

## Abgabe:

Diese Laboraufgabe ist in der kommenden Woche abzugeben.

## Serverseitige-OOP mit PHP

LESEN SIE ZUNÄCHST DIE GESAMTE AUFGABENSTELLUNG DURCH!

In diesem Labor sollen Sie Klassen erstellen welche im nächsten Labor die Daten aus der Datenbank bereitstellen. In diesem Labor sollen die Objekte gerendert werden.

Es werden die Dateien Helper.php und dataOOP.inc.php mitgeliefert. Erklärung folgt.

Alle zu erstellende Klassen sollen von der Klasse Helper erben. Zu jeder Klasse kann es eine View Datei mit dem Schema „ClassName.phtml“ geben. Erklärung zur Verwendung der View folgt.

Entwickelt werden sollen folgende Klassen (Klassenname verwenden, auch als Dateiname.php):

- **GalleryCollection** ~~Fasst alle Gallery Images in einer Collection zusammen und kann diese durch folgende Methoden zurückgeben.~~
  - ~~Die Klasse soll die folgenden Methode besitzen:~~
    - ~~**Get()** Liefert alle Gallery Images aus dem Array \$ImagesData zurück. Rückgabe ist ein Array von GalleryImage **Objekten** mit der Image-Uid als Key.~~
- **GalleryImage** ~~Jedes einzelne Image kann als GalleryImage Objekt erstellt werden. Jedes GalleryImage kann z.B. seine Kommentare zurück geben.~~
  - ~~Die Klasse soll die folgende Methode besitzen:~~
    - ~~**GetComments()** Gibt ein Array aller Comments zu diesem Blogpost zurück~~
- **GalleryImageComment** ~~Jeder einzelne Kommentar kann als GalleryImageComment Objekt erstellt werden. Jeder **GalleryImageComment** gibt alle Kommentardaten als Array zurück.~~
  - Die Klasse braucht keine weiteren Methoden durch die Vererbung.

Die Ausgabe in der index.php soll nach der Implementierung der Klassen nur noch auf diese zugreifen und über die Objekte ausgeben. Verwenden Sie die Methode „Render()“!

## Erklärung der Verwendung der Klassen

Jedes Objekte dessen Klasse von Helper erbt besitzt folgende Fähigkeiten / Methoden:

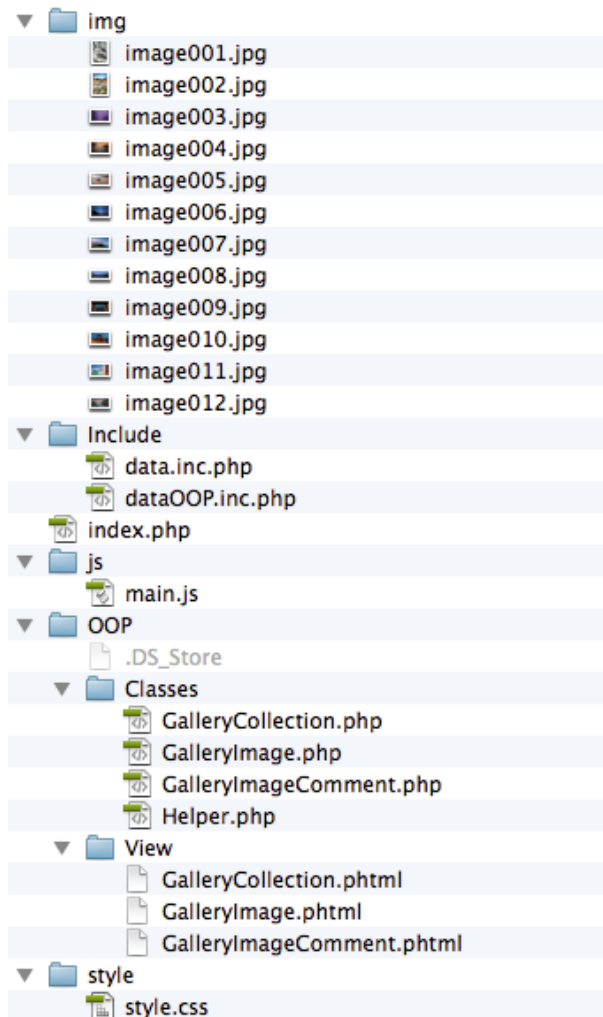
- **\_\_construct(\$init)** - Jedes Objekt erhält einen Konstruktor. Dieser kann 3 Typen an Daten annehmen.
  - null - wenn keine Daten im Konstruktor angegeben werden
  - int - Uid des Datensatzes. Der Konstruktor holt alle weiteren Daten. Funktioniert für die Objekte „GalleryImage“ und „GalleryImageComment“. Auf die Daten kann über den Getter zugegriffen werden.
  - array - Daten werden für das Objekt abgespeichert und können über Getter abgefragt werden. Sinnvoll in der Methode **GalleryCollection->Get()** um die Daten nicht doppelt zu holen.
- **Render()** - Schaut ob es eine Datei „View/Classname.phtml“ (View) gibt. Wenn vorhanden wird der Inhalt ausgegeben. Innerhalb der View kann über „\$this“ auf das zu rendernde Objekt zugegriffen werden.  
Gibt es keine View Datei, wird ein print\_r() des Objekts ausgegeben.
- **Magic Getter** - Alle Objektdaten werden in dem protected Array `$_Data` abgespeichert. Um auf diese Daten zuzugreifen, kann vom Objekt über `$Obj->GetValue()` zugegriffen werden.

## Vorgehen Umsetzung OOP

Folgende Dateien sollen eingebunden werden:

```
include_once 'Include/data00P.inc.php';  
include_once 'OOP/Classes/Helper.php';  
include_once 'OOP/Classes/GalleryCollection.php';  
include_once 'OOP/Classes/GalleryImage.php';  
include_once 'OOP/Classes/GalleryImageComment.php';
```

Legen Sie zunächst alle benötigten Dateien und Ordner an. Empfohlen wird folgende Struktur:



Erstellen Sie eine neue Klasse in einem Unterordner (OOP/Classes) dieser Datei mit dem Namen "GalleryCollection.php". Erstellen Sie die Klasse „GalleryCollection“, erben Sie von dem Helper (extends) und implementieren Sie die Methoden, so dass Sie über

```
$Collection = new GalleryCollection;  
$GalleryImagesData = $Collection->Get();
```

ein Array aller GalleryImages erhalten welche Sie nun rendern können.

Um die Daten zu bearbeiten, greifen Sie bitte wieder auf Bedingungen und Schleifen zurück.