

## Allgemeines zur OOP (Objektorientierte Programmierung)

Dieser Text erläutert überblicksartig die beiden wichtigsten Begriffe der objektorientierten Programmierung (OOP): **Klassen und Objekte**.

Diese Begriffe werden in allen weiteren Kapiteln ständig auftauchen und ihr Verständnis ist unabdingbar, um die weiteren Abschnitte erfolgreich bearbeiten zu können.

Ziel bei der (Objektorientierten)Softwareentwicklung ist es, einen Teil der Realität in einem Informatiksystem abzubilden. Um diese Abbildung der Realität in eine (objektorientierte) Softwarearchitektur zu erreichen, muss ein **Modell** der Realität entwickelt werden. Dieses Modell besteht in der Objektorientierten Modellierung aus **Klassen**. Diese können in Beziehungen zueinanderstehen. **Grafisch** werden die Klassen und ihre Beziehungen in einem **UML-Diagramm** dargestellt (siehe unten).

### Klassen und Objekte<sup>1</sup>

Genauso wie ein Architekt einen Bauplan erstellt, erstellt der Softwarearchitekt einen Bauplan für das Programm-„Haus“.

Dabei hilft der Ansatz, die reale Welt mit Hilfe von Objekten abzubilden. Da es sich mit einzelnen Objekten wie z.B. der Person Jürgen Behrendt oder Kathrin Fischer nur schlecht modellieren lässt, gibt es das Konstrukt der Klasse. Eine Klasse stellt so etwas wie einen Bauplan für daraus abgeleitete Objekte bereit. Ein Objekt entsteht somit aus einer Klasse bzw. eine Klasse stellt eine Sammlung von Objekten dar.

Wenn ein Architekt ein Haus entwirft, so erstellt er eine Zeichnung, einen Plan des Hauses. Dieser Plan ist dann die Grundlage für die Errichtung des Hauses. Es ist auch möglich zu einem Plan mehrere Häuser zu bauen, die dann natürlich gleich aussehen. Man hat es hier mit einer Vorlage, dem Plan, und konkreten Realisierungen, den Häusern, zu tun. In der objektorientierten Programmierung entspricht dem Plan eine *Klasse* und den konkreten Häusern entsprechen *Objekte*. Man kann auch sagen: Es gibt mehrere Exemplare des Hauses. Ein Computerprogramm in einer objektorientierten Sprache versucht in der Regel einen Ausschnitt der realen Welt im Computer abzubilden. Dieser Ausschnitt besteht normalerweise aus mehreren Objekten, die miteinander kommunizieren bzw. Nachrichten austauschen. Viele dieser Objekte gehören einer gemeinsamen Gruppe an, die man Klasse nennt. Jedes Objekt gehört zu einer bestimmten Klasse. Mehrere Objekte können einer gemeinsamen Klasse angehören.

Allgemein: eine Klasse fasst **Gemeinsamkeiten** einer Gruppe von Objekten (**Instanzen**) zusammen,

- mit denselben Eigenschaften (Attributen)
- mit demselben Verhalten (Operationen oder auch Methoden genannt)

### **Man kann eine Klasse als einen Bauplan für Objekte auffassen.**

*Die Objekte werden zur Programmausführung aus dieser Vorlage Klasse erzeugt und „leben“ im Speicher während der Programmausführung.*

Um besser **Modellieren** zu können, benutzt man ein sogenanntes **Klassendiagramm**. Hier lassen sich Beziehungen von Klassen (und damit auch von Objekten untereinander) gut darstellen und strukturieren. Diese grafische Darstellung der Struktur lässt sich dann in objektorientierten Programmiersprachen umsetzen. Eine Konvention in der Darstellung von Klassendiagrammen heißt **UML** (Unified Modelling Language). Da UML eine formale Sprache ist, lässt sich diese grafisch gut darstellen.

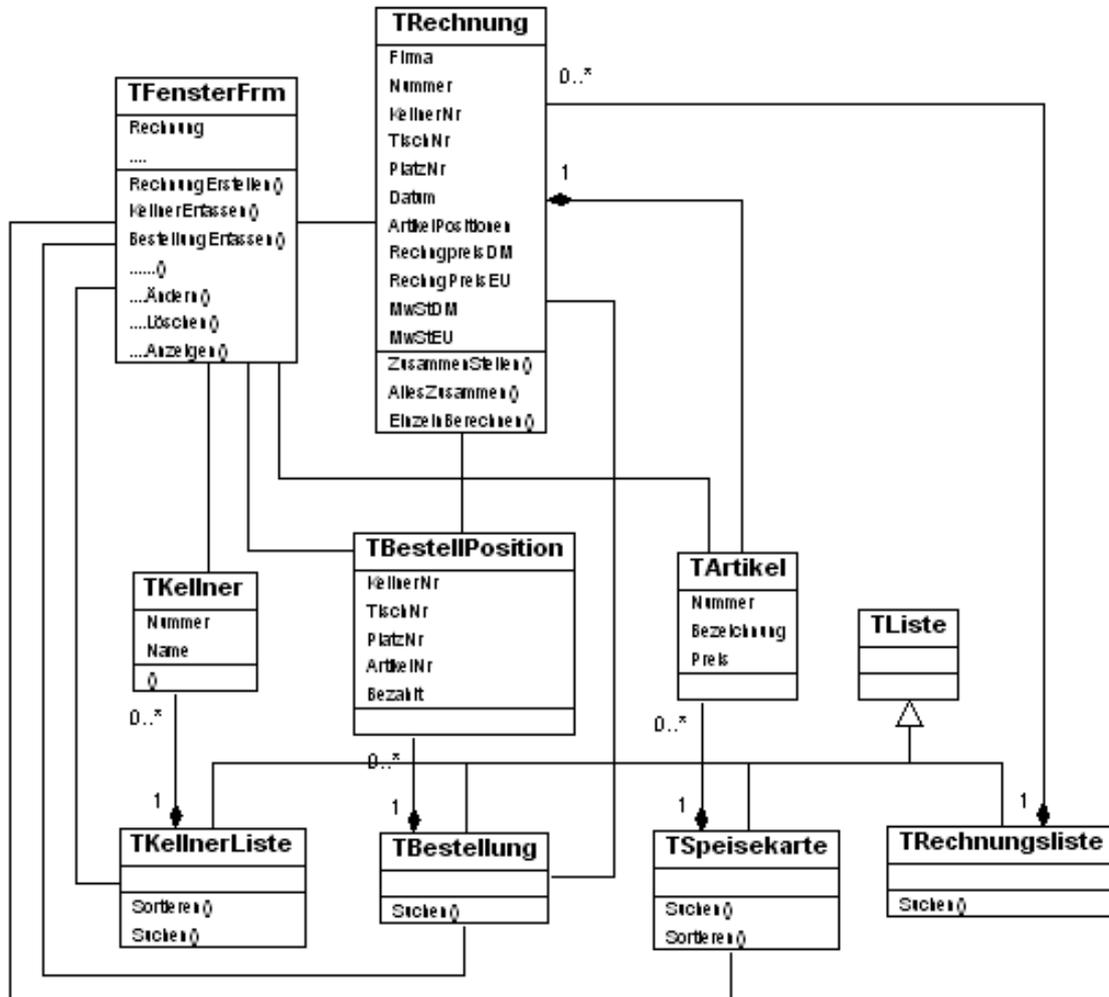
Hier ein Beispiel für ein komplexeres Klassendiagramm in der sogenannten UML-Notation. Dies ist eine Konvention zur Darstellung von Klassen und ihren Beziehungen untereinander.

In diesem Beispiel sind drei verschiedene Beziehungen erkennbar:

- **Assoziation** („hat“-Beziehung)
- **Aggregation** („besteht aus“-Beziehung) und
- **Vererbung**

---

<sup>1</sup> mit Material aus: Bernard Schriek: Informatik mit Java. S.16



2

UML-Diagramme erinnern an ERM-Diagramme aus der Datenbankmodellierung. Beide sind sich grundsätzlich sehr ähnlich. Der Kontext der Benutzung ist allerdings ein etwas anderer.

Beim ERM liegt der Fokus auf der Datenhaltung (die sogenannte Datenschicht).

Beim UML besteht der Fokus in der Abbildung von Funktionalitäten in einem Programm (Fachkonzept und GUI).

UML enthält insbesondere die Konzepte der Vererbung und der Aggregation, die eine besondere Form der Assoziation ist.

Grundsätzlich stellt die Objektorientierte Programmierung bei der Implementierung in einer Programmiersprache eine Erweiterung bekannter Datentypen zu einer neuen Datenstruktur dar. Eine Klasse beschreibt somit eine neue Datenstruktur (Abbildung der Realität). Aus der Klasse wird ein konkretes Objekt erzeugt. Das konkrete Objekt beinhaltet die bekannten Datentypen einer Programmiersprache und benutzt diese, um einen Zustand seiner Daten (Informationen) zu speichern.

<sup>2</sup> aus: [http://oszhd1.be.schule.de/gymnasium/faecher/informatik/projekte/k122/bilder/ood\\_3.1.gif](http://oszhd1.be.schule.de/gymnasium/faecher/informatik/projekte/k122/bilder/ood_3.1.gif)