



1. Die Objektorientierte Modellierung (OOM)

Die Objektorientierte Modellierung (OOM) stellt eine der bedeutendsten Erweiterungen der Programmiermethoden der vergangenen 10-15 Jahre dar.

Dabei handelt es sich um ein Konzept, wie Software geplant und Schritt für Schritt umgesetzt werden kann. Sinn ist v.a. die schnelle Realisierung von Software-Projekten. Daneben hat OOM den Vorteil, meist leicht erweiterbar zu sein und Programmcode in anderen Programmen leicht wiederverwenden zu können.

Die Objektorientierte Modellierung ist uns bereits seit den Datenbanken bekannt. Erweitert wird die Klasse allerdings noch um die Methoden.

Bsp.: Die Klasse "EINTRAG" für ein Gästebuch

EINTRAG
\$Name \$Datum \$eMail \$Text
anzeigen(); aendern(); loeschen();

Der Programmierer kann nun Stück für Stück diese Klasse implementieren. Die zu erzeugenden Objekte haben also vier Variablen. In der Klasse werden die drei Funktionen implementiert. Diese Funktionen stehen dann allen Objekten (Instanzen) der Klasse zur Verfügung.

Aus den Datenbanken kennen wir die Punktnotation, die auch in vielen OOM-basierten Programmiersprachen verwendet werden, z.B.

```
Eintrag1.anzeigen();  
Eintrag2.loeschen();
```

In PHP wird der Punkt durch ein "->" ersetzt, die Objekte haben außerdem das vorangestellte \$-Zeichen:

```
$Eintrag1->anzeigen();  
$Eintrag2->loeschen();
```

Der Aufbau einer Klasse in PHP ist einfach:

2. Realisierung in PHP

```
// Jede Klasse beginnt mit dem Wort class und einem selbstgewählten  
// Bezeichner.  
class Eintrag {  
  
// Dann werden die Variablen deklariert  
var $Name;  
var $Datum;  
var $eMail;  
var $Text;
```



```
// Ein Konstruktor sorgt für das Erstellen der Objekte, er hat
// denselben Bezeichner wie die Klasse, hier werden die Attribute
// beim Instanzieren mit einem ersten Wert versehen.
// Werden innerhalb der Klasse Eintrag die Attribute des Objektes
// angesprochen, so wird ihnen der Zusatz $this-> vorangestellt!
function Eintrag($N,$D,$e,$T){
    $this->Name=$N;
    $this->Datum=$D;
    $this->eMail=$e;
    $this->Text=$T;
}

// Alle weiteren Funktionen werden auch innerhalb der Klasse realisiert
function anzeigen(){

echo $this->Name, " (eMail: ",$this->eMail;
echo ") hat am ", $this->Datum, " folgenden Eintrag verfasst:<br>";
echo $this->Text,"<br>";

}

die anderen beiden Funktionen werden dann ebenfalls hier implementiert.

}
```

3. Das Erzeugen der Objekte und die Methoden-Aufrufe

In PHP können nun Objekte (Instanzen) der Klasse erzeugt werden durch den Befehl **new**.

```
Bsp: $Eintrag1 = new Eintrag("Müller","2005-02-02","info@test.de",
    "Hallo Leute, bla bla bla, viele Gruesse euer Heinz");

    $Eintrag2 = new Eintrag("Test","2.2.05","keine","Hurra, Hurra");
```

Die Funktionen können nun durch die Objekt-Notation aufgerufen werden:

```
$Eintrag2->anzeigen();
```

Aufgabe:

Erstelle ein Programm, das fünf Objekte der Klasse EINTRAG erzeugt. Die Objekte sollen in einem Array liegen. Programmiere einer Schleife, in der durch einen einzigen Befehl alle Einträge am Bildschirm angezeigt werden!