

**Leistungsnachweis "Entwicklung Webbasierter Anwendungen"**  
**Sommersemester 2020**  
**20.08.2020**

Familienname	Vorname	Matrikelnummer	Note
Mustermann	Max	123456	1,0

Aufgabe	1	2	3	4	5	Gesamt
Punkte	17	7	18	16	7	65
Erreichte Punkte						

Bemerkungen:

- Bearbeitungszeit: 90 Minuten.
- Trennen Sie das letzte Blatt mit der allgemeinen Aufgabenbeschreibung und dem vorgegebenen Code der Klassen **Page** und **News** von der Klausur. Schreiben Sie keine Ergebnisse auf dieses Blatt, weil es nicht abgegeben werden soll! Lösen Sie bitte nicht die Klammerung der Aufgabenblätter.
- Lesen Sie zuerst die gesamte Aufgabenstellung bevor Sie mit der Lösung der Aufgaben beginnen.
- Als Hilfsmittel sind 10 beidseitig bedruckte oder beschriebene DinA4-Seiten mit beliebigem Inhalt (z.B. Cheat-Blätter für HTML, CSS oder Praktikumlösungen etc.) und echte Bücher (also keine gebundenen Ausdrucke sondern Bücher mit einer ISBN) erlaubt. Elektronische Geräte sind nicht erlaubt!
- Schreiben Sie klar und deutlich.
- Schreiben Sie standardkonformen Code, der die Regeln der Veranstaltung für ordentlichen Code und die Vorgaben durch die Seitenklassen berücksichtigt. Verwenden Sie Exceptions zur Fehlerbehandlung.
- Sollten Sie mehr Platz für die Beantwortung einer Frage benötigen, verwenden Sie die enthaltenen leeren Seiten (bitte entsprechende Verweise zur Aufgabe schreiben).
- Sie können sowohl in Deutsch als auch in Englisch antworten (You may also answer in English).

### Aufgabe 1: PHP/HTML (17 Punkte)

Erzeugen Sie das Grundgerüst der News-Seite inklusive Formular (siehe letztes Blatt) in standardkonformen HTML. Schreiben Sie den HTML-Code in den vorgegebenen HEREDOC-Block der Methode *generateHTMLView()* der Klasse *News* (siehe unten). Beachten Sie folgende Hinweise:

- Die News-Blöcke in der Mitte der Seite werden später mit AJAX eingefügt und sollen hier nicht erzeugt werden. Legen Sie hier nur ein Container-Element an, in das Sie die News später bequem einfügen können.
- Das Layout und Design der Seite entsteht später mit CSS. Machen Sie keine Formatierungen mit HTML!
- Das Logo liegt in der Datei "logo.png" vor.
- Der HTML-Befehl für das Copyright-Zeichen lautet: *&copy;*
- Die Formulardaten schickt die Klasse an sich selbst ab (news.php).

```
// Ergänzen Sie folgende Methode aus der Klasse News so, dass er die statische News-Seite erzeugt
protected function generateHTMLView() {
    $this->generatePageHeader("News");
echo <<<EOT

<body onload="pollNews();" > <!-- onload für Aufgabe 4b -->
    <header>
        <h1>Meine Zeitung</h1>
        
    </header>
    <nav>
        <ul>
            <li>Home</li> <li>Mediathek</li> <li>Politik</li> <li>Sport</li>
        </ul>
    </nav>
        <section>
            <h2>News-Ticker</h2>
            <div id="news"></div>
        </section>

    <section>
        <h2>Ihre News</h2>
        <form action="news.php" method="post" accept-charset="UTF-8">
            <div class="formular">
                <input type="text" name="Titel" placeholder="Titel Ihrer News" value="" tabin-
dex="1"/>
                <textarea name="Text" placeholder="Ihre News" tabindex="2" access-
key="n"></textarea>
                <input type="submit" value="Absenden" tabindex="-1"/>
            </div>
        </form>
    </section>

    <footer><p>&copy; Fachbereich Informatik</p></footer>

    </body>
EOT;

    $this->generatePageFooter();
}
```

Weiterer Platz für Ihre Lösung...

Lösung

## Aufgabe 2: PHP (7 Punkte)

Vervollständigen Sie die Klasse *News*:

Wenn eine Benutzer\*in über das Formular eine neue News abschickt, soll die News in der Tabelle *news* der Datenbank abgelegt werden.

// Ergänzen Sie hier Ihren Code

```
protected function processReceivedData() {
```

```
    parent::processReceivedData();
```

```
    // Aufgabe 3d *****
```

```
    if (isset($_GET["JSON"])) {$this->JSON=true;}
```

```
    // Ende Aufgabe 3d *****
```

```
    if (isset($_POST["Titel"])&& isset($_POST["Text"])){
```

```
        $titel = $this->db->real_escape_string($_POST["Titel"]);
```

```
        $text = $this->db->real_escape_string($_POST["Text"]);
```

```
        $sql = "INSERT INTO news (title, text) values ('$titel', '$text');";
```

```
        if(!$this->db->query($sql))
```

```
            throw new Exception("Insert failed: ".$this->db->error);
```

```
        header('Location:news.php'); // löst POST-Blockade; nicht gefordert für die Klausur
```

```
    }
```

```
}
```

### Aufgabe 3: PHP (18 Punkte)

Vervollständigen Sie die Klasse News.

#### Aufgabe 3a:

Lesen Sie die News-Einträge aus der Datenbank aus, um sie anschließend (Aufgabe 3b) als JSON-Antwort bereitzustellen. Beachten Sie die Vorgaben der Seitenklassen und wandeln Sie die Datensätze in eine geeignete Datenstruktur für die anschließende Weiterverarbeitung um. Geben Sie NICHT das Datenbankobjekt nach außen, sondern verbergen Sie die Datenbankzugriffe innerhalb der Methode.

**Hinweis:** Die News sollen chronologisch sortiert nach Timestamp (neueste News oben links) ausgegeben werden.

```
// Ergänzen Sie hier Ihren Code
protected function getViewData() {
    $news = array();
    $sql = "SELECT id, title, text, timestamp FROM news ORDER BY timestamp DESC";
    $recordset = $this->db->query($sql);
    if (!$recordset)
        throw new Exception("Fehler in Abfrage: " . $this->db->error);
    while ($record = $recordset->fetch_assoc()) {
        // Aufgabe 3c) *****
        $record["timestamp"] = $this->getLocalizedDate($record["timestamp"]); // 3c)
        // Ende Aufgabe 3c) *****
        $news[] = $record;
    }
    $recordset->free();

    return $news;
}
```

**Aufgabe 3b:** Schreiben Sie eine Methode, welche die ausgelesenen News-Einträge als JSON-Antwort liefert.

```
// Ergänzen Sie hier Ihren Code
protected function generateJSONView () {

    header("Content-Type: application/json; charset=UTF-8");
    $news = $this->getViewData();
    $serializedData = json_encode($news);
    echo $serializedData;

}
```

**Aufgabe 3c:** Leider liefert die Datenbank die Zeitstempel in einem amerikanischen Format.

Die vorgegebene Methode `getLocalizedDate($date)` der Klasse `News` wandelt das Datumsformat (für einen Browser mit deutscher Sprache) in das deutsche Format.

- Bauen Sie den Aufruf dieser Methode an geeigneter Stelle in Ihren bisherigen Code ein und markieren Sie die Codestelle(n) gut erkennbar mit "3c)"

siehe 3a) `$record["timestamp"] = $this->getLocalizedDate($record["timestamp"]);`

**Aufgabe 3d:** Wenn die Klasse `News` über JSON aufgerufen wird, übermittelt der AJAX-Request den Formularwert "JSON=1" als GET-Parameter (siehe JavaScript-Teil). Setzen Sie das vordefinierte Attribut `JSON` der Klasse `News` auf `true`, wenn der Aufruf über JSON erfolgt.

- Bauen Sie den neuen Code in Ihren bisherigen Code ein und markieren Sie die eingefügten Codestelle(n) gut erkennbar mit "3d)"

siehe 2) in `processReceivedData` : `if (isset($_GET["JSON"])) {$this->JSON=true;}`

**Aufgabe 3e:** Die Klasse `News` muss, wenn Sie über JSON aufgerufen wird, eine JSON-Antwort (`generateJSONView()`) liefern. Wird die Seite als normale Webseite über den Browser aufgerufen, soll sie eine HTML-Antwort (`generateHTMLView()`) erzeugen. Schreiben Sie eine Funktion, die diese Fallunterscheidung umsetzt.

Hinweis: Die Methoden `generateHTMLView()` und `generateJSONView()` haben Sie bereits in Aufgabe 1) bzw. 3b) geschrieben.

```
// Ergänzen Sie hier Ihren Code
protected function generateView () {

    if ($this->JSON){
        $this->generateJSONView();
    } else {
        $this->generateHTMLView();
    }

}
```

Platz für Notizen oder weitere Lösungen

Lösung

#### Aufgabe 4: ECMAScript / AJAX (16 Punkte)

Die News-Seite soll mit AJAX zyklisch alle News-Einträge vom Server holen und anzeigen. Die bekannten Standard-Methoden für einen AJAX-Request sind gegeben.

```
//news.js
let request = new XMLHttpRequest();

function requestData() {
  "use strict";
  request.open("GET", "news.php?JSON=1");
  request.onreadystatechange = processData;
  request.send(null);
}
```

```
function processData() {
  "use strict";
  if (request.readyState == 4) {
    if (request.status == 200) {
      if (request.responseText != null)
        processNews(request.responseText);
      else console.error("Dokument ist leer");
    } else console.error("Uebertragung
    fehlgeschlagen");
  } else;
}
```

**Aufgabe 4 a:** Schreiben Sie eine JavaScript-Funktion **pollNews()**, welche die News über AJAX abfragt, indem sie die (gegebene) Funktion *requestData()* aufruft. Die News-Daten sollen sofort (beim Laden der Seite) und dann zyklisch alle 8 Sekunden angefordert werden.

```
//news.js
function pollNews() {
  "use strict";

  requestData();
  window.setInterval(requestData, 8000);

}
```

**Aufgabe 4b:** An welcher Stelle der Klasse *News* würden Sie den Aufruf von *pollNews()* machen? Schreiben Sie hier die veränderte Codezeile hin (oder beschreiben Sie die Stelle):

Siehe Aufgabe 1: `<body onload="pollNews();">`

---



**Aufgabe 4c:** Nun sollen die über AJAX empfangenen News in die Webseite eingebaut werden. Analysieren Sie folgende Funktion **createDOM()** und ergänzen Sie die Lücken so, dass eine valide HTML-Struktur entsteht.

// Vervollständigen Sie die Lücken im Code

```
function createDOM(row){  
  "use strict";  
  let newsEntry = document.createElement("article");  
  let title = document.createElement("h3");  
  let textNodeTitle = document.createTextNode(row["title"]);  
  title.appendChild(textNodeTitle);  
  let time = document.createElement("p");  
  time.className = "timestamp";  
  let textNodeTime = document.createTextNode(row["timestamp"]);  
  time.appendChild(textNodeTime);  
  let text = document.createElement("p");  
  let textNodeText = document.createTextNode(row["text"]);  
  
  text.appendChild(textNodeText);  
  newsEntry.appendChild(title);  
  newsEntry.appendChild(time);  
  newsEntry.appendChild(text);  
  return newsEntry;  
}
```

**Aufgabe 4d:** Implementieren Sie nun die Methode, welche die AJAX-Daten in die Webseite einfügt.

- Nutzen Sie dabei die Methode `createDOM()` aus Aufgabe 4c.
- Beachten Sie, dass der Request zyklisch ausgeführt wird und die News nicht mehrfach angezeigt werden sollen.
- Benennen Sie die Methode so, dass sie zum AJAX-Aufruf passt

Tipp: Mit `parent.firstChild` erhalten Sie den ersten Kindknoten des DOM-Knoten `parent`.

Mit `parent.removeChild(ChildNode)` löscht man einen Kindknoten.

```
// Verarbeitung der AJAX-Anfrage
function _processNews__(data) {
    "use strict";

    let dataObject = JSON.parse(data);
    let newsSection = document.getElementById("news");

    // delete all existing news
    while (newsSection.firstChild) {
        newsSection.removeChild(newsSection.lastChild);
    }

    // insert news from JSON
    for (let i = 0; i < dataObject.length; i++){
        let row = dataObject[i];
        if (newsSection && row["title"] && row["text"] && row["timestamp"]){
            let article = createDOM(row);
            newsSection.appendChild(article);
        }
    }
}
```

**Platz für Notizen, Korrekturen oder weitere Lösungen... es kommt aber noch eine Aufgabe!**

Lösung

### Aufgabe 5: CSS (7 Punkte)

Geben Sie den CSS-Code an, der die folgenden Aspekte des Designs der in der Aufgabenstellung dargestellten Zeitung umsetzt. Designteile nach denen nicht gefragt wird, müssen Sie nicht umsetzen!

a) Alle Texte sollen in einer schnörkel-freien Schrift (ohne Serifen), vorzugsweise 'Roboto' dargestellt werden.	<pre>body { /* Aufgabe 5a */   font-family: Roboto, sans-serif; }</pre>
b) Die Navigationselemente (Home, Mediathek, ...) sollen horizontal nebeneinander ausgerichtet werden.	<pre>li { /* Aufgabe 5b */   list-style-type: none; /* oder list-style: none; */   display: inline-block; }</pre>
c) Die News-Einträge sollen als "quadratische Kacheln" nebeneinander angezeigt werden. Jede Kachel hat eine <u>feste</u> Breite und <u>feste</u> Höhe von 15em. Bei längeren Texten soll rechts ein Scrollbalken an der News erscheinen. Die Anzahl der Kacheln in einer Zeile ändert sich mit der Breite des Browserfensters.	<pre>article { /* Aufgabe 5c */   display: inline-block;    width: 15em;   height: 15em;    overflow: auto; }</pre>
d) Der Footer (unterhalb des Formulars) soll mit 2 em Abstand nach oben, zentriert, mit Hintergrundfarbe <i>grau</i> und Schriftfarbe <i>weiß</i> dargestellt werden.	<pre>footer { /* Aufgabe 5d */   margin-top: 2em;   text-align: center;   background-color: grey;   color: white; }</pre>

Achtung! Dieses Blatt dürfen Sie von der Klausur abtrennen.  
Es soll nicht abgegeben werden. Schreiben Sie keine Ergebnisse auf dieses Blatt!

## Aufgabe „News-Ticker“

**h\_da**  
HOCHSCHULE DARMSTADT  
UNIVERSITY OF APPLIED SCIENCES  
  
**fbi**  
FACHBEREICH INFORMATIK

Meine Zeitung

Home
Mediathek
Politik
Sport

### News-Ticker

**Ist das <script>-Tag böse?**

09.07.2020 11:05:38

Am 13.07.2020 berichtet Prof. Dr. E. Wa über die Vor- und Nachteile der JavaScript-Tags in HTML

**Solidarität statt Rendite? – Mit ihrem 1. Dialog-Forum trifft die h\_da einen Nerv**

17.03.2020 10:12:19

Mit ihrem 1. Dialog-Forum trifft die h\_da einen Nerv: Rund 300 Bürgerinnen und Bürger verfolgen die Podiumsdiskussion über das Verhältnis von Gemeinwohl-Ökonomie und Wirtschaftswachstum

**Campus Haardtring: Studierende und Beschäftigte setzen sich für mehr Verkehrssicherheit ein**

17.03.2020 10:10:13

Viele Studierende und Beschäftigte am Campus Haardtring erreichen die dortigen Gebäude über die Haltestelle „Hochschule West“.

**Hochschulpakt unterzeichnet: h\_da erwartet positiven Effekt auf Forschung und Lehre**

17.03.2020 10:09:05

Der hessische Hochschulpakt für die Jahre 2021 bis 2025 ist unterschrieben. Darin haben die Landesregierung und die Hochschulen die Rahmenbedingungen für die

### Ihre News

© Fachbereich Informatik

Schreiben Sie eine Webseite zur Anzeige und Bearbeitung eines vereinfachten barrierefreien und responsiven News-Tickers. Es werden die vorliegenden News aus einer Datenbank angezeigt und man kann selbst neue News erstellen. Die Seite aktualisiert die Einträge automatisch (via AJAX). Mit den Elementen Home, Mediathek, Politik und Sport kann auf andere Seiten navigiert werden. Diese Seiten sind aber kein Bestandteil der Klausur.

Mit einer Ajax-Get-Anfrage werden zyklisch alle News-Einträge vom Server geholt und angezeigt. Jeder News-Eintrag in der Datenbank-Tabelle *news* hat eine eindeutige Id, einen Titel, einen Text und einen Zeitstempel der Erstellung.

news			
id - int (auto)	title - varchar(255)	text - varchar(1024)	timestamp - timestamp DEFAULT current_timestamp()
1	Ist das <script>-Tag böse?	Am 13.07.2020 berichtet Prof. Dr. E. Wa über die Vor- und Nachteile der JavaScript-Tags in HTML	2020-07-09 11:05:38
2	Solidarität statt Rendite? ...	Mit ihrem 1. Dialog-Forum trifft die h_da einen Nerv...	2020-03-17 10:12:19

Achtung! Dieses Blatt dürfen Sie von der Klausur abtrennen.  
Es soll nicht abgegeben werden. Schreiben Sie keine Ergebnisse auf dieses Blatt!

Die bekannte Klasse **Page**, das Grundgerüst der Klasse **News** und die Datenbank sind vorgegeben.

<pre> abstract class Page {     protected \$db;      protected function __construct() {         \$this-&gt;db = new MySQLi(             ["localhost", "public", "public", "2020_News"];         if (mysqli_connect_errno())             throw new Exception(mysqli_connect_error());         if (!\$this-&gt;db-&gt;set_charset("utf8"))             throw new Exception(\$this-&gt;db-&gt;error);     }      public function __destruct() {         \$this-&gt;db-&gt;close();     }      protected function generatePageHeader(\$headline="")     {         header("Content-type: text/html; charset=UTF-8");         \$headline = htmlspecialchars(\$headline);         echo &lt;&lt;&lt;EOT         &lt;!DOCTYPE html&gt;         &lt;html lang="de"&gt;         &lt;head&gt;         &lt;meta charset="UTF-8" /&gt;         &lt;title&gt; \$headline &lt;/title&gt;         &lt;link rel="stylesheet" type="text/css"             href="news.css" /&gt;         &lt;script src="news.js"&gt; &lt;/script&gt;         &lt;/head&gt;         EOT;     }      protected function generatePageFooter() {         echo "&lt;/html&gt;";     }      protected function processReceivedData() {     } } </pre>	<pre> // class News require_once 'Page.php'; class News extends Page     private \$JSON = false; {     protected function __construct() {         parent::__construct();     }      protected function __destruct() {         parent::__destruct();     }      protected function getLocalizedDate(\$date){         \$date = new DateTime(\$date);         if         (strpos(\$_SERVER['HTTP_ACCEPT_LANGUAGE'], "         de-DE")&gt;-1){             return \$date-&gt;format("d.m.Y H:i:s");         } else { // English as default             return \$date-&gt;format("Y/m/d H:i:s");         }     }      public static function main()     {         try {             \$page = new News();             \$page-&gt;processReceivedData();             \$page-&gt;generateView();         }         catch (Exception \$e) {             header("Content-type: text/plain;             charset=UTF-8");             echo \$e-&gt;getMessage();         }     } }  News::main(); </pre>
--	--