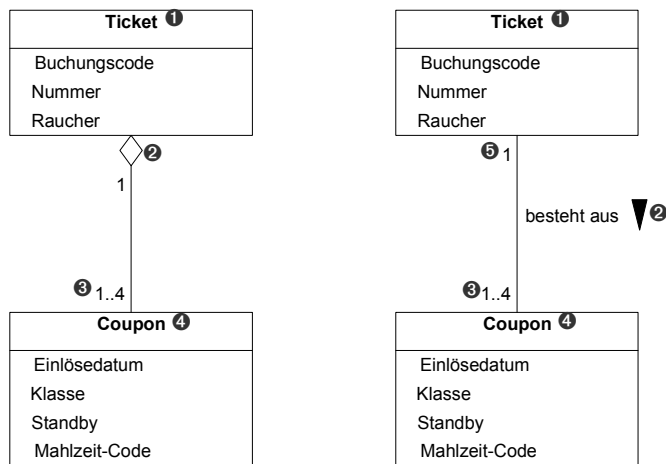


Abbildung 4 Ein Klassendiagramm mit Aggregation

Im Klassendiagramm in Abbildung 4 wird auf der linken Seite Aggregation verwendet (der weiße Rombus), was sich wie folgt liest:

- Ein *Ticket*^① besteht aus^② 1 bis 4^③ *Coupons*^④.
- oder umgekehrt:
- Ein *Coupon*^④ ist Teil von^② genau 1^⑤ *Ticket*^①.

Das Beispiel ohne Rhombus, dafür mit einem Namen für die Assoziation, hat die exakt gleiche Bedeutung!

Das letzte fehlende Element der UML, das wir für die Modellierung von Klassendiagrammen verwenden, ist die Generalisierung / Spezialisierung. Es dient dazu, die Beziehung zwischen einer Oberklasse und einer Unterklasse darzustellen.

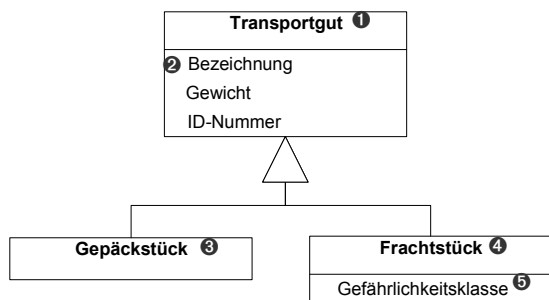


Abbildung 5 Ein Klassendiagramm mit Generalisierung/Spezialisierung

Die Generalisierung/Spezialisierung im Klassendiagramm der Abbildung 5 kann man von oben nach unten oder von unten nach oben lesen.

Beginnt man oben, so findet man eine Klasse Transportgut¹ mit den Attributen Bezeichnung, Gewicht und ID-Nummer². Von dieser Klasse gibt es zwei Spezialisierungen, Gepäckstück³ und Frachtstück⁴. Beide Unterklassen erben alle Attribute² von Transportgut¹, die Klasse Frachtstück hat zusätzlich noch das Attribut Gefährlichkeitsklasse⁵.

Beginnt man unten, so findet man die Klassen Gepäckstück³ und Frachtstück⁴. Diese haben eine Oberklasse, die Klasse Transportgut¹, welche die gemeinsamen Attribute (und Funktionen) der Unterklassen enthält.

Referenzen

1. KnowSolution "Statik von Geschäftsobjekten T2": Notation von Klassendiagrammen
2. Patrick Grässle, Henriette Baumann, Philippe Baumann: UML projektorientiert - Geschäftsprozeßmodellierung, IT-System-Spezifikation und Systemintegration mit der UML, Galileo Press, 2000, ISBN 3-934358-58-6