

PHP8 - Kompaktkurs

Dr.sc.nat. Michael J.M. Wagner, New Elements*

Revision 1.75



*michael@wagnertech.de

Inhaltsverzeichnis

1	Einführung	4
1.1	Was ist PHP?	4
1.2	Grundlagen	4
1.3	Verzweigungen	6
1.4	Schleifen	7
1.5	Funktionen	8
2	Dateien	9
2.1	Zugriffe auf Dateien und Verzeichnisse	9
2.2	POST/REDIRECT/GET-Muster	10
2.3	Dateien auf den Server hochladen	10
3	Felder und Zeichenketten	11
4	Fehlerbehandlung	13
5	Modularisierung	14
6	Objektorientierung	15
6.1	Klassen	16
6.2	Vererbung	19
6.3	Abstrakte Klassen, Schnittstellen und Traits	21
7	Weitere Themen	22
7.1	Namensräume	22
7.2	Datum und Zeit	22
7.3	Sessions und Cookies	23
7.4	Reguläre Ausdrücke	25
7.5	Socketkommunikation	26
7.6	Grafik	27
8	Datenbanken mit MySQL	27
8.1	MySQL und phpMyAdmin	28
8.2	PHP und MySQL	28
9	XML	29
9.1	SimpleXML	30
9.2	DOMDocument	31
10	Ajax	32
10.1	Hallo Ajax	32
10.2	Parameter senden	32
10.3	XML-Datei lesen	32
11	PHP 8	33
12	Quellen	34

IT-Schulungen.com Portfolio

IT-Schulungen.com ist eines der führenden, herstellerunabhängigen Seminarportale von Schulungen rund um die Informationstechnologie (IT) und das IT-Management. Seit über 15 Jahren ist IT-Schulungen.com eine anerkannte Anlaufstelle für viele Unternehmen und Behörden, wenn es um die Durchführung von DACH-weiten Schulungen geht.

- | | | |
|--------------------------------------|---------------------------|-----------------------|
| • Applikationsserver /
Middleware | • ERP-Systeme | • Open Source |
| • Business Intelligence | • IT Management | • Portale |
| • Business-Skills und
Führung | • IT-Recht / Lizenzierung | • SAP® |
| • Cloud | • ITIL | • Security |
| • CRM | • Mobile | • Serversysteme |
| • Datenbanken | • Multimedia | • Softwareentwicklung |
| • eBusiness | • Office | • Systemmanagement |

www.IT-Schulungen.com

New Elements GmbH | IT-
Schulungen.com

Zertifizierungen & Partnerschaften



www.IT-Schulungen.com

New Elements GmbH | IT-
Schulungen.com

1 Einführung

1.1 Was ist PHP?

PHP ist die Abkürzung für PHP Hypertext Preprocessor. PHP ermöglicht Entwicklern die Erzeugung dynamischer Internetseiten, mit denen sogenannte Web Applications erstellt werden, wie z. B. E-Commerce-Systeme, Chats oder Foren. Im Unterschied zu statischen Internetseiten kann sich der Inhalt aufgrund von Aktionen des Benutzers oder neuer Basisinformationen, die z. B. aus Datenbanken stammen, jederzeit ändern.

PHP unterstützt insbesondere die einfache Auswertung von Formularen, mit denen ein Benutzer Daten an eine Website senden kann. Es ermöglicht die Zusammenarbeit mit vielen verschiedenen Datenbanksystemen. Die weitaus meisten PHP-Entwickler setzen das Datenbanksystem MySQL ein.¹

PHP bietet im Vergleich zu anderen Programmiersprachen viele Vorteile. Als wichtigste Gründe für die Nutzung von PHP sind zu nennen:

- Es wurde zur Entwicklung von Internetanwendungen erschaffen.
- Es ermöglicht die einfache Entwicklung von Programmen.
- Es unterstützt verschiedene Plattformen.
- Es arbeitet sehr gut mit dem verbreiteten Apache Webserver zusammen, ...²

1.2 Grundlagen

Einbettung von PHP in HTML, Kommentare: `[kommentar.php]`³

Aufgabe:

Bringen Sie das angegebene Beispiel zum Laufen.

Hinweise:

- Der apache der Entwicklungsumgebung ist so konfiguriert, dass die URL `http://localhost/kurs` in den *workspace* (`~/workspace_php`) zeigt und nur noch mit Projekt- und Dateiname ergänzt werden muss.
- Eclipse enthält einen Debugger, der sich automatisch in den Start des PHP-Interpreters einhängt. Dies kann nützlich oder auch lästig sein. Gesteuert wird das über das (Nicht-) Vorhandensein der Datei `xdebug.ini` in der Konfiguration. Um auf einfache Weise das Debuggen ein- und ausschalten zu können, sind in den Umgebungen die Kommandos `debug-on` und `debug-off` installiert.

¹Theis: S. 18.

²Theis: S. 18.

³Alle Beispielprogramme können unter https://www.rheinwerk-verlag.de/einstieg-in-php-7-und-mysql_4644/ heruntergeladen werden.

Hinweis: Wenn der PHP-Interpreter startet, wird als allererstes die `php.ini` eingelesen. Diese Datei, gegliedert in Unterdateien bestimmt das Verhalten von PHP. Auf einem Linux-System finden sich üblicherweise zwei Konfigurationen:

- `/etc/php/VERSION/cli/php.ini` für den Aufruf als eigenständiges Programm
- `/etc/php/VERSION/apache2/php.ini` für den Aufruf aus Apache heraus

Mit dem Befehl `phpinfo()` kann die aktuelle Konfiguration ausgegeben werden.

Aufgabe:

Fügen Sie in das Beispiel `kommentar.php` das `phpinfo()` ein.

Innerhalb eines Programms können Informationen zur späteren Verwendung in Variablen gespeichert werden. Diese Variablen unterscheiden sich in ihren Datentypen. PHP unterstützt Datentypen für:⁴

- ganze Zahlen
- Zahlen mit Nachkommastellen (auch genannt: Fließkommazahlen)
- Zeichenketten (Strings)
- Felder (ein- und mehrdimensionale Felder von Variablen)
- Objekte

Variablennamen beginnen stets mit einem `$`: `[zahl_variable.php]`

Rechenoperationen für Zahlen:

Operator	Bedeutung
+	Addition
-	Subtraktion
*	Multiplikation
/	Division
%	Modulo-Operation: der Rest einer ganzzahligen Division: $7 \% 3 = 1$
**	Exponentialoperator ("hoch nehmen"): $3 ** 3 = 27$

Zeichenketten und Variablenexpansion: `[zeichenkette.php]`

Aufgabe:

Bringen Sie das Beispiel `zeichenkette.php` zum Laufen.

Konstanten: `[konstanten.php]`

Referenzen: `[referenz.php]`

Eingabeformular: `[eingabe.htm]`, ergibt dieses Formular:

⁴Theis: S. 25.

Auswertung des Formulars: [eingabe.php]

Pflicht-Eingabefelder können mit einem zusätzlichen `required`-Kennzeichen im `<input>`-Tag gekennzeichnet werden.

Aufgabe:

- Bringen Sie das Beispiel `eingabe.htm/.php` zum Laufen.
- Ändern Sie die Eingabefelder in Pflichtfelder.

1.3 Verzweigungen

Operatoren: Tabelle 1

Operator	Bedeutung	Geltungsbereich
<code>==</code>	gleich	Zahlen und Zeichenketten
<code>!=</code>	ungleich	Zahlen und Zeichenketten
<code>></code>	größer als	Zahlen
<code><</code>	kleiner als	Zahlen
<code>>=</code>	größer als oder gleich	Zahlen
<code><=</code>	kleiner als oder gleich	Zahlen

Tabelle 1: Vergleichsoperatoren in PHP⁵

Darüber hinaus können logische Ausdrücke mit Und (`&&`), Oder (`||`) oder Nicht (`!`) verknüpft werden.

Mehrfache Verzweigung: [src/if_mehrfach.php]

switch/case: [switch.php]

PHP hat einen etwas großzügigen Umgang mit Typumwandlungen. Gerade bei Vergleichen lässt sich „alles mit allem“ vergleichen. Um hier möglichen Fehlerquellen zu begegnen, gibt es die explizite Typumwandlung (`intval()`, `doubleval()`, `boolval()`) und den Vergleich ohne Typkonvertierung (`===`, `!==`). [umwandeln.php]

⁵Theis: S. 43.

[wahrheitswert.php]

Ternärer Operator: [ternaer.php]

Der Spaceship Operator (<=>) liefert

- den Wert 1, falls der erste Wert größer ist,
- oder den Wert -1, falls der zweite Wert größer ist,
- oder den Wert 0, falls beide Werte übereinstimmen.

[spaceship.php]

Undefinierte Variablen: [existenz.php]

Dynamische Typprüfung: [typ_pruefen.php]

Koaleszenzoperator: [koaleszenz.php]

Aufgabe:

Mit den folgenden Übungen soll Schritt für Schritt eine Büchereiverwaltung aufgebaut werden.

- Legen Sie ein Projekt *Bucherei* an.
- Erstellen Sie ein Formular `add_medium.html`, das folgende sechs Eingabefelder aufweist: Signatur, Autor, Titel, Medientyp, Seitenzahl, Spieldauer.
- Erstellen Sie eine Verarbeitungsdatei `add_medium.php`, die prüft, ob der Medientyp `B` (Buch) oder `c` (CD) ist.
 - Wenn ja, geben Sie die Daten aus. Bei einem Buch lassen Sie die Spieldauer weg, bei einer CD die Seitenzahl.
 - Wenn nein, geben Sie eine Fehlermeldung aus.

1.4 Schleifen

for-Schleife:

```
for ($i=0; $i<5; $i++)  
{  
    echo "Zeile $i\n";  
}
```

Aufgabe:

Analysieren Sie die Datei `u_for_schachtel.php` und bringen Sie diese zum Laufen.

while-Schleife:

```
# read ahead
$data = get_data();

# working loop
while(isset($data)) {
    echo "Data: $data";
    $data = get_data();
}
```

Die Prüfung erfolgt bereits vor dem ersten Eintritt in die Schleife. Wenn also überhaupt keine Daten vorhanden sind, wird die Schleife nie durchlaufen.

do-while-Schleife: [dowhile.php]

Schleifenabbruch: [break.php]

Schleifenfortsetzung: [continue.php]

Aufgabe:

Analysieren Sie die Datei `u_while.php` und bringen Sie diese zum Laufen.

1.5 Funktionen

Grundlagen

Funktionen ohne Parameter: [funktion_einfach.php]

Funktionen mit Parametern: [funktion_mehrere.php]

Funktion mit Rückgabewert: [funktion_rueckgabewert.php]

Optionale Parameter: [funktion_optional.php]

Benannte Parameter (PHP8): [funktion_benannt.php]

Aufgabe:

Analysieren Sie die Datei `u_funktion_rueckgabewert.php` und bringen Sie diese zum Laufen.

Call by value vs. *call by reference*: [src/call_value_reference.php]

Typhinweise: [funktion_typhinweise_weitere.php, funktion_rueckgabewert_nullable.php]

Die Möglichkeiten für Typhinweise, eingeführt mit PHP7, wurden mit PHP8 erweitert: Union types⁶.

⁶<https://php.watch/versions/8.0/union-types> (25.5.2021)

2 Dateien

2.1 Zugriffe auf Dateien und Verzeichnisse

Schreiben einer sequenziellen Datei: [6312/schreiben.php]

Lesen einer sequenziellen Datei: [src/sequentiell_lesen.php]

Aufgabe:

Ergänzen Sie die Büchereiverwaltung:

- Ergänzen Sie `add_medium.php` um eine Funktion `addMedium`, die
 - die sechs Datenfelder als Parameter übernimmt,
 - die Datei `/var/media/medien.csv` zum Anhängen öffnet,
 - die Daten kommasepariert anhängt.

Anmerkung: Falls das Verzeichnis `/var/media/` noch nicht existiert, muss es angelegt werden:

```
sudo mkdir /var/media/  
sudo chown www-data:www-data /var/media/
```

- Rufen sie diese Funktion in der Requestverarbeitung.
- Schreiben Sie ein Programm `show_media.php`, das den Inhalt der Datei im Browser ausgibt.

Beim Betrieb von Webanwendungen ist es oft nötig, dass die Anwendung Dateien schreibt. Da die Anwendungen oft unter einem speziellen Nutzer laufen, ist nicht gesichert, dass die Anwendung einem bestimmten Verzeichnis auch Schreibrechte hat. Soll also eine Anwendung eine Datei anlegen, ist es von Vorteil, wenn zunächst das Schreibrecht überprüft wird.

PHP stellt zur Analyse des Dateisystems folgende Funktionen zur Verfügung:

- `is_dir`: Prüft, ob der angegebene Dateiname ein Verzeichnis ist.
- `is_executable`: Prüft, ob der Dateiname ausführbar ist.
- `is_file`: Prüft, ob der Dateiname eine reguläre Datei ist.
- `is_link`: Prüft, ob der Dateiname ein symbolischer Link ist.
- `is_readable`: Prüft, ob eine Datei existiert und lesbar ist.
- `is_writable`, `is_writeable`: Prüft, ob in eine Datei/Verzeichnis geschrieben werden kann.

Der Inhalt eines Verzeichnisses lassen sich mit `opendir/readdir` lesen: [datei_verzeichnis.php]

Zum Löschen einer Datei dient der Befehl `unlink($filename)`.

Aufgabe:

Ergänzen Sie die Büchereiverwaltung:

- Ergänzen Sie die Funktion `addMedium` um folgende Prüfung:
 - Gibt es die Datei `/var/media/medien.csv` und ist die beschreibbar?
 - Falls es die Datei nicht gibt, ist das Verzeichnis `/var/media/` beschreibbar?
 - Falls beide Prüfungen scheitern, soll eine Fehlermeldung ausgegeben werden. Testen Sie den Code, indem Sie die Anwendung mit einem nicht-beschreibbaren Verzeichnis (z.B. `/usr/`) laufen lassen.
- Ergänzen Sie `show_media.php` um eine Prüfung, die feststellt, ob die Datei `/var/media/medien.csv` vorhanden und lesbar ist.

2.2 POST/REDIRECT/GET-Muster

Oben beschriebene Lösung ist alles andere als komfortabel:

Wenn Sie nach dem (versuchten) Anlegen eines Mediums bereits eine Seite weiter navigiert sind und zur Ausgabe des Mediums/Fehlermeldung zurück wollen, führen Sie (spätestens beim *reload*) ein erneutes Anlegen aus.

Dieses Verhalten wird mit dem POST/REDIRECT/GET-Muster verbessert: Nach einem POST erfolgt nicht direkt die Ausgabe des Ergebnisses, sondern ein Redirect auf eine Ausgabeseite, die einen reinen GET-Request darstellt.

Aufgabe:

Nach dem Aufruf von `addMedium` in `add_medium.php` machen Sie einen REDIRECT auf `show_media.php`.

Der REDIRECT erfolgt in PHP über folgende Anweisung:

```
header("Location: show_media.php");
```

Bei einem REDIRECT dürfen zuvor keine HTML-Anweisungen sein. Daher müssen aus `add_medium.php` alle HTML-Anweisungen entfernt werden.

2.3 Dateien auf den Server hochladen

Dateien auf den Server hochladen: `[6312/upload.htm/php]`

Anmerkung: Das Programm ist aufgrund von Lese-/Schreibrechten nicht direkt lauffähig.

Das Beispiel verwendet `copy` zum Kopieren von Dateien. Darüber hinaus gibt es auch `rename($quelle, $ziel)` um Umbenennen von Dateien.

Aufgabe:

Integrieren Sie `upload.htm/php` in die Büchereiverwaltung:

- `upload.htm` wird unverändert übernommen.
- `upload.php` soll:
 - prüfen, ob die Dateilänge größer 0 ist.
 - prüfen, ob bereits eine Datei `/var/media/medien.csv` existiert. Falls ja, soll diese umbenannt werden.
 - die hochgeladene Datei nach `/var/media/medien.csv` kopieren.
 - über einen `REDIRECT` auf `show_media.php` den Inhalt der Datei anzeigen.

3 Felder und Zeichenketten

Felder

Numerisch adressierte Felder: `[src/feld_numerisch.php]`

Adressierung mit Zeichenketten: `[feld_assoziativ.php]`

Zeichenketten

Definition von Zeichenketten

- Hochkomma (ohne Variablenexpansion)

```
$a = 'Hans';  
echo 'Das ist $a';    // Das ist $a
```
- Doppeltes Hochkomma (mit Variablenexpansion)

```
$a = "Hans";  
echo "Das ist $a";    // Das ist Hans
```
- HERE-Dokument ohne Variablenexpansion

```
$a = 'Hans';  
echo <<<'END'  
"Das" 'ist' $a  
END;  
// "Das" 'ist' $a
```
- HERE-Dokument mit Variablenexpansion

```

$a = "Hans";
echo <<<END
"Das" 'ist' $a
END;
// "Das" 'ist' Hans

```

Funktionen auf Zeichenketten:

- `strlen(STR)`: Länge messen
- `strtolower(STR)`: In Kleinbuchstaben umwandeln
- `strtoupper(STR)`: In Großbuchstaben umwandeln
- `ucfirst(STR)`: Ersten Buchstaben in Großbuchstaben umwandeln
- `ucwords(STR)`: Die ersten Buchstaben jedes Wortes in Großbuchstaben umwandeln
- `strrev(STR)`: Zeichenkette umdrehen
- `strtr(STR, "abc", "ABC")`: Einzelne Buchstaben ersetzen
- `str_replace("weg", "dazu", STR)`: Zeichenketten austauschen
- `str_contains(STR, "enthalten")`: Prüft, ob eine Teilzeichenkette enthalten ist (PHP8)
- `str_starts_with(STR, "start")`: Prüft, ob STR mit "start" beginnt (PHP8)
- `str_ends_with(STR, "end")`: Prüft, ob STR mit "end" endet (PHP8)

Anmerkung: Zum besseren Umgang mit UTF-8 kodierten Zeichenketten gibt es von vielen String-funktionen auch eine `mb_ (multi byte)`-Version (z.B. `mb_strlen`).

[6312/text_umwandlung.php / text_eigenschaft.php]

Umwandlung von/in Felder: [(6312/)text_feld.php]

Teilzeichenketten: [src/teilzeichenketten.php]

Suchen nach Position: [text_position.php]

Aufgabe:

Ergänzen Sie `add_medium.php`

- um eine Funktion `getMediumArr($sign)`, die
 - prüft, ob die Datei vorhanden ist. Falls nein, soll `null` zurückgegeben werden.
 - die Datei einliest
 - jede Zeile in ein Feld verwandelt und das erste Feldelement mit der übergebenen Signatur vergleicht.
 - Stimmt die Signatur überein, wird die Datei geschlossen und das Feld an den Aufrufer zurückgegeben

- Kommt die Funktion am Ende der Datei an, so wurde keine übereinstimmende Signatur gefunden und `null` an den Aufrufer zurückgegeben.
- Rufen sie diese Funktion in der Funktion `addMedium`: Falls die Signatur schon in der Datei ist, soll sie nicht erneut an die Datei angehängt werden. Das Programm terminiert (erst mal) mit einem `exit`.

4 Fehlerbehandlung

Zur Fehlerbehandlung gibt es verschiedene Ansätze:

- Rückgabewerte vom Typ `int`
- Rückgabewerte eines speziellen Fehlertyps
- Exceptions
- Eigene Routine zur Ermittlung des Fehlerstatus

Vorteil der Rückgabewerte:

- Einfach in der Behandlung, wenn an Ort und Stelle auf den Fehler reagiert werden kann.

Vorteil der Exceptions:

- Fehlerbehandlung stört nicht den logischen Programmfluss.

Empfehlung:

- Zu erwartende (oft fachliche) Fehler werden auf Rückgabewerte abgebildet.
- Unerwartete Fehler (oft technische Fehler, Logikfehler) werden über Exceptions behandelt.

Syntax für das Werfen und Fangen von Exceptions: `[exception_mit.php]`

Wichtig ist, dass Exceptions nicht bis zum Anwender durchschlagen dürfen, da sonst Implementierungsdetails preis gegeben werden und so das Programm angreifbar wird.

Ausnahmen, die nicht gefangen werden, führen zum Abbruch des Programms. Diese Fehler sind im Logfile des Apache (`/var/log/apache2/error.log`) zu finden. Sollen diese auch gefangen werden, so muss der Obertyp `Throwable` gefangen werden.

Aufgabe:

Ändern Sie das Beispiel `exception_mit.php` in folgender Weise ab:

- Entfernen Sie den `try/catch`-Block: Die Exceptions sind nun auf der Oberfläche sichtbar, falls das nicht mit `display_errors = Off` in der `php.ini` unterdrückt wird.
- Entfernen Sie die 0-Prüfung: Das Programm bricht ab (ab PHP 8).
- Ändern Sie den gefangenen Typ auf `Throwable`: Das Programm bricht nicht mehr ab.

Aufgabe:

Ergänzen Sie `add_medium.php`:

- Werfen Sie in `addMedium` eine Exception in dem Fall, dass sich die Datei nicht zum Schreiben öffnen lässt.
- In der Funktion `getMediumArr` soll für jede gelesene Zeile geprüft werden, ob es sechs Elemente sind (`count(...)`). Falls nein, soll eine Exception geworfen werden.
- Die Funktion `addMedium` soll einen `int`-Rückgabewert an den Aufrufer zurück geben. Definieren Sie dazu in einer eigenen Datei (`ReturnCodes.php`) die Konstanten. Das Einbinden von `ReturnCodes.php` erfolgt über `require_once`.
- Fügen Sie die entsprechenden `return`-Statements in `addMedium` ein.
- Werten Sie im Hauptteil von `add_medium.php` den Rückgabewert aus und fangen Sie die Exceptions.
- Legen Sie eine Datei `error_view.php` an, die in Abhängigkeit eines Errorcodes, der mit dem GET-Request übergeben wird⁷, eine entsprechende Ausgabe macht. Leiten Sie im Fehlerfall (Rückgabewert oder Exception) auf diese Seite weiter.

5 Modularisierung

In „richtigen“ Anwendungen wird der Code auf mehrere Dateien verteilt. Zum einen sollen die Dateien nicht zu groß werden, zum anderen soll eine Codeverdoppelung vermieden werden.

PHP hat mit `require_once` eine Funktion, die es ermöglicht, externe Definitionen exakt einmal einzulesen. Ist eine externe Datei bereits eingelesen worden, wird die Anweisung beim zweiten mal ignoriert.

Für die Strukturierung des Codes und damit die Aufteilung in Dateien kann man auf bewährte Muster (*design patterns*) zurückgreifen. Das wichtigste Muster im Web-Umfeld ist das MVC-Muster (*model-view-control*). Dabei wird die Bearbeitung von HTTP-Anfragen nach folgenden Gesichtspunkten zerlegt:

- *model*: Alle Operationen, die mit den Daten (dem Datenmodell) zu tun haben. In unserem Beispiel wäre das das Lesen und Schreiben in die Medien-Datei.
- *view*: Alles, was mit der Anzeige, also mit der Erstellung der HTML-Ausgabe zu tun hat. Nur im *view* dürfen HTML-Tags vorkommen.
- *control*: Dieser Teil der Software nimmt die HTTP-Anfragen entgegen, leitet die extrahierten Daten an das *model* weiter und ruft zum Schluss den passenden *view*.

Das MVC-Muster gibt es in vielen Varianten. In Abb. 1 ist der Nachrichtenfluss einer sehr einfachen Variante gezeigt:

⁷http://.../error_view.php?code=2

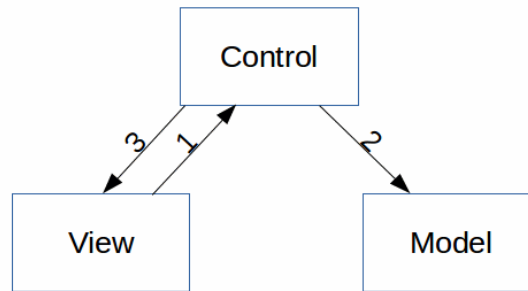


Abbildung 1: MVC-Muster

1. Von einem *view* aus wird eine neue Aktion angestoßen. Der Client wendet sich an den Server. Die Anfrage wird vom *controller* bearbeitet.
2. Dieser greift dazu auf das *model* zu. War es eine datenändernde Operation wird gemäß dem POST-REDIRECT-GET-Muster eine neue Anfrage an den *controller* ausgelöst.
3. Bei einer lesenden Operation werden die gelesenen Daten dem *view* übergeben. Dieser fügt die Daten in das HTML ein und schickt sie zurück an den Client.

Aufgabe:

Mit der folgenden Aufgabe soll der bereits bestehende Code in Module gemäß dem MVC-Muster gekennzeichnet werden.

- Entflechten Sie ggf. Ihre Dateien gemäß diesem Muster.
- Packen Sie alle *model*-Funktionen in eine Datei `Medienverwaltung.php`

6 Objektorientierung

Folgende Erfahrungen in der Softwareentwicklung haben zur Idee der Objektorientierung geführt:

- Strukturen sind eine sehr nützliche Sache, um Daten, die logisch zusammen gehören, zusammen zu verwalten.
- Wird eine Struktur im Speicher angelegt, wurde es in großen Programmen schnell unübersichtlich, wer diese Struktur für welchen Zweck gebraucht und wer Änderungen daran vornimmt.
- Unklar war oft, ab welchem Zeitpunkt welche Bestandteile einen gültigen Wert besitzen.

Vor diesem Hintergrund kam die Idee auf, die Zugriffe auf Strukturen (lesend, wie schreibend) zu kontrollieren. Der allgemeine Zugriff auf die Datenstruktur wurde also verboten, stattdessen wurden Funktionen geschaffen, über die auf die Daten zugegriffen werden konnte. Eine Datenstruktur mit den dazugehörigen Zugriffsfunktionen nennt sich *Klasse*.

Die Instanziierung einer Klasse bedeutet, einer Struktur einen konkreten Speicherbereich zuzuordnen. Nur eine instanzierte Struktur (=Objektinstanz) kann auch verwendet werden.

Bei einer Klasse sind im Gegensatz zu einer Struktur die internen Datenelemente von außen nicht zugreifbar. Alle Zugriffe erfolgen prozedural über öffentliche *Methoden*.

Auf der anderen Seite dienen Klassen oft als Container, um Funktionen, die in logischem Zusammenhang stehen, in einer gemeinsamen Einheit zusammenzufassen. Sie bilden ein „Funktionsmodul“. Funktionen einer Klasse, die nicht auf der „internen“ Datenstruktur arbeiten, nennen sich *statisch*. Zur Verwendung von statischen Funktionen muss zuvor keine Objektinstanz angelegt werden.

Real existierende Klasse befinden sich irgendwo im Spektrum zwischen „Funktionsmodul“ und „Datenverwaltungsmodul“.

6.1 Klassen

Klassendefinition: [oop_klasse.php (erster Teil)]

Anwenden einer Klasse: [oop_klasse.php (zweiter Teil)]

Konstruktor und Ausgabe: [oop_konstruktor_klassisch.php]

Constructor property promotion (PHP8): [oop_konstruktor.php]

Auf die klasseninternen Daten wird typischer Weise mit *getter* und *setter* zugegriffen:

```
function getGeschwindigkeit() {
    return $this->geschwindigkeit;
}
function setGeschwindigkeit($g) {
    $this->geschwindigkeit = $g;
}
```

Aufgabe:

- Bringen Sie das Beispiel 6312/oop_optional.php zum Laufen. Beachten Sie die optionalen Parameter im Konstruktor.
- Wenden Sie die *constructor property promotion* an.

Statische Elemente: [oop_statisch.php]

In PHP werden Objektinstanzen ausschließlich im Freispeicher angelegt und vom *garbage collector* wieder frei gegeben. Eine Variable im Programm stellt daher lediglich einen Verweis in den Freispeicher dar. Über eine Zuweisung wird nur der Verweis kopiert, nicht die Instanz selbst. Um die Instanz selbst zu kopieren stellt PHP die Funktion `clone` zur Verfügung.

Soll beim Kopieren einer Instanz mehr passieren, als dass der Speicher, den die internen Daten der Instanz einnehmen, verdoppelt wird, kann hierfür eine Methode `__clone` angelegt werden. [oop_kopie.php]

Aufgabe:

Bringen Sie das Beispiel `oop_kopie.php` zum Laufen.

Aufgabe:

Ergänzen Sie die Bibliotheksanwendung. Verwenden Sie dabei, wo möglich, Typhinweise. Können an einer Stelle mehrere Typen auftreten, können diese als *union* geschrieben werden (z.B. `Nutzer|string`).

- Legen Sie eine Datei `Benutzer.php` an.
 - Definieren Sie darin eine Klasse `Benutzer` mit den entsprechenden Datenfeldern, einem Konstruktor, der die Felder versorgt, eine `__toString()`-Methode, die die Daten kommasepariert als Zeichenkette zurückgibt.
 - Diese Klasse hat zusätzlich die Eigenschaft `Nutzernummer`, die beim Anlegen mit 0 belegt wird.
 - Legen Sie *getter* für die *members* an.
- Legen Sie eine Datei `Nutzerverwaltung.php` an, darin
 - eine Funktion `addNutzer`, die nichts weiter macht (Datenhaltung kommt später).
 - eine Funktion `getNutzer($id)`, die für IDs zwischen 1 und 10 eine Benutzerinstanz mit festen Werten erzeugt, sonst `null` zurückgibt. Die `Nutzernummer` soll mit im Objekt abgelegt werden.
- Legen Sie die Datei `suche.php` an, die
 - ein Formular für *Signatur* und *Nutzernummer* enthält. Dieses Formular ruft beim *Submit* sich selbst auf (`suche.php` als GET).
 - falls in der Anfrage der Parameter `nutzer_id` gesetzt und nicht leer ist, versucht über `getNutzer` einen Benutzer zu laden und auszugeben.
- Fügen Sie die Datei `add_user.html` hinzu, die Benutzerdaten (Name, Vorname, ...) abfragt und übermittelt (Vorlage: `add_medium.html`). `add_user.html` ruft beim *Submit* `add_user.php`.
- In `add_user.php`
 - werten Sie das Formular `add_user.html` aus, legen mit diesen Daten eine Instanz von `Benutzer` an und rufen `addNutzer` auf.
 - Geben Sie die Instanz auf einer HTML-Seite aus.

Beziehungen

Objekte können Beziehungen zu anderen Objekten haben. Diese können unterschiedlich ausgestaltet werden. Entweder wird nur ein passender Schlüssel zum anderen Objekt gespeichert oder

gleich eine Referenz auf das andere Objekt. Grundsätzlich haben beide Vorgehensweisen Vor- und Nachteile:

- Beim Anlegen geht es schneller, wenn vom referenzierten Objekt nur ein Schlüssel abgelegt werden muss.
- Beim Lesen geht es schneller, wenn das referenzierte Objekt bereits im Speicher vorliegt.

Da PHP keine getypte Sprache ist, können wir auch Objekte schreiben, die beides können: Im Konstruktor wird abgelegt, was übergeben wird: Schlüssel oder Objekt. Bei einem lesenden Zugriff wird geprüft, ob bereits eine Objektreferenz vorliegt. Falls nein, wird versucht über den Schlüssel ein entsprechendes Objekt zu erzeugen und zurückzugeben. Der Schlüssel kann dann durch die Objektreferenz ersetzt werden.

Zur Prüfung, ob eine Variable eine Objektreferenz eines bestimmten Typs ist, wird der Operator `instanceof` verwendet.

Aufgabe:

Schreiben Sie eine Klasse `Ausleihe`:

- Der Konstruktor hat als Parameter (damit die Klasse als Member) *Medium*, *Benutzer*. Beim Anlegen kann statt der Objekte auch nur der Schlüssel (`nutzernummer`, `signatur`) übergeben werden.
- In der Methode `getNutzer` wird geprüft, ob es sich beim Wert der Membervariable um eine Instanz von *Benutzer* handelt. Falls nein, wird über `getNutzer` eine erzeugt und zurückgegeben.
- `getNutzernummer()` prüft gleichfalls den Typ der Membervariable und gibt entweder die Nutzernummer direkt zurück oder liest sie aus dem Nutzer-Objekt.
- `getSignatur()` gibt die Signatur zurück.
- `getMedium` wird später implementiert und wirft jetzt erst mal eine Exception.

Verwendet wird die Klasse erst später.

ClassLoader

In PHP-Programmen ist es praktisch, Klassen automatisch laden zu lassen. Dazu kann am System ein *classloader* angemeldet werden, der dann gerufen wird, wenn eine bis dato unbekannte Klasse verwendet werden soll. Dazu muss:

- Eine Klasse mit statischer Methode definiert werden. Die statische Methode bekommt den Klassennamen als Parameter.
- Die Methode sucht nach der Datei, die die gewünschte Klasse definiert und lädt sie mit `include`.
- Klassen- und Methodennamen werden mit `spl_autoload_register(array(CLASS_NAME,METHOD))` dem System bekannt gegeben.

Aufgabe:

In `add_user.php` mussten Sie die Klassendefinition explizit laden.

- Schreiben Sie eine Klasse `ClassLoader` mit der statischen Methode `autoload($class_name)`, die aus dem aktuellen Verzeichnis die Datei "`$class_name.php`" lädt.
- Melden Sie am Ende der Datei, nach der Klassendefinition diese Klasse als *class loader* an.
- Laden Sie diese Klasse in `add_user.php`.
- `add_user.php` muss nun auch ohne den Import von `Benutzer.php` funktionieren.

Serialisierung

Um das POST/REDIRECT/GET-Muster implementieren zu können ist es oftmals nötig, den aktuellen Zustand einer Objektinstanz zwischenspeichern. Dazu dient `serialize` und `deserialize`. [T:282f, `oop_serial_*`]

Da sowohl die schreibende als auch die lesende Datei die Definition der Klasse `Fahrzeug` benötigt, wird die Definition der Klasse in eine eigene Datei ausgelagert und in beiden anderen Dateien mit `include` importiert.

Aufgabe:

Bringen Sie das Beispiel bestehend aus den Dateien `oop_serial_class.inc.php`, `oop_serial_put.php` und `oop_serial_get.php` zum Laufen.

6.2 Vererbung

Der Vererbungsmechanismus wird häufig angewendet, um bereits vorhandene Definitionen zu übernehmen. Sie erzeugen dadurch eine Hierarchie von miteinander verwandten Klassen. Diese ermöglichen die Darstellung von Objekten, die teilweise übereinstimmende sowie auch unterschiedliche Merkmale aufweisen.

Im folgenden Beispiel wird eine Klasse `Pkw` definiert, mit deren Hilfe die Eigenschaften und Methoden von Personenkraftwagen dargestellt werden sollen. Bei der Erzeugung bedient man sich der existierenden Klasse `Fahrzeug`, in der ein Teil der gewünschten Eigenschaften und Methoden bereits vorhanden ist. Bei der Klasse `Pkw` kommen noch einige Merkmale hinzu. Hierbei handelt es sich um eine spezialisierte Klasse – im Gegensatz zur allgemeinen Klasse `Fahrzeug`.

Von der Klasse `Pkw` aus gesehen ist die Klasse `Fahrzeug` eine Basisklasse. Von der Klasse `Fahrzeug` aus gesehen ist die Klasse `Pkw` eine abgeleitete Klasse.⁸

⁸Theis: S. 270f.

Definition einer Vererbungshierarchie: [src/oop_vererbung.php, 1. Teil]

Sollen auch in der abgeleiteten Klasse die Elemente der Mutterklasse zugreifbar sein, so wird die Sichtbarkeit `protected` (statt `private`) gewählt.

Mit `parent::` können der Konstruktor oder überschriebene Methoden der Basisklasse wiederverwendet werden.

Instanziierung von Objekten: [src/oop_vererbung.php, 2. Teil]

Aufgabe:

Ergänzen Sie die Bibliotheksanwendung:

- Legen Sie die Datei `Medium.php` an. Darin
 - definieren Sie eine Klasse `MediumBase`, sowie die abgeleiteten Klassen `Buch` und `CD`.
 - Verteilen Sie die Attribute sinnvoll auf die Klassen. Das Typkennzeichen wird nicht mehr benötigt.
 - Die Konstruktoren von `Buch` und `CD` nehmen vier Parameter.
 - Implementieren Sie auch eine `__toString()`-Methode.
- Passen Sie den Classloader so an, dass wenn eine der implementierten Klassen gesucht ist, `Medium.php` geladen wird.
- Schreiben Sie im Medienverwalter eine globale Fabrik-Funktion `createMedium`, die als Parameter ein Array mit den Elementen der csv-Datei übernimmt und die passende Instanz von `CD` oder `Buch` zurückgibt.
- Ergänzen Sie `Medienverwaltung.php` um die Funktion `getMedium($signatur)`, die `getMediumArr` aufruft und, falls ein Medium gefunden wurde über `createMedium` eine passende Instanz zurückgibt.
- Implementieren Sie nun in der Klasse `Ausleihe` die Methode `getMedium()`, die, wenn nur eine Signatur vorliegt, mit Hilfe dieser Funktion ein Medium holt und zurückgibt.
- Ergänzen Sie die Datei `suche.php`:
 - Falls in der Anfrage der Parameter `signatur` gesetzt und nicht leer ist, wird versucht über `getMedium` ein Medium zu laden und auszugeben.
 - falls Medium und Benutzer geladen werden konnten, wird
 - * eine `Ausleihe` instanziiert, diese serialisiert und in einer Datei speichert,
 - * der Ausgabe ein Formular anhängt, das einen `submit`-Knopf mit der Aufschrift *Ausleihen* hat und einen `POST`-Request auf `ausleihen.php` hervorruft.
- Legen Sie die Datei `ausleihen.php` an, die das Objekt deserialisiert und ausgibt.

6.3 Abstrakte Klassen, Schnittstellen und Traits

Da das Anlegen einer Instanz der Klasse `MediumBase` keinen Sinn macht, kann so einem Fall eine *abstrakte Klasse* definiert werden. Dazu wird mindestens eine Methode der Basisklasse als `abstract` gekennzeichnet:⁹

```
abstract class Animal
{
    public $name;
    public $age;

    public function Describe()
    {
        return $this->name . ", " . $this->age . " years old";
    }

    abstract public function Greet();
}

class Dog extends Animal
{
    public function Greet()
    {
        return "Woof!";
    }

    public function Describe()
    {
        return parent::Describe() . ", and I'm a dog!";
    }
}
```

Die Definition einer abstrakten Methode zwingt alle abgeleiteten Klassen diese zu implementieren.

Gibt eine Klasse nur „Implementierungsverpflichtungen“ in Form von abstrakten Methoden vor, nennt man das eine *Schnittstelle*. PHP unterstützt die Einfachvererbung, d.h. eine abgeleitete Klasse kann von einer Basisklasse erben, aber beliebig viele Schnittstellen implementieren.

Definition von Schnittstellen: [T:279, `oop_schnittstelle.php`]

Traits (Eigenschaft, Merkmal) bieten eine alternative Methode der Wiederverwendung: [`src/traits.php`]

Während die Vererbung fachlich motiviert sein sollte, können mit *traits* technische Aspekte ergänzt werden. So könnte in unserem Beispiel das Speichern (in der Datei) über ein *trait* realisiert werden.

Aufgabe:

Schreiben Sie einen Trait `Tspeichern.php` mit der Methode `speichern()`.

- `speichern()` hängt eine Datenzeile an eine bestehende Datei an.

⁹<http://www.php5-tutorial.com/classes/abstract-classes> (27.3.2018)

- Zur Erzeugung der kommasparierten Daten muss jede Klasse `format()` implementieren.
- Jede Klasse muss die Konstante `FNAME` mit dem Dateipfad definieren.
- Fügen Sie den Trait den fachlichen Klassen hinzu und testen Sie den Code.

7 Weitere Themen

7.1 Namensräume

Namensräume (englisch: *namespaces*) ermöglichen Ihnen, Elemente gleichen Namens zu definieren, zu nutzen und eindeutig zuzuordnen. So wie Sie Dateien gleichen Namens in unterschiedlichen Verzeichnissen auf Ihrer Festplatte haben dürfen, so dürfen Sie Klassen, Funktionen und Konstanten gleichen Namens in unterschiedlichen Namensräumen verwenden.¹⁰

Definition und Verwendung: `[src/namespace.php]`

Vorsicht: Der derzeitige Interpreter akzeptiert (erste) Namensraumangaben nur am Anfang einer Datei. Dabei stören ihn vorgestellte BOM-Bytes¹¹.

Aufgabe:

Ergänzen Sie die Büchereianwendung:

- Packen Sie die Klassendefinition in `Medium.php` in einen Namespace `Medium`.
- Bauen Sie `ClassLoader` so um, dass, wenn ein Namensraum vorhanden ist, eine gleichnamige PHP-Datei geladen wird.

7.2 Datum und Zeit

`time()` ermittelt die Sekunden seit 1.1.1970:

```
$jetzt = time();  
echo "<p>Sekunden seit 01.01.1970: $jetzt</p>";
```

`date()`, `strftime()`¹² formatiert einen Zeitstempel: `[zeit_format.php]`

`mktime()` erzeugt einen Zeitstempel.

```
$dz = mktime(15, 32, 55, 03, 31, 2016);
```

Zeitstempel verändern: `[zeit_relativ.php]`

¹⁰Theis: S. 279

¹¹byte order mark

¹²deprecated seit PHP 8.1

Alternativ zum Zeitstempel lässt sich in PHP mit dem `DateTime`-Objekt arbeiten: `DateTime`¹³

Aufgabe:

Bringen Sie das Beispiel `zeit_feiertag.php` zum Laufen.

Anmerkung: Wird der Zeitstempel in 4 Byte abgelegt, reicht der adressierbare Zeitraum bis zum 19.1.2038.

Aufgabe:

Ergänzen Sie die Büchereianwendung:

Im Konstruktor von `Ausleihe` sollen zusätzlich zwei Membervariablen angelegt werden:

- `ausleihe_von` wird mit der aktuellen Zeit belegt.
- `ausleihe_bis` liegt vier Wochen in der Zukunft.

Zur Realisierung können Sie entweder den Zeitstempel oder das `DateTime`-Objekt verwenden.

7.3 Sessions und Cookies

In der vorigen Aufgabe existiert noch ein grundlegendes Problem:

Eine Webanwendung, die Daten in eine Datei speichert und diese in einem Folge-request wieder ausliest, ist nicht mehrbenutzerfähig. Das grundsätzliche Problem einer serverseitigen Dateiablage ist die Frage, wann die Daten, im Falle eines ausbleibenden Folge-requests wieder gelöscht werden dürfen. Hier helfen die Konzepte *session* und *cookie*. Bei der Session erfolgt eine benutzerindividuelle serverseitige Ablage, beim Cookie auf Anwenderseite. Da Sessions serverseitig gespeichert werden, haben sie eine relative kurze Lebensdauer. Cookies können über einen längeren Zeitraum gespeichert werden.

Session

Eine Session muss explizit begonnen werden (`session_start()`). Sie endet entweder mit dem Schließen des Browserfensters durch den Benutzer oder indem sie durch das Programm explizit geschlossen wird (`session_destroy()`). Die Funktion `session_start()` muss auf jeder Seite, die zu einer Session gehört, aufgerufen werden. Diese Funktion beginnt entweder eine neue Session oder nimmt eine vorhandene Session wieder auf. Die Daten einer Session werden im Array `$_SESSION` gespeichert. [`sc_zae_hler.php`]

Ein typisches Beispiel für ein Session-Management ist das Verwalten eines Logins, so dieser das Schließen des Browsers nicht überleben soll (Internet-Banking).

¹³<https://www.php.net/manual/de/class.datetime.php> (21.2.2024)

Aufgabe:

- Bringen Sie das Beispiel bestehend aus den Dateien 6312/sc_schutz_a.php, 6312/sc_schutz_b.php und 6312/sc_schutz_c.php
- Testen Sie den Code auch mit verschiedenen Browsertabs.
- Testen Sie den Code auch mit dem „Privaten Fenster“ des Firefox.

Cookies

Falls Daten über einen längeren Zeitraum aufbewahrt werden sollen, z. B. über mehrere Tage oder ein Jahr, können Sie Cookies verwenden. Dabei handelt es sich um kleine Dateien auf dem Rechner des Benutzers.

Zum Lesen und Schreiben von Cookies stehen das Array `$_COOKIE` und die Funktion `set_cookie` zur Verfügung. `set_cookie` hat folgende Parameter:

- Name des Cookies: Unter diesem Namen kann es später aus `$_COOKIE` wieder ausgelesen werden.
- Wert: Zu speichernde Daten
- Ablaufdatum des Cookies
- Domain, die das Cookie speichert (opt.)
- Pfad der Anwendung (opt.), dient der Eindeutigkeit des Namens

Zum Löschen eines Cookies wird *Wert* auf `null` gesetzt.

Anmerkung zu Cookies:

[...] Freilich sind Cookie-IDs als solche kein personenbezogenes Datum - dazu werden sie erst, wenn diese mit Informationen verknüpft werden, die einen Personenbezug herstellen. Nach Art. 5 Abs. 3 sind jedoch „klare und umfassende Informationen insbesondere über die Zwecke der Verarbeitung“ sowie ein Hinweis auf das Recht zur Verweigerung der Verarbeitung erforderlich, soweit nicht ausnahmsweise „der alleinige Zweck die Durchführung oder Erleichterung der Übertragung einer Nachricht über ein elektronisches Kommunikationsnetz ist“ oder „dies unbedingt erforderlich ist, um einen vom Teilnehmer oder Nutzer ausdrücklich gewünschten Dienst der Informationsgesellschaft zur Verfügung zu stellen“. Damit sind rein technische Cookies auch ohne Einwilligung zulässig.¹⁴

Aufgabe:

Bringen Sie das Beispiel `sc_besuch.php` zum Laufen.

¹⁴<https://www.bundesanzeiger-verlag.de/betrifft-unternehmen/compliance-sicherheit/aktuelles/news-details-compliance/artikel/eugh-zur-eprivacy-richtlinie-cookies-erfordern-aktive-gesonderte-und-ausdrueckliche-einwilligung-33293.html> (6.11.2019)

Aufgabe:

Jetzt werden noch Session und Cookie in die Büchereianwendung eingebaut:

- In `add_user.php` soll im Gutfall eine Session geöffnet werden und die Benutzer-Instanz statt in eine Datei in die Session geschrieben werden.
- In `benutzer_anzeigen.php` (neue Datei) soll diese wieder aus der Session gelesen werden. Denken Sie auch an eine Fehlerbehandlung, falls die Session nicht vorhanden ist.
- Prüfen Sie bei einem GET-Request auf `suche.php`, ob ein Cookie namens „Suchdaten“ vorhanden ist. Falls ja, übernehmen Sie die Daten für Signatur und Nutzernummer aus dem Cookie in das Formular.
- Bei einem POST-Request auf `suche.php` schreiben Sie ein Cookie mit den übergebenen Daten. Die Daten werden gleichfalls in das Formular übernommen.

7.4 Reguläre Ausdrücke

In der Programmiersprache Perl sind Reguläre Ausdrücke fester Sprachbestandteil. In PHP sind sie über entsprechende Funktionen nutzbar. PHP implementiert die „Perl-kompatibel“ (PCRE: *Perl compatible regular expression*).

Regex-Tester¹⁵

PHP lässt für die Begrenzung des Ausdrucks auch andere Zeichen als `/` zu: Delimiters¹⁶

Einige Zeichen haben innerhalb regulärer Ausdrücke eine spezielle Bedeutung und müssen bei der Verwendung maskiert werden. Dies kann mit der Funktion `preg-quote`¹⁷ erfolgen.

<code>/</code>	leitet einen regulären Ausdruck ein
<code>.</code>	steht für ein beliebiges Zeichen
<code>+ * ? {}</code>	kennzeichnen Kardinalitäten
<code>^ \$</code>	verankern ein Suchmuster am Anfang/Ende
<code> </code>	kennzeichnet eine Auswahl
<code>()</code>	gruppiert Treffer
<code>[]</code>	definiert Buchstabenmengen

Suchmuster können mit Modifikatoren¹⁸ angepasst werden.

Wichtige PHP-Funktionen:

- Suche nach Treffern in einem String: `preg-match`¹⁹
- Suche nach Treffer in einem Array: `preg-grep`²⁰

¹⁵<http://regex101.com> (21.2.2020)

¹⁶<https://www.php.net/manual/de/regexp.reference.delimiters.php> (12.11.2019)

¹⁷<https://www.php.net/manual/de/function.preg-quote.php> (12.11.2019)

¹⁸<https://www.php.net/manual/de/reference.pcre.pattern.modifiers.php> (12.11.2019)

¹⁹<https://www.php.net/manual/en/function.preg-match.php> (12.11.2019)

²⁰<https://www.php.net/manual/de/function.preg-grep.php> (12.11.2019)

- Suchen und Ersetzen: preg-replace²¹

Reguläre Ausdrücke können auch gleich im HTML-Formular eingesetzt werden:²²

```
<form action="/action_page.php">
  <label for="country_code">Country code:</label>
  <input type="text" id="country_code" name="country_code"
    pattern="[A-Za-z]{3}" title="Three letter country code"><br><br>
  <input type="submit">
</form>
```

Aufgabe:

Ergänzen Sie die Büchereianwendung um eine Parameterprüfung.

- Ergänzen Sie Medienverwaltung.php um eine Funktion checkMedium(), die die 6 Parameter übernimmt und mit regulären Ausdrücken prüft. Die Funktion soll true oder einen geeigneten Fehlertext zurückliefern.
- Rufen Sie die Funktion in add_medium.php. Falls die Funktion nicht true zurückliefert, rufen Sie über einen *redirect* den Errorview auf (INVALID_PARAMETER) und geben Sie den Fehlertext als weiteren Parameter mit.
- Ergänzen Sie den Errorview so, dass er bei Vorhandensein eines zusätzlichen Fehlertextes diesen auch anzeigt.

7.5 Socketkommunikation

Über Sockets werden unter UNIX/in Internet die Kommunikation zwischen Prozessen abgebildet. Grundsätzlich lässt sich zwischen Kommunikation mit (TCP) und ohne (UDP) Verbindung unterscheiden.

[src/server.php, src/client.php]

Aufgabe:

Bringen Sie das gezeigte Beispiel zum Laufen.

²¹<https://www.php.net/manual/de/function.preg-replace.php> (12.11.2019)

²²https://www.w3schools.com/TAGS/att_input_pattern.asp (2.7.2024)

Sollen über PHP E-Mails versendet werden, stehen zwei Möglichkeiten zur Verfügung:

- Funktion `mail`²³
Vorteil: Einfach zu Bedienen
Nachteil: Windows-Implementierung von mangelnder Qualität, Anhänge müssen selbst encodiert werden.
- Implementierung mit eigener Socket-Kommunikation.
Nachteil: Mail-Protokoll muss genau eingehalten werden.
Es gibt Bibliotheken, die die Socket-Kommunikation implementieren und dem Nutzer eine angenehme Schnittstelle zu Verfügung stellen, z.B. `phpmailer/class.phpmailer.php`.²⁴ Die Kommunikation findet hier in der Datei `phpmailer/class.smtp.php` statt. Die Anwendung findet sich in `mail_pm.php`.

7.6 Grafik

PHP gibt Ihnen mit der GD-Bibliothek eine Möglichkeit, eigene Grafiken zu erstellen. Diese Grafiken lassen sich unmittelbar anzeigen oder als Grafikdatei speichern. Außerdem können Sie vorhandene Grafiken laden und verändern.²⁵

Beispiel: Darstellung von Werten: `[im_text_intern.php]`

Anmerkung: Um das Programm laufen lassen zu können, muss das Bild in einem Verzeichnis gespeichert werden, in dem der Apache Zugriff hat.

Anmerkung: Es können auch *true type fonts* verwendet werden. Diese müssen dann in einem geeigneten Verzeichnis bereit gestellt werden. Die Pfadangabe muss entweder mit einem Schrägstrich (absolut) oder mit einem Punkt (relativ) beginnen. `[im_text_ttf.php]`

Beispiel: CAPTCHA: `[captcha_a.php, captcha_a.php]`

8 Datenbanken mit MySQL

MySQL ist die Datenbank, die das „Internet zum Laufen brachte“. In den Anfängen realisierte MySQL eine extrem lesestarke Datenbank mit Volltextsuche. Ein geordnetes transaktionales Verhalten wurde erst später eingebaut. Heute ist MySQL mit seinen Forks MariaDB und PerconaDB eine voll funktionsfähige Datenbank, die vermehrt auch in geschäftskritischen Bereichen eingesetzt wird.

MySQL und PHP traten von Anfang an als Team auf, was sich heute noch in der extrem guten Einbindung zeigt. LAMP (*Linux-Apache-PHP-MySQL*) galt lange Zeit als *die* Standardarchitektur für Web-Anwendungen.

²³<https://www.php.net/manual/de/function.mail.php> (13.11.2019)

²⁴Theis

²⁵Theis: S. 495.

8.1 MySQL und phpMyAdmin

Ein nützliches Werkzeug zum Verwalten einer MySQL-Datenbank ist phpMyAdmin. Auf einem Linux-Server kann er einfach mit

```
sudo apt-get install phpmyadmin
```

installiert werden.

Aufgabe:

Für die nächsten Aufgaben können Sie entweder den phpMyAdmin oder den Kommandozeilenclient `mysql` verwenden.

Für die weiteren Übungen wird eine Datenbank für die Nutzer verwendet.

- Legen Sie die Datenbank `bucherei` mit der Tabelle `nutzer` an. `nutzer` enthalte neben einer numerischen ID-Spalte Spalten für Name, Vorname, Adressdaten (Anleitung s. `src/bucherei.txt`).
- Legen Sie ein paar Datensätze an.

8.2 PHP und MySQL

Für die Verbindung zur MySQL-Datenbank stehen in PHP zwei Schnittstellen zur Verfügung: `mysqli` und `PDO`. `mysqli` ist besser auf die Datenbank abgestimmt, `PDO` hat den Vorteil auch andere Datenbanken wie `postgreSQL` zu unterstützen. Werden nicht gerade sehr spezielle MySQL-Eigenschaften benötigt, wird die Verwendung von `PDO` empfohlen.²⁶

Verbindung aufnehmen, Datensätze anzeigen: `[db_anzeigen.php (mysqli), pdo_zugriff.php (PDO)]`

Aufgabe:

- Legen Sie die Datei `Nutzerverwaltung.php` an.
- Schreiben Sie darin eine Funktion `alleNutzer()`, die alle Nutzer aus der DB liest und als Array von Nutzerobjekten zurückgibt.
- Ergänzen Sie die Büchereianwendung um `show_user.php`, die `alleNutzer()` aufruft und die Benutzer dann ausgibt.

Anmerkung zu `db_anzeigen.php`: Diese Art der Web-Programmierung, die kunterbunt Request-Bearbeitung, Datenbankzugriffe und HTML-Erzeugung mischt, ist (leider) weit verbreitet. Eine klare Trennung von Datenwelt (*model*) und Anzeige (*view*) schafft aber robusteren Code, der auch weniger Anfällig auf Sicherheitsangriffe ist.

Datensätze erzeugen: `[db_erzeugen.php]`

²⁶<https://websitebeaver.com/php-pdo-vs-mysqli> (16.2.2024)

Aufgabe:

Implementieren Sie die Datenhaltung für die Funktion „Nutzer anlegen“:

- Die fachliche Implementierung in `Nutzerverwaltung.php` in der Funktion `addNutzer()`.
- Implementieren Sie `getNutzer()` nun mit einem Datenbankzugriff.

Soll bei komplexeren Bearbeitungen vermieden werden, dass für jeden Datenbankzugriff die Verbindung zur Datenbank geöffnet und geschlossen wird, benötigen wir eine Steuerung von Zugriff und Transaktion.

- Legen Sie eine Datei `DBMysql.php` an, die die gleichnamige Klasse enthält. Diese Klasse soll

- im Konstruktor die Verbindung zur DB öffnen und als *member* speichern.
- die Verbindung über einen *getter* zugreifbar machen.
- eine Funktion `close()` haben, die die Verbindung schließt.
- als *singleton* verwendet werden können:

```
static $instance = NULL;
static function getInstance() {
    if (!isset($instance)) {
        $instance = new DBMysql();
    }
    return $instance;
}
```

- Verwenden Sie bei den DB-Zugriffen diese Klasse.

```
$con = DBMysql::getInstance()->getConnection();
```

- Schließen Sie im *controller* am Ende einer Datenoperation die Verbindung.
- Der *controller* ist auch für die Transaktionssteuerung zuständig. Die Befehle hierfür lauten:

```
DBMysql::getInstance()->begin_TA();
DBMysql::getInstance()->commit();
DBMysql::getInstance()->rollback();
```

- Alternativ können Sie die Musterdatei `src/DBMysql.php` in Ihre Anwendung einbauen.

9 XML

PHP bietet zwei Implementierungen für den Umgang mit XML-Dateien:

- SimpleXML
- DOMDocument

Hier die Vor- und Nachteile²⁷:

- SimpleXML
 - is for simple XML and/or simple UseCases
 - limited API to work with nodes (e.g. cannot program to an interface that much)
 - all nodes are of the same kind (element node is the same as attribute node)
 - nodes are magically accessible, e.g. `$root->foo->bar['attribute']`
- DOM
 - is for any XML UseCase you might have
 - is an implementation of the W3C DOM API (found implemented in many languages)
 - differentiates between various Node Types (more control)
 - much more verbose due to explicit API (can code to an interface)
 - can parse broken HTML
 - allows you to use PHP functions in XPath queries

9.1 SimpleXML

Lesen von XML-Dateien

Einlesen und auswerten eines XML-Dokuments: `[xml_sammlung.php]`

Schreiben von XML-Dateien

`addChild`²⁸ Anmerkung: Das `include "example.php"` definiert die Variable `$xmlstr` mit dem Inhalt:

```
<?xml version="1.0" standalone="yes"?>
<movies/>
```

Aufgabe:

Erzeugen Sie ein XML-Dokument, das wie `src/fahrzeuge.xml` aussieht.

²⁷<https://stackoverflow.com/questions/4803063/whats-the-difference-between-phps-dom-and-simplexml-extensions> (13.11.2019)

²⁸<https://www.php.net/manual/de/simplexmlelement.addchild.php> (13.11.2019)

XPath

XPath²⁹ ist der wahlfreie Zugriff auf XML-Knoten.

Aufgabe:

Extrahieren Sie aus `xml_sammlung.xml` alle Reifendaten.

9.2 DOMDocument

Das Arbeiten mit XML-Dokumenten über `DOMDocument` ist zwar für den Anfang etwas schwieriger, bietet aber Vorteile: Zum einen ist es standardisiert und kann in ähnlicher Weise auch in anderen Programmiersprachen implementiert werden. Zum anderen ist der Funktionsumfang größer.

Für den einfachen Umgang mit `DOMDocument` gibt es die Dateien `XmlExtractor.php` und `XmlFormatter.php`, deren Basisdefinitionen in `InputExtractor.php` und `OutputFormatter.php` zu finden sind.

Hier ein Beispiel zur Verwendung: `[src/xml_dom_appl.php]`

Aufgabe:

Erzeugen Sie ein XML-Dokument, das wie `src/fahrzeuge.xml` aussieht und lesen Sie es wieder ein.

Aufgabe:

Die Ausleihvorgänge sollen in einer XML-Datei abgelegt werden. Eine einzelne Ausleihe habe folgendes Format:

```
<ausleihen>
<ausleihe>
    <from>04.05.2020</from>
    <to>04.06.2020</to>
    <medium signatur="A001"/>
    <user id="23"/>
</ausleihe>
</ausleihen>
```

- Legen Sie die Datei `Ausleiheverwaltung.php` an. Darin eine Klasse
 - die im Konstruktor die XML-Datei lädt (falls vorhanden) und die Ausleihe-Objekte in einem assoziativen Array (Schlüssel ist die Signatur) ablegt.
 - `getAusleihe($signatur)`, die ein Ausleiheobjekt oder `false` zurückgibt,
 - `addAusleihe($ausleihe)`, die dem Ausleihe-Array einen weiteren Eintrag hinzufügt,
 - `save()`, die das XML wieder erzeugt und schreibt.

²⁹<https://www.php.net/manual/de/simplexmlelement.xpath.php> (13.11.2019)

- Ergänzen Sie in `ausleihen.php` das Anlegen der Ausleihe. Prüfen Sie über `getAusleihe`, ob das Medium nicht schon verliehen ist. Vergessen Sie nicht am Ende mit `save()` die Änderung zu speichern.

Hinweis: Zum Formatieren von Datumsangaben beachten Sie die Datei `zeit_format.php`.

10 Ajax

Ajax steht für Asynchronous JavaScript and XML. Diese Technik bietet eine asynchrone Datenübertragung zwischen Browser und Webserver. Das ermöglicht Ihnen, Teile einer Internetseite zu ändern, ohne die gesamte Seite neu erstellen und übermitteln zu müssen. Auf diese Weise reduziert sich der Entwicklungsaufwand, verringert sich der Netzverkehr und werden die Internetseiten schneller und flexibler. Die Ajax-Technik wird seit Jahren erfolgreich im Web eingesetzt und ist ein selbstverständlicher Bestandteil vieler Internetseiten.³⁰

Anmerkung: Im Firefox lässt sich mit F12 ein Javaskript-Debugger starten.

10.1 Hallo Ajax

Webseite, die über Ajax weitere Inhalte anfordert: [T:477, `ajax_hallo.htm`]

PHP-Programm, das die angeforderten Daten sendet: [T:478, `ajax_hallo.php`]

10.2 Parameter senden

Im folgenden Beispiel soll nun ein Parameter (hier: MA-Id) mit an Server übergeben werden.

Webseite, die über Ajax weitere Inhalte zu einem bestimmten Mitarbeiter anfordert: [T:480, `ajax_parameter.htm`]

PHP-Programm, das die angeforderten Daten sendet: [T:481, `ajax_parameter.php`]

10.3 XML-Datei lesen

Bei den bisherigen Beispielen wurde nur Text eingefügt/ersetzt. Sollen Daten aus größeren Strukturen über Ajax in die bestehende Internetseite eingefügt werden, erfolgt dies über XML.

(Statisches) XML: [T:485, `ajax_xml_einzel.xml`]

HTML, das das XML anfordert und auswertet: [T:486, `ajax_xml_einzel.htm`]

Beispiel mit dynamischer Suche: [`src/suche.html`, `output.php`]

³⁰Theis: S. 473.

Aufgabe:

Nun wird die AJAX-Technik in das Projekt `Bucherei` integriert:

In `suche.php` sollen die Trefferkandidaten bei der Mediensuche angezeigt werden.

11 PHP 8

Like I said time and time again: PHP isn't the same language it was ten years ago, and we're very thankful for that. It's a fast and reliable language, used to build large applications at scale. So let's discuss some of the most notable changes of PHP in the last year, to the language and the community.³¹

Mit PHP 8 kamen folgende Neuerungen in die Sprache:

- PHP JIT

Ein *just in timer compiler* (JIT) compiliert den PHP-Code in die Maschinensprache des Zielsystems und speichert das Compilat in einem Zwischenspeicher. Wird dasselbe Modul ein zweites mal aufgerufen, kann direkt das Compilat aufgerufen werden, was deutlich performanter ist. Dies bringt dort Vorteile, wo wie einem Webserver oft derselbe Code auszuführen ist.

Um den JIT-Compiler zu aktivieren müssen in der `php.ini` (oder deren Unterdateien) folgende Einträge gemacht werden:³²

```
opcache.enable=1          # für Server
opcache.enable_cli=1      # für CLI
opcache.jit_buffer_size=100M
```

Die Einstellungen in der `php.ini` können (wie immer) mit dem Aufruf `phpinfo()` (nach einem Neustart von Apache) überprüft werden.

Mit PHP 7 wurde der *opcache* eingeführt. Durch die Zwischenspeicherung von maschinenunabhängigen Zwischencode konnte die PHP-Performance signifikant verbessert werden. Der JIT hingegen speichert Maschinencode, was zu einer weiteren Performanceverbesserung führen sollte.³³

Zur Erstellung von *benchmarks* können die Skripte von

<https://github.com/SergiX44/php-benchmark-script>³⁴ verwendet werden.

- Constructor Property Promotion → 6.1
- Benannte Parameter → 1.5
- Union Types → 1.5

³¹<https://stitcher.io/blog/php-in-2021>

³²<https://stitcher.io/blog/php-8-jit-setup> (25.5.2021)

³³<https://www.php-einfach.de/2020/09/php-8-neuer-performance-turbo-durch-jit-compiler/> (2.11.2023)

³⁴(6.6.2024)

- Weitere Funktionen für Zeichenketten → 3
- Match Kontrollstruktur → Match or switch?³⁵
- Nullsafe Operator → Einleitung³⁶
- Weak maps

`Map`³⁷ ist eine Klasse ähnlich dem assoziativen Array. Im Gegensatz zu diesem ist es bei einer `Map` möglich auch Objekte als Schlüssel zu verwenden. Damit entsteht aber ein neues Problem: Wird zu einem Objekt ein Eintrag gemacht, erhöht dies den Referenzzähler und das Objekt kann nie wieder verschwinden.

Dieses Problem wird durch den Einsatz einer *weak map* gelöst. → The WeakMap class³⁸

An einigen Stellen bricht PHP8 auch mit der Abwärtskompatibilität:

- `0 == "foo"` evaluiert neu auf `false`.
- `.-` Operator wird neu nach `+/-` ausgewertet.
- *Namespaces* dürfen keine Leerzeichen mehr enthalten.
- Kleinere Änderungen im Reflection-API
- Änderungen in den Error-Leveln

Aufgabe:

- Erstellen Sie *benchmarks* mit/ohne OP-Cache mit/ohne JIT.
- Stellen Sie die `switch`-Struktur in `error_view.php` auf `match` um.
- Dechiffrieren Sie das `return`-Statement in Weak maps³⁹ ("??" ist der Coalesce-Operator).

12 Quellen

- Theis, Thomas Einstieg in PHP7 und MySQL. Ideal für Programmierneinsteiger, 12. Auflage, 2017
- [T] Einstieg in PHP7 und MySQL. Ideal für Programmierneinsteiger, 13. Auflage, 2018

³⁵<https://stitcher.io/blog/php-8-match-or-switch> (26.5.2021)

³⁶<https://stitcher.io/blog/php-8-nullsafe-operator> (26.5.2021)

³⁷<https://www.php.net/manual/en/class.ds-map.php>

³⁸<https://www.php.net/manual/en/class.weakmap.php> (26.5.2021)

³⁹<https://stitcher.io/blog/new-in-php-8> (26.5.2021)

Vielen Dank für Ihre Aufmerksamkeit

Ihr Referent und das Team von
IT-Schulungen.com

Fon +49 (0) 911 650 08 - 30

Fax +49 (0) 911 650 08 - 399

Mail info@it-schulungen.com

Web www.it-schulungen.com

Education Center der New Elements GmbH
Thurn-und-Taxis-Straße 10
90411 Nürnberg

www.newelements.de



New Elements GmbH | IT-
Schulungen.com