

Prüfung K. Baltes für
Einführung in Web-bezogene Sprachen
WS 2024/2025

Aufgabe 1:

Es soll eine Web-Präsenz zur Verwaltung von Büchersammlungen erstellt werden. Dafür können vorgegebene Templates verwendet werden, die die HTML- bzw. CSS-Anteile der Seiten beinhalten.

Damit beschränkt sich die Aufgabe auf die PHP-Programmierung der Web-Präsenz. Hierzu ist ein Hauptprogramm in einer Datei `index.php` zu erstellen. Desweiteren sollen drei PHP-Klassen implementiert werden: `Buch`, `Sachbuch` und `Buchliste`.

Diese werden im folgenden beschrieben:

Klasse Buch

Die Klasse `Buch` soll in der Datei `cls/Buch.php` implementiert werden. Sie soll Attribute `author`, `title`, `year`, `genre`, `isbn` und `cover` enthalten. Für diese Klasse sollen die folgenden Methoden implementiert werden:

`__construct` überträgt die Daten aus einem Array mit Indizes 0,...,5 an die Attribute eines Buch-Objektes.

`tableRow` erzeugt eine Zeile für das aktuelle Buch-Objekt in der Tabelle für die Listenansichten (vgl. Folien 2,...,5). Die Methode nutzt das Template `tpl/bookTableRow.htm` und setzt die Werte der Attribute für die Platzhalter ein. Die Methode liefert einen String als Ergebnis zurück.

`tile` erzeugt eine Kachel für das aktuelle Buch-Objekt in der Kachelansicht (vgl. Folien 6,...,9). Die Methode nutzt das Template `tpl/bookTile.htm` und setzt die Werte der Attribute für die Platzhalter ein. Die Methode liefert einen String als Ergebnis zurück.

`bookDetails` erzeugt eine Teiltabelle für das aktuelle Buch-Objekt. Die Methode nutzt das Template `tpl/bookDetails.htm` und setzt die Werte der Attribute für die Platzhalter ein. Die Methode liefert einen String als Ergebnis zurück und kann in der folgenden Methode genutzt werden.

`bookView` erzeugt eine Einzeltabelle für das aktuelle Buch-Objekt (vgl. Folien 3 und 7). Die Methode nutzt das Template `tpl/bookView.htm` und setzt die Werte der Attribute bzw. das Ergebnis der Methode `bookDetails` für die Platzhalter ein. Die Methode liefert einen String als Ergebnis zurück.

Klasse Sachbuch

Die Klasse **Sachbuch** soll in der Datei `cls/Sachbuch.php` implementiert werden. Die Klasse ist Kindklasse der Klasse **Buch** und enthält zusätzlich das Attribut `topic` für das Thema des Sachbuches. Für diese Klasse sollen die folgenden Methoden implementiert werden:

__construct überträgt die Daten aus einem Array mit Indizes 0,...,6 an die Attribute eines Buch-Objektes. Im Array an Index 6 steht jetzt das Thema des Sachbuches.

tableRow erzeugt eine Zeile für das aktuelle Buch-Objekt in der Tabelle für die Listenansichten (vgl. Folien 2,...,5). Die Methode nutzt die entsprechende Methode der Elternklasse **Buch** und setzt das zusätzliche Attribut für den Platzhalter für das Thema. Die Methode liefert einen String als Ergebnis zurück.

tile muss hier nicht implementiert werden, sondern wird von der Elternklasse geerbt.

bookDetails erzeugt eine Teiltabelle für das aktuelle Buch-Objekt. Die Methode nutzt die entsprechende Methode der Elternklasse **Buch** und setzt das zusätzliche Attribut für den Platzhalter für das Thema. Die Methode liefert einen String als Ergebnis zurück und kann in der folgenden Methode genutzt werden.

bookView muss hier nicht implementiert werden, sondern wird von der Elternklasse geerbt.

Klasse Buchliste

Die Klasse **Buchliste** soll in der Datei `cls/Buchliste.php` implementiert werden. Sie soll das Attribut `list` enthalten. Das Attribut speichert ein assoziatives Array mit Buchobjekten bzw. Sachbuchobjekten in den einzelnen Arrayelementen. Dabei sollen die Indizes im Array die Titel der jeweiligen Bücher sein. Für diese Klasse sollen die folgenden Methoden implementiert werden:

__construct bekommt als Parameter den Namen einer Textdatei aus dem Ordner `txt` und einen Parameter, in dem die angeforderte Sortierung der Buchliste angegeben ist. Beide Parameter können auch in einem Array übergeben werden, z.B. dem Array `$_POST`.

Die Methode setzt das Attribut `list` zunächst auf ein leeres Array und liest dann die Bücher aus der angegebenen Textdatei in das Array ein. Die Textdateien enthalten jeweils eine Zeile pro Buch. Die Einträge für jedes Buch sind jeweils durch ein Semikolon getrennt. Zeilen für normale Bücher enthalten 6 Einträge, Zeilen für Sachbücher enthalten 7 Einträge. Aus jeder dieser Zeilen soll ein passendes Objekt erzeugt und im Array `list` gespeichert werden. Zuletzt wird das Array `list` entsprechend der Angaben im Parameter sortiert.

bookListTable erzeugt eine Tabelle für die Bücher in `list` für die Listenansicht (vgl. Folien 2,...,5). Die Methode nutzt das Template `tpl/bookListTable.htm`, berechnet einen String aus den Aufrufen der Methode `tableRow` für alle Bücher dieser Liste und setzt diesen String für den entsprechenden Platzhalter ein. Die Methode liefert einen String als Ergebnis zurück.

bookListTiles erzeugt eine Menge von Kacheln für die Bücher in `list` für die Kachelansicht (vgl. Folien 6,...,9). Die Methode berechnet einen String aus den Aufrufen der Methode `tile` für alle Bücher dieser Liste und liefert diesen String als Ergebnis zurück.

htmlPage bekommt einen Parameter, der angibt welche Ansicht angezeigt werden soll (initiale Ansicht (vgl. Folie 1), Listenansicht (vgl. Folien 2,...5) oder Kachelansicht (vgl. Folien 6,...,9)) und einen Parameter, der angibt, welches Buch in Einzelansicht angezeigt werden soll (kein Buch (Folien 1,2,4,5,6,7,9) oder ein Buch aus der Liste (Folien 3,7)). Die Methode nutzt das Template `tpl/mainPage.htm` und ersetzt die entsprechenden Platzhalter wie folgt:

- **FORM_ACTION** wird ersetzt durch einen Selbstaufruf
- **DATA_OPTIONS** wird ersetzt durch eine Liste aller Dateien mit Endung `.txt` im Unterordner `txt`.
- **BOOK_TABLE** wird ersetzt durch einen leeren String in der initialen Ansicht, das Ergebnis der Methode `bookListTable` der Klasse `Buchliste` in der Listenansicht oder das Ergebnis der Methode `bookListTiles` der Klasse `Buchliste` in der Kachelansicht.
- **BOOK_VIEW** wird ersetzt durch einen leeren String, wenn kein einzelnes Buch angezeigt werden soll oder das Ergebnis der Methode `bookView` der Klasse `Buch` bzw. `Sachbuch` des anzuzeigenden Buches sonst.

Mögliche Vorgehensweise

1. Implementierung der Klasse `Buch`: `__construct`, `tableRow`, `tile`
2. Test im Hauptprogramm: Erzeugung eines Buches aus einem Array
3. Implementierung der Klasse `Sachbuch`: `__construct`, `tableRow`
4. Test im Hauptprogramm: Erzeugung eines Sachbuches aus einem Array
5. Implementierung der Klasse `Buchliste`: `__construct`
6. Test im Hauptprogramm: Erzeugung einer Buchliste aus der Datei `txt/alle.txt`.
7. Implementierung der Klasse `Buchliste`: `bookListTable`
8. Test im Hauptprogramm: Ausgabe einer Tabelle aller Bücher.
9. Implementierung der Klasse `Buchliste`: `bookListTiles`
10. Test im Hauptprogramm: Ausgabe der Kacheln aller Bücher.
11. Implementierung der Klasse `Buchliste`: Erste Version von `htmlPage`
12. Implementierung des Hauptprogramms: Erzeugung einer Buchliste aus `txt/alle.txt` und Ausgabe der Listen- bzw. Kachelansicht durch Benutzung von `htmlPage`
13. Test im Hauptprogramm: initiale, Listen- und Kachelansicht ohne Bücher in Einzelansicht.
14. Implementierung der Klasse `Buch`: `bookDetails`, `bookView`
15. Test im Hauptprogramm: Ausgabe der Einzelansicht eines Buches.

16. Implementierung der Klasse **Sachbuch**: `bookDetails`
17. Test im Hauptprogramm: Ausgabe der Einzelansicht eines Sachbuches.
18. Implementierung der Klasse **Buchliste**: Finale Version von `htmlPage` (mit Paramtern)
19. Test im Hauptprogramm: initiale, Listen- und Kachelansicht.
20. Implementierung des Hauptprogramms: Finale Logik: Erzeugung einer Buchliste, ggf. wie im Formular angefordert. Erzeugung der Web-Seite durch Aufruf der Methode `htmlPage` und Ausgabe dieser Web-Seite.