



# Objektorientierung

---

Die bisher besprochene "klassische" Programmieretechnik wird prozedural genannt. Bei der Programmierung im Industrieumfeld wird heute aber nahezu ausschließlich von der Technik der Objektorientierung Gebrauch gemacht.

Auf der Seite der Anwendungsprogrammierer bedeutet das, daß man fertig vorhandene Bausteine von Komponentenherstellern übernimmt, und wie Lego zu einem Gesamtwerk zusammensteckt.

Kernbegriffe : Klassenbibliothek, Klasse, Objekt, Methode, Attribut, Namespace, Konstruktor.

## Klasse

ist eine Struktur, in der Programme und Daten enthalten sind.  
Beispiel : Eine Klasse Seitenfenster könnte eine Methode Öffnen und eine Methode Schießen haben. Als Attribute sind Motorstrom oder Schalterstellung oder ähnliches denkbar. Schreibweise ist immer : Klasse.Methode oder : Klasse.Attribut

## Klassenbibliothek

ist eine Sammlung verschiedener Klassen, die alle zu einem gleichen Aufgabengebiet gehören, also z.b eine Klassenbibliothek Karosserieelemente für eine Auto-Software, in der die Klasse Seitenfenster enthalten ist.

## Namespace

In großen Softwaresammlungen werden wegen der sehr vielen benötigten Elemente, die ja alle Namen haben müssen, die Strukturen in Bereiche eingeteilt, innerhalb derer solche Namen Gültigkeit haben. Diese Bereiche heißen Namespaces (Namensräume). Es könnte z.b. im namespace Motorsteuerung einen Namen Grenzstrom geben, und im Namespace Lichtsteuerung den Gleichen nochmal.

## Objekt

entsteht, wenn man eine Klasse im eigenen Programm benutzen möchte. Als Bild (das technisch nicht richtig, aber anschaulich gut ist) stelle man sich einen Stempel vor, der alle Methoden und Attribute einer Klasse ins eigene Programm reinstempelt. Der Abdruck ist das Objekt. Es kann öfter benutzt werden z.B. die Klasse Seitenfenster vier mal, wobei jeder Abdruck einen eigenen Namen bekommt, also z.B. Fahrerfenster, Beifahrerfenster usw.

## Konstruktor

ist die programmtechnische Umsetzung des Stempels, der das Objekt erzeugt

Wichtig ist, daß jedes Objekt seine eigenen Attribute hat, die von anderen Objekten nicht überschrieben werden (z.B. vier verschiedene Motorströme an den vier Fensterhebern).

Man kann das alles auch noch modellhafter beschreiben, vielleicht hilft ihnen das :

Eine Klasse ist wie ein Bauplan für ein Haus.

Der Konstruktor baut dieses Haus.

Das Objekt ist das fertige, "funktionsfähige" Haus.

Ein Attribut ist eine Eigenschaft dieses speziellen Hauses, Farbe z.b.

Eine Methode ist eine Funktion des Hauses. Hier hinkt das Modell ein wenig, weil ein Haus eigentlich "nichts kann". ("alt werden" vielleicht ?)

Prinzipielle Vorgehensweise :

Nachschauen, welche Klasse die gewünschten Funktionen beinhaltet. Lesen der Dokumentation.

Zugang zur Klassenbibliothek einbinden

Erzeugen eines Objekts im eigenen Programm, um diese Funktionen zugänglich machen

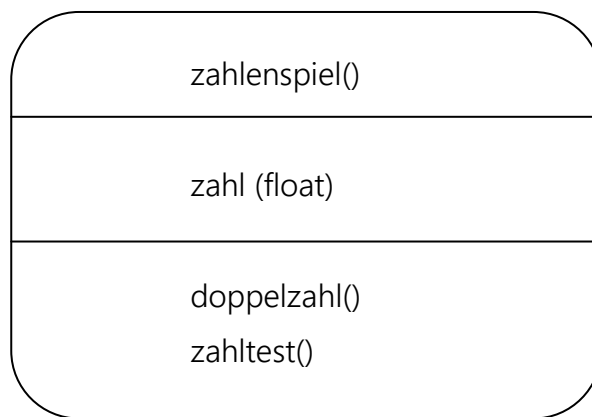
Die Methoden aufrufen, Attribute benutzen

## Beispiel zur Objektorientierung

---

Man beschreibt Klassen mit dem sogenannten Klassendiagramm, das etwa so wie unten aussehen kann.

Beschreibung der Klasse zahlenspiel :



Die Klassenbibliothek heißt add\_lib.py

Die Klasse heißt zahlenspiel()

Die Methode doppelzahl() verdoppelt bei Aufruf den Inhalt des Attributs zahl.

Die Methode zahltest() prüft den Inhalt des Attributs zahl, dieses wird verdoppelt, falls der Wert unter 100 ist, halbiert, falls der Wert über 150 liegt, ansonsten unverändert ausgegeben.