**Klassendiagramm-Beziehungen**

Klassen sind auf bestimmte Weise miteinander verbunden. Zu den Beziehungen in Klassendiagrammen gehören insbesondere verschiedene Arten von logischen Verbindungen. Im Folgenden sind solche Arten von logischen Verbindungen aufgeführt, die in der UML möglich sind:

- Assoziation

- gerichtete Assoziation

- Reflexive Assoziation

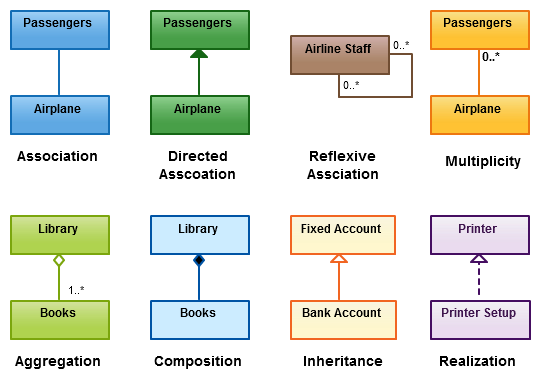
- Multiplizität

- Aggregation

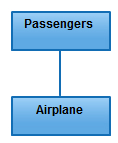
- Komposition

- Vererbung/Verallgemeinerung

- Verwirklichung

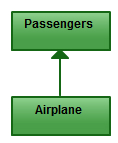
[](https://d3n817fwly711g.cloudfront.net/blog/wp-content/uploads/2012/03/Class-Diagram-Relationships.png)Relationships in UML class diagrams

**Association**

[](https://d3n817fwly711g.cloudfront.net/blog/wp-content/uploads/2012/03/Association-Relationship.jpeg)

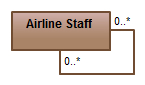
Assoziation ist ein weit gefasster Begriff, der so gut wie jede logische Verbindung oder Beziehung zwischen Klassen umfasst. Zum Beispiel können Passagier und Fluggesellschaft wie oben beschrieben verknüpft werden.

**Directed Association**

[](https://d3n817fwly711g.cloudfront.net/blog/wp-content/uploads/2012/03/Directed-Association-Relationship.jpeg)

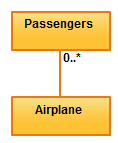
Die gerichtete Assoziation bezieht sich auf eine gerichtete Beziehung, die durch eine Linie mit einer Pfeilspitze dargestellt wird. Die Pfeilspitze stellt einen in einem Container enthaltenen gerichteten Fluss dar.

**Reflexive Association**

[](https://d3n817fwly711g.cloudfront.net/blog/wp-content/uploads/2012/03/Reflexive-Association-Relationship.jpeg)

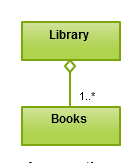
Reflexive Assoziation. Dies ist der Fall, wenn eine Klasse mehrere Funktionen oder Verantwortlichkeiten haben kann. Ein Flughafenmitarbeiter kann zum Beispiel Pilot, Luftfahrtingenieur, Ticketabfertiger, Wachmann oder Wartungspersonal sein. Wenn das Mitglied der Wartungsmannschaft vom Luftfahrtingenieur verwaltet wird, könnte es in zwei Instanzen derselben Klasse eine Beziehung "verwaltet von" geben.

**Multiplicity**

[](https://d3n817fwly711g.cloudfront.net/blog/wp-content/uploads/2012/03/Multiplicity-Relationship.jpeg)

Multiplizität ist die aktive logische Assoziation, wenn die Kardinalität einer Klasse im Verhältnis zu einer anderen dargestellt wird. Zum Beispiel kann eine Flotte mehrere Flugzeuge umfassen, während ein Verkehrsflugzeug null bis viele Passagiere enthalten kann. Die Notation 0..\* im Diagramm bedeutet "null bis viele".

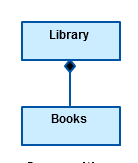
**Aggregation**

[](https://d3n817fwly711g.cloudfront.net/blog/wp-content/uploads/2012/03/Aggregation-Relationship.png)

Unter Aggregation versteht man die Bildung einer bestimmten Klasse als Ergebnis der Aggregation oder Bildung einer Sammlung einer Klasse. So besteht beispielsweise die Klasse "Bibliothek" aus einem oder mehreren Büchern und anderen Materialien. Bei der Aggregation sind die enthaltenen Klassen nicht stark vom Lebenszyklus des Containers abhängig. Im gleichen Beispiel bleiben die Bücher auch dann erhalten, wenn die Bibliothek aufgelöst wird. Um die Aggregation in einem Diagramm darzustellen, zeichnen Sie eine Linie von der übergeordneten Klasse zur untergeordneten Klasse mit einer Rautenform in der Nähe der übergeordneten Klasse.

Um die Aggregation in einem Diagramm darzustellen, zeichnen Sie eine Linie von der übergeordneten Klasse zur untergeordneten Klasse mit einer Rautenform in der Nähe der übergeordneten Klasse.

**Composition**

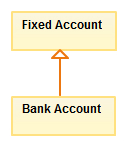
[](https://d3n817fwly711g.cloudfront.net/blog/wp-content/uploads/2012/03/Composition-Relationship-UML.png)

Komposition

Die Kompositionsbeziehung ist der Aggregationsbeziehung sehr ähnlich. Der einzige Unterschied besteht darin, dass der Hauptzweck der Kompositionsbeziehung darin besteht, die Abhängigkeit der enthaltenen Klasse vom Lebenszyklus der Containerklasse zu verdeutlichen. Das heißt, die enthaltene Klasse wird ausgelöscht, wenn die Containerklasse zerstört wird. So wird beispielsweise die Seitentasche einer Umhängetasche ebenfalls nicht mehr existieren, wenn die Umhängetasche zerstört wird.

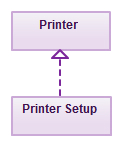
Um eine Kompositionsbeziehung in einem UML-Diagramm darzustellen, verwenden Sie eine gerichtete Linie, die die beiden Klassen verbindet, mit einer gefüllten Raute neben der Containerklasse und einem Richtungspfeil zur enthaltenen Klasse.

**Inheritance / Generalization**

[](https://d3n817fwly711g.cloudfront.net/blog/wp-content/uploads/2012/03/Inheritance-Relationship.jpeg)

Vererbung bezieht sich auf eine Art von Beziehung, bei der eine assoziierte Klasse ein Kind einer anderen ist, weil sie die gleichen Funktionen der Elternklasse übernimmt. Mit anderen Worten, die untergeordnete Klasse ist ein bestimmter Typ der übergeordneten Klasse. Um Vererbung in einem UML-Diagramm darzustellen, wird eine durchgezogene Linie von der untergeordneten Klasse zur übergeordneten Klasse mit einer ungefüllten Pfeilspitze gezeichnet.

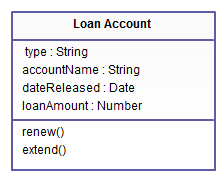
**Realization**

[](https://d3n817fwly711g.cloudfront.net/blog/wp-content/uploads/2012/03/Realization-Relationship.jpeg)

Als Realisierung wird die Implementierung der in einer Klasse definierten Funktionalität durch eine andere Klasse bezeichnet. Um die Beziehung in UML darzustellen, wird eine gestrichelte Linie mit einer ungefüllten durchgezogenen Pfeilspitze von der Klasse, die die Funktionalität definiert, zu der Klasse, die die Funktion implementiert, gezogen. In diesem Beispiel werden die Druckeinstellungen, die über die Schnittstelle für die Druckereinrichtung festgelegt werden, vom Drucker implementiert.

**What are the Class Diagrams?**

Klassendiagramme sind der wichtigste Baustein in der objektorientierten Modellierung. Sie werden verwendet, um die verschiedenen Objekte in einem System, ihre Attribute, ihre Operationen und die Beziehungen zwischen ihnen darzustellen.

Die folgende Abbildung ist ein Beispiel für eine einfache Klasse:[](https://d3n817fwly711g.cloudfront.net/blog/wp-content/uploads/2012/03/Class-Diagram.jpeg)

Einfaches Klassendiagramm mit Attributen und Operationen

Im Beispiel wird eine Klasse namens "Darlehenskonto" dargestellt. Klassen in Klassendiagrammen werden durch Kästchen dargestellt, die in drei Bereiche unterteilt sind:

1. Die obere Partition enthält den Namen der Klasse.

2. Der mittlere Teil enthält die Attribute der Klasse.

3. Die untere Partition zeigt die möglichen Operationen, die mit der Klasse verbunden sind.

Das Beispiel zeigt, wie eine Klasse alle relevanten Daten eines bestimmten Objekts auf eine sehr systematische und klare Weise kapseln kann. Ein Klassendiagramm ist eine Sammlung von Klassen ähnlich dem obigen Beispiel.