# Assoziationen

## Assoziationen Modellierung

Eine Beziehung zwischen zwei oder mehreren Klassen wird als Assoziation bezeichnet. Eine Assoziation beschreibt die Beziehung aller Objekte der beteiligten Klassen. Sie wird auf Objektebene hergestellt und auf Klassenebene modelliert.



Beispielsweise besteht zwischen der Klasse *Kunde* und der Klasse *Girokonto* eine Beziehung. Denn ein Kunde kann ein Girokonto haben und der Kontoinhaber eines Girokontos ist ein Kunde. Beziehungen zwischen Klassen (Assoziationen) werden genauer beschrieben durch **Rollen**, **Multiplizitäten** und **Navigierbarkeit**.

### Rollen

Rollennamen beschreiben die Bedeutung der Objekte der assoziierten Klasse näher.



Aus der Sicht des Girokontos ist der Kunde der Kontoinhaber. Aus der Sicht des Kunden ist das Girokonto sein Konto. Die Rolle einer assoziierten Klasse kann ggf. Auswirkungen auf die Multiplizitäten und die Navigierbarkeit haben.

### Multiplizitäten

Multiplizitäten geben an mit wie vielen Objekten der assoziierten Klasse ein Objekt verbunden werden kann (vgl. Kardinalitäten bei ERDs). Sie stellen folglich Mengenverhältnisse zwischen Klassen dar.

|  |  |
| --- | --- |
| Jeder Kunde hat **kein** oder **ein** Girokonto.  Jedes Girokonto hat genau **einen** Kontoinhaber (= Kunden). |  |

Im oben beschriebenen Beispiel kann ein Kunde ein Girokonto haben. Es ist aber auch vorstellbar, dass der Kunde bei der Bank kein Girokonto hat, sondern nur ein Sparkonto. Deswegen spricht man hier von einer **Kann-Assoziation** (Untergrenze 0). Dagegen muss ein Girokonto einen Kunden als Kontoinhaber haben, weswegen hier von einer **Muss-Assoziation** (Untergrenze > 0) gesprochen wird.

Mögliche Multiplizitäten sind:

|  |  |  |
| --- | --- | --- |
| 0 .. 1 | kein oder ein assoziiertes Objekt | KANN-Assoziation |
| 1 | genau ein assoziiertes Objekt | MUSS-Assoziation |
| \* | kein, ein oder beliebig viele assoziierte Objekte | KANN-Assoziation |
| 1 .. \* | ein oder beliebig viele assoziierte Objekte | MUSS-Assoziation |
| n .. m | von n bis m assoziierte Objekte | - |

Weitere Beispiele:

|  |  |
| --- | --- |
| Jeder Kunde hat **kein**, **ein** oder **beliebig viele** Girokonten.  Jedes Girokonto hat **ein** oder **beliebig viele** Kontoinhaber (= Kunden). |  |
| Jeder Kunde hat kein, ein oder beliebig viele Girokonten.  Jedes Girokonto hat **1** oder **4** Kontoinhaber (= Kunden). |  |

### Navigierbarkeit

Bei der Navigierbarkeit wird die Frage beantwortet, wer kann auf wen zugreifen? Dabei unterscheidet man zwischen **unidirektionalen** und **bidirektionalen** Assoziationen.

|  |  |
| --- | --- |
|  |  |
| **unidirektionale Assoziation** | **bidirektionale Assoziation** |
| Jedes Objekt der Klasse *Girokonto* kennt seinen Kontoinhaber (= *Kunde*). Allerdings kennt ein Objekt der Klasse *Kunde* nicht seine Girokonten. Das heißt, man kann über das Girokonto z.B. den Namen des Kontos ausgeben, aber nicht über den Kunden den Kontostand seines Girokontos.  Die Navigierbarkeit ist also nur in eine Richtung möglich, nämlich von Kunde zu Girokonto (Pfeil), aber nicht von Girokonto zu Kunde („X“) | Jedes Objekt der Klasse *Kunde* weiß, welche Girokonten ihm zugeordnet sind. Umgekehrt weiß aber auch jedes Girokonto welche Kontoinhaber (= Kunden) es hat. Die Navigierbarkeit ist in beide Richtungen möglich. Das heißt, über einen Kunden kann der Kontostand seines Girokontos herausgefunden werden und über ein Girokonto kann der Name des Kunden ermittelt werden. |

|  |  |
| --- | --- |
| Neben der Pfeilspitze und dem „X“ kann auch „nichts“ geschrieben werden. Dies bedeutet, dass noch keine Beziehung festgelegt wurde. |  |

Mögliche Beziehungen:

|  |  |
| --- | --- |
| Pfeilspitze | Zugriff auf die Klasse möglich. |
| „X“ | Kein Zugriff auf die Klasse möglich. |
| (nichts) | Keine Beziehung festgelegt. |