

ANTIPATTERN

BELEGARBEIT

Beleg von

Simon Retsch (s74077)

Falk-Jonatan Strube (s74053)

Jürgen Tomaszewski (s76451)

Betreuender Hochschullehrer

Prof. Dr. Anna Sabine Hauptmann

Eingereicht am: 7. Juli 2017

INHALTSVERZEICHNIS

1 Aufgabe	2
2 Quellcode	2
2.1 Originaler Quellcode	3
2.2 Analyse des Quellcodes	15
3 Symptome	16
4 Beseitigung	16

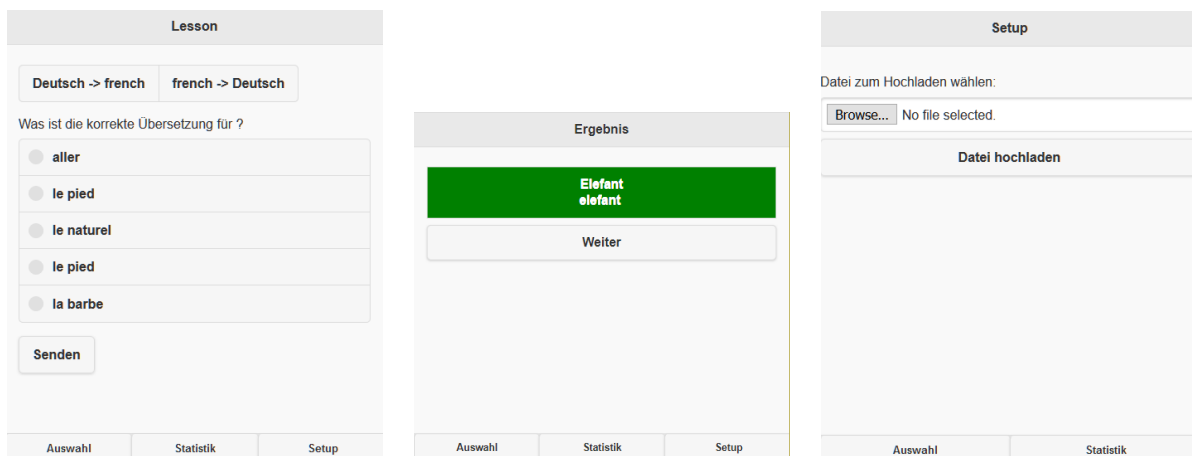
1 AUFGABE

Die Aufgabe dieses Belegs ist, einen (objektorientiert) geschriebenen Quellcode zu betrachten, bei dem Verbesserungspotential vorhanden ist. Der originale Quellcode, die Symptome und deren Beseitigung sind zu beschreiben. Auszüge aus Klassendiagrammen sind erwünscht.

2 QUELLCODE

Der Quellcode ist aus einer Belegarbeit des Fachs INTERNETTECHNOLOGIEN 1 entnommen. Dort war die Aufgabe einen primitiven Vokabeltrainer über eine Webseite zu programmieren. Zur Veranschaulichung wurden daraus drei Dateien ausgewählt, die gleichzeitig auch drei Anzeigen der Software darstellen:

- lessonsJQ.php
- resultJQ.php
- setupJQ.php



Der vollständige originale Quellcode ist auf Folgenden Seiten in Unterabschnitt 2.1, die Analyse dieses Quellcodes ab Seite 15 im Unterabschnitt 2.2 zu finden.

2.1 ORIGINALER QUELLCODE

- lessonsJQ.php

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset = "UTF-8"/>
5
6     <!-- jQuery stuff -->
7     <meta name="viewport" content="width=device-width, initial-
8       scale=1">
9     <link rel="stylesheet" href="https://code.jquery.com/mobile
10      /1.4.5/jquery.mobile-1.4.5.min.css">
11     <script src="https://code.jquery.com/jquery-1.11.3.min.js">
12       </script>
13     <script src="https://code.jquery.com/mobile/1.4.5/jquery.
14       mobile-1.4.5.min.js"></script>
15     <style>
16       //[data-role=page]{height: 100% !important; position:
17         relative !important;}
18       [data-role=footer]{bottom:0; position:absolute !important
19         ; top: auto !important; width:100%;}
20     </style>
21 </head>
22
23 <body>
24
25   <div data-role="page">
26     <div data-role="header">
27       <h1>Lesson</h1>
28     </div>
29     <div data-role="main" class="ui-content">
30
31     <!-- main content -->
32
33     <!-- javascript -->
34     <script>
35       var mode = 1;
36
37       // function for the ajax call
38       function showHint(str, lesson)
39       {
40         num = 0;
41
42         var xmlhttp = new XMLHttpRequest();
43         xmlhttp.onreadystatechange = function()
44         {
45           if (this.readyState == 4 && this.status == 200)
46           {
47             var myObj = JSON.parse(this.responseText);
```

```

43     var ok = myObj[10];
44
45     // was there an error getting the words?
46     if(ok == 0)
47     {
48         // error field
49         document.getElementById("err").innerHTML = "";
50
51         // start the form
52         document.getElementById("q").innerHTML = myObj[0];
53
54         // radio
55         document.getElementById("lab1").innerHTML = myObj
56             [1];
57         document.getElementById("poss1").value = myObj[1];
58
59         document.getElementById("lab2").innerHTML = myObj
60             [2];
61         document.getElementById("poss2").value = myObj[2];
62
63         document.getElementById("lab3").innerHTML = myObj
64             [3];
65         document.getElementById("poss3").value = myObj[3];
66
67         document.getElementById("lab4").innerHTML = myObj
68             [4];
69         document.getElementById("poss4").value = myObj[4];
70
71         document.getElementById("lab5").innerHTML = myObj
72             [5];
73         document.getElementById("poss5").value = myObj[5];
74
75         // end
76         document.getElementById("e1").value = myObj[6];
77         document.getElementById("e2").value = myObj[7];
78         document.getElementById("e3").value = myObj[8];
79
80         // mode for POST
81         document.getElementById("postMode").value = myObj
82             [9];
83     }
84     else if(ok == 1)
85     {
86         document.getElementById("send").type = "hidden";
87
88         // error field
89         document.getElementById("err").innerHTML = "<br>In
90             der Lektionsdatei befinden sich Zeichen die
91             nicht zulässig sind!<br>";
92
93         // radio

```

```

86     document.getElementById("lab1").innerHTML = "error"
      ;
87     document.getElementById("poss1").value = myObj[1];
88     document.getElementById("poss1").type = "hidden";
89
90     document.getElementById("lab2").innerHTML = "error"
      ;
91     document.getElementById("poss2").value = myObj[2];
92     document.getElementById("poss2").type = "hidden";
93
94     document.getElementById("lab3").innerHTML = "error"
      ;
95     document.getElementById("poss3").value = myObj[3];
96     document.getElementById("poss3").type = "hidden";
97
98     document.getElementById("lab4").innerHTML = "error"
      ;
99     document.getElementById("poss4").value = myObj[4];
100    document.getElementById("poss4").type = "hidden";
101
102    document.getElementById("lab5").innerHTML = "error"
      ;
103    document.getElementById("poss5").value = myObj[5];
104    document.getElementById("poss5").type = "hidden";
105
106    }
107    else
108    {
109        document.getElementById("send").type = "hidden";
110
111        // error field
112        document.getElementById("err").innerHTML = "<br>Die
          Datei kann nicht geöffnet werden!<br>";
113
114        // radio
115        document.getElementById("lab1").innerHTML = "error"
          ;
116        document.getElementById("poss1").value = myObj[1];
117        document.getElementById("poss1").type = "hidden";
118
119        document.getElementById("lab2").innerHTML = "error"
          ;
120        document.getElementById("poss2").value = myObj[2];
121        document.getElementById("poss2").type = "hidden";
122
123        document.getElementById("lab3").innerHTML = "error"
          ;
124        document.getElementById("poss3").value = myObj[3];
125        document.getElementById("poss3").type = "hidden";
126
127        document.getElementById("lab4").innerHTML = "error"

```

```

128         ;
129         document.getElementById("poss4").value = myObj[4];
130         document.getElementById("poss4").type = "hidden";
131
132         document.getElementById("lab5").innerHTML = "error"
133         ;
134         document.getElementById("poss5").value = myObj[5];
135         document.getElementById("poss5").type = "hidden";
136     }
137 }
138 };
139
140 xmlhttp.open("GET", "switchLesson.php?m=" + str + "&
141         lesson=" + lesson, true);
142 xmlhttp.send();
143 }
144 </script>
145
146 <?php
147 // get post variables
148
149 $lesson = $_POST["lesson"];
150 $mode = $_POST["mode"];
151
152 // if no mode was given so far, set mode to 1
153 if($mode == "")
154 {
155     $mode = 1;
156 }
157
158 // get .stat fileName
159 $lessons_name = str_replace(".txt", "", $lesson);
160 $lessons_name = str_replace("uploads/", "", $lessons_name);
161
162 // get the php vars 'lesson' and 'mode' into js vars cause
163 js can't read in POST's
164 echo "<script>";
165 echo "var lesson = " . json_encode($lesson) . " ";
166 echo "mode = " . json_encode($mode) . " ";
167 echo "showHint(mode, lesson);";
168 echo "</script>";
169 ?>
170
171 <!-- Error message -->
172 <p id="err" style="display:inline"></p>
173
174 <!-- switch words buttons -->

```

```

175 <div data-role="controlgroup" data-type="horizontal">
176   <a href="#" class="ui-btn ui-btn-up-c" onclick="showHint(1,
      lesson)">
177     <?php echo "Deutsch -> ".$lessons_name; ?>
178   </a>
179   <a href="#" class="ui-btn" onclick="showHint(0, lesson)">
180     <?php echo $lessons_name." -> Deutsch"; ?>
181   </a>
182 </div>
183
184 <!-- start Radio with question -->
185 <form method="POST" action="resultJQ.php?lesson=<?php echo
      $lesson;?>">
186   <fieldset data-role="controlgroup">
187     <legend>
188       Was ist die korrekte Übersetzung für <p id="q" style="
          display:inline"></p>?
189     </legend>
190
191     <!-- radio options -->
192     <label for="poss1" id="lab1"></label>
193     <input type="radio" name="answer" id="poss1" value="">
194
195     <label for="poss2" id="lab2"></label>
196     <input type="radio" name="answer" id="poss2" value="">
197
198     <label for="poss3" id="lab3"></label>
199     <input type="radio" name="answer" id="poss3" value="">
200
201     <label for="poss4" id="lab4"></label>
202     <input type="radio" name="answer" id="poss4" value="">
203
204     <label for="poss5" id="lab5"></label>
205     <input type="radio" name="answer" id="poss5" value="">
206
207   </fieldset>
208   <input type="hidden" name="solution" id="e1" value="">
209   <input type="hidden" name="translation" id="e2" value="">
210   <input type="hidden" name="lesson" id="e3" value="" />
211   <input type="hidden" name="mode" id="postMode" value="" />
212
213   <input type="submit" data-inline="true" id="send" value="
      Senden">
214 </form>
215 </div>
216
217 <!-- footer -->
218 <div data-role="footer"><div data-role="navbar"><ul><li>
219
220   <!-- Option 1 -->
221   <start>

```

```

222         <form action="startJQ.php">
223             <input type="submit" value="Auswahl" />
224         </form>
225     </start>
226
227 </li><li>
228
229     <!-- Option 2 -->
230     <statistic>
231         <form action="statisticJQ.php">
232             <input type="submit" value="Statistik" />
233         </form>
234     </statistic>
235
236 </li><li>
237
238     <!-- Option 3 -->
239     <setup>
240         <form action="setupJQ.php">
241             <input type="submit" value="Setup" />
242         </form>
243     </setup>
244
245 </li></ul></div></div>
246
247
248 </div>
249 </body>
250 </html>

```

- resultJQ.php

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset = "UTF-8"/>
5
6     <!-- jQuery stuff -->
7     <meta name="viewport" content="width=device-width, initial-
8       scale=1">
9     <link rel="stylesheet" href="https://code.jquery.com/mobile
10      /1.4.5/jquery.mobile-1.4.5.min.css">
11     <script src="https://code.jquery.com/jquery-1.11.3.min.js">
12       </script>
13     <script src="https://code.jquery.com/mobile/1.4.5/jquery.
14       mobile-1.4.5.min.js"></script>
15   <style>
16     // [data-role=page]{height: 100% !important; position:
17       relative !important;}
18     [data-role=footer]{bottom:0; position:absolute !important
19       ; top: auto !important; width:100%;}
20   </style>

```



```

15 </head>
16
17 <body>
18
19 <div data-role="page">
20 <div data-role="header">
21 <h1>Ergebnis</h1>
22 </div>
23 <div data-role="main" class="ui-content">
24
25 <!-- main content -->
26 <?php
27 // read in the chosen lesson
28 $from = $_POST["from"];
29 $lesson = $_POST["lesson"];
30 $mode = $_POST["mode"];
31
32 $answer = $_POST["answer"];
33 $solution = $_POST["solution"];
34 $translation = $_POST["translation"];
35
36 $info = array(2);
37
38
39 // get .stat fileName
40 $lessons_name = str_replace(".txt", "", $lesson);
41 $lessons_name = str_replace("uploads/", "", $lessons_name
42 );
43 $fileName = $lessons_name.".stat";
44
45 // open statistic file
46 $file = fopen($fileName, "c+") or die("<br>Die Datei kann
47 nicht geöffnet werden!");
48
49 // get content
50 $string = fgets($file);
51 fclose($file);
52
53 // was the file just created?
54 if($string != "")
55 {
56 $line = explode(';', $string);
57 }
58 else
59 {
60 $line[0] = 0;
61 $line[1] = 0;
62 }
63
64 // correct the numbers of the statistic and display

```

```

64
65     if(strcmp($answer, $solution) == 0)
66     {
67         echo "<a class=\"ui-btn\" style=\"background: green;
68             color: white;\>";
69         echo $translation."<br>";
70         echo $solution."<br>";
71         echo "</a>";
72
73         // correct answer counter
74         $line[0] += 1;
75     }
76     else
77     {
78         echo "<a class=\"ui-btn\" style=\"background: red;
79             color: white;\>";
80         echo $translation."<br>";
81
82         if(strcmp($answer, "") == 0)
83             echo "(keine Antwort ausgewählt!)<br>";
84         else
85             echo "<strike>".$answer."</strike><br>";
86
87         echo $solution."<br>";
88         echo "</a>";
89     }
90
91     // question counter
92     $line[1] += 1;
93
94     $output = $line[0].";".$line[1];
95
96     // open statistic file
97     $file = fopen($fileName, "w") or die("<br>Die Datei kann
98         nicht zum schreiben geöffnet werden!");
99
100    // write content
101    fwrite($file, $output);
102
103    fclose($file);
104    ?>
105
106    <forward>
107    <form action="lessonsJQ.php?lesson=<?php echo $lesson;?>"
108        method="POST">
109        <input type="hidden" name="lesson" value="<?php echo
110            $lesson;?>" />
111        <input type="hidden" name="mode" value="<?php echo
112            $mode;?>" />
113        <input type="submit" value="Weiter" />

```

```

109     </form>
110 </forward>
111
112 </div>
113
114     <!-- footer -->
115     <div data-role="footer"><div data-role="navbar"><ul><li>
116
117         <!-- Option 1 -->
118         <start>
119             <form action="startJQ.php">
120                 <input type="submit" value="Auswahl" />
121             </form>
122         </start>
123
124     </li><li>
125
126         <!-- Option 2 -->
127         <statistic>
128             <form action="statisticJQ.php">
129                 <input type="submit" value="Statistik" />
130             </form>
131         </statistic>
132
133     </li><li>
134
135         <!-- Option 3 -->
136         <setup>
137             <form action="setupJQ.php">
138                 <input type="submit" value="Setup" />
139             </form>
140         </setup>
141
142     </li></ul></div></div>
143 </div>
144 </body>
145 </html>

```

- setupJQ.php

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset = "UTF-8"/>
5
6     <!-- jQuery stuff -->
7     <meta name="viewport" content="width=device-width, initial-
8       scale=1">
9     <link rel="stylesheet" href="https://code.jquery.com/mobile
10      /1.4.5/jquery.mobile-1.4.5.min.css">
11     <script src="https://code.jquery.com/jquery-1.11.3.min.js">
12       </script>

```

```

10 <script src="https://code.jquery.com/mobile/1.4.5/jquery.
    mobile-1.4.5.min.js"></script>
11 <style>
12     //[data-role=page]{height: 100% !important; position:
        relative !important;}
13     [data-role=footer]{bottom:0; position:absolute !important
        ; top: auto !important; width:100%;}
14 </style>
15 </head>
16
17 <body>
18
19     <div data-role="page">
20     <div data-role="header">
21         <h1>Setup</h1>
22     </div>
23     <div data-role="main" class="ui-content">
24
25     <!-- main content -->
26     <?php
27         // read info where it is coming from
28         $from = $_POST["from"];
29
30         // get uploaded file name
31         $target_dir = ""; //uploads/";
32         $target_file = $target_dir . basename($_FILES["
            fileToUpload"] ["name"]);
33
34         $uploadOk = 1;
35
36         // if there are html tag close stuff in the name replace
            the name with err
37         if (preg_match('/[...]/', $target_file))
38         {
39             // one or more of the 'special characters' found in
                $string
40             $target_file = "err";
41             echo "<br>Im Dateinamen befinden sich unzulässige
                Zeichen.";
42             $uploadOk = 0;
43         }
44
45
46
47         $fileType = pathinfo($target_file,PATHINFO_EXTENSION);
48
49         if(strcmp($from, "setup") == 0)
50         {
51
52             // Check if the file already exists
53             if (file_exists($target_file)) {

```

```

54     echo "<br>Die Datei existiert bereits.";
55     $uploadOk = 0;
56 }
57
58 // check if the ending of the file is txt
59 if($fileType != "txt") {
60     echo "<br>Die Dateiendung muss txt sein.";
61     $uploadOk = 0;
62 }
63
64 // Check if $uploadOk is set to 0 by an error
65 if ($uploadOk == 0) {
66     echo "<br>Datei wurde nicht hochgeladen.";
67
68 // if everything is ok, try to upload file
69 }
70 else
71 {
72
73     // move the uploaded file to the servers filesystem
74     if (move_uploaded_file($_FILES["fileToUpload"]["
75         tmp_name"], $target_file))
76     {
77         echo "<br>Die Datei ". basename( $_FILES["
78             fileToUpload"]["name"]). " wurde hochgeladen.";
79
80         // write the new path in the lessons.info
81         $file = fopen("lessons.data", "a") or die("<br>Die
82             Lessons-Datei kann nicht geöffnet werden!");
83         fwrite($file, $target_file."\n");
84         fclose($file);
85     }
86     else
87     {
88         echo "<br>Es gab einen Fehler beim speichern der
89             hochgeladenen Datei.";
90     }
91 }
92 ?>
93
94 </div>
95
96 <form action="setupJQ.php" method="post" data-ajax="false"
97     enctype="multipart/form-data">
98     Datei zum Hochladen wählen:
99     <input type="file" name="fileToUpload" id="fileToUpload">
100    <input type="hidden" name="from" value="setup">
101    <input type="submit" value="Datei hochladen" name="submit"
102        ">

```

```
99     </form>
100
101     <!-- footer -->
102     <div data-role="footer"><div data-role="navbar"><ul><li>
103
104         <start>
105             <form action="startJQ.php" method="POST">
106                 <input type="submit" value="Auswahl" />
107             </form>
108         </start>
109
110     </li><li>
111
112         <statistic>
113             <form action="statisticJQ.php">
114                 <input type="submit" value="Statistik" />
115             </form>
116         </statistic>
117
118     </li></ul></div></div>
119
120
121     </div>
122 </body>
123 </html>
```

2.2 ANALYSE DES QUELLCODES

Die entnommenen Dateien können wie folgt beschrieben werden:

- `lessonsJQ.php` gibt eine Vokabellektion zur Abfrage aus.
- `resultJQ.php` gibt das Ergebnis einer beantworteten Vokabelabfrage aus (ob die Antwort richtig/falsch war).
- `setupJQ.php` ermöglicht dem Nutzer eine Datei mit einer Vokabellektion hochzuladen.

Diese Dateien haben grob folgende Struktur. Dabei ist zu beachten, dass der originale Quellcode keine Funktionen hat – alles wird prozedural erledigt. Die Funktionalität der Quellcodezeilen wird im Folgenden durch deskriptive Funktionsnamen angedeutet. Weiterhin funktioniert jede Datei unabhängig von den anderen:

<code>lessonsJQ.php</code>	Beschreibung:
<code>printHeader()</code>	in HTML und CSS geschrieben. Gibt den HTML-Header aus.
<code>insertContent()</code>	in Javascript/Ajax geschrieben.
<code>getPostVars()</code>	in php geschrieben. Holt die <code>\$_POST</code> Variablen zur Weiterverarbeitung.
<code>execJavaScript()</code>	in php als HTML/Javascript geschrieben.
<code>printContentPlaceholder()</code>	in HTML geschrieben. Stellt "Hüllen"bereit, die per Javascript befüllt werden.
<code>printFooter()</code>	in HTML. Gibt den HTML-Footer aus.

<code>resultJQ.php</code>	Beschreibung:
<code>printHeader()</code>	in HTML und CSS geschrieben. Gibt den HTML-Header aus.
<code>printMainA()</code>	in HTML geschrieben. Gibt ersten Teil des Seiteninhalts aus.
<code>getPostVars()</code>	in php geschrieben. Holt die <code>\$_POST</code> Variablen zur Weiterverarbeitung.
<code>openStatFile()</code>	in php geschrieben. Öffnet Statistik-Datei.
<code>writeStatFile()</code>	in php geschrieben. Schreibt Antwort in Statistik.
<code>printMainB()</code>	in php geschrieben. Gibt zweiten Teil des Seiteninhalts, nämlich die Statistik der Statistik-Datei, aus.
<code>printMainC()</code>	in HTML geschrieben. Gibt dritten Teil des Seiteninhalts aus.
<code>printFooter()</code>	in HTML. Gibt den HTML-Footer aus.

<code>setupJQ.php</code>	Beschreibung:
<code>printHeader()</code>	in HTML und CSS geschrieben. Gibt den HTML-Header aus.
<code>printMainA()</code>	in HTML geschrieben. Gibt ersten Teil des Seiteninhalts aus.
<code>getPostVars()</code>	in php geschrieben. Holt die <code>\$_POST</code> Variablen zur Weiterverarbeitung.
<code>tryUpload()</code>	in php geschrieben. Versucht (gepostete) Datei hochzuladen.
<code>printMainB()</code>	in HTML geschrieben. Gibt zweiten Teil des Seiteninhalts, nämlich die Form für das Hochladen, aus.
<code>printFooter()</code>	in HTML. Gibt den HTML-Footer aus.

3 SYMPTOME

Die Hauptsymptome des Systems sind:

- Fire Drill:
Die folgenden Symptome sind vor allem dieses Architektur-Antipatterns – dem Umstand des Entwicklungsprozesses – zu Schulden: Das System wurde auf den letzten Drücker entwickelt.
- Spaghetti-Code:
Das System hat wenig Struktur. Die Funktionen sind in Dateien aufgeteilt, was positiv zu bewerten ist. In diesen Dateien ist jedoch HTML-, CSS-, PHP- und Javascript-Code durcheinander geschrieben.
Die Dateien sind zwar wie Objekte bezeichnet, allerdings nur prozedural programmiert.
- Blob (light):
Das System ist überhaupt nicht objektorientiert geschrieben. Allerdings haben die einzelnen Dateien den Charakter eines Objektes, das alles alleine macht.
- Goldener Hammer:
In der Datei `lessonsJQ.php` wird der Hammer Javascript/Ajax grundsätzlich zum Füllen der Webseite mit der ausgewählten Lektion angewendet – egal, ob die Dynamik benötigt wird oder nicht: Sie wird nicht benötigt, wenn die Datei initial aufgerufen wird (was der Regelfall ist). Sie wird eigentlich nur benötigt, wenn die Richtung der Vokabelabfrage gewechselt werden soll. Das führt zu teils doppelter Last für den Webserver, der den php-Code der Seite und gesonderten php-Code über die Ajax-Abfrage ausführen muss.

4 BESEITIGUNG

Zunächst wurde der Quellcode der 3 Dateien analysiert und bereits in grobe Funktionalitäten (als Funktionen benannt) eingeteilt, die den Ablauf des Programms beschreiben. Mit diesen Funktionen wurde nun ein Ist-Diagramm erstellt (siehe Unterabschnitt 2.2).

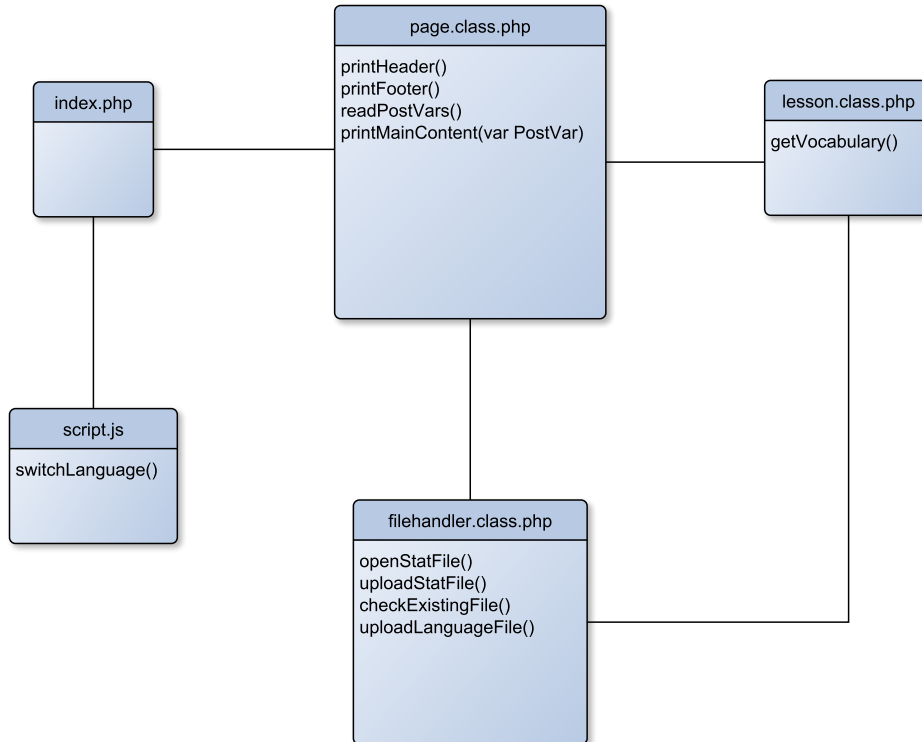
Wie dort gut zu erkennen, wiederholen sich in den drei PHP-Dateien einige Funktionalitäten immer wieder.

Ebenfalls ist zu erkennen, dass einige Aufgaben in gemeinsame Kategorien, also Klassen, eingeteilt werden können.

Mit diesen Erkenntnissen können Funktionen wie `printHeader()` und `printFooter()` in einer Klasse `page.class.php` als zusammen gelegt. Damit sind die Funktionalitäten gleichen Typs an einem Ort zu finden. Weitere Funktionen, die zum Beispiel den Aufbau der Seite betreffen, können dann ebenfalls in dieser Klasse untergebracht werden. Im Falle von Änderungen einer Funktion(alität) muss diese nicht in jeder einzelnen Datei, sondern nur an einer Stelle – der entsprechenden Klasse – angepasst werden.

Auch die Verwendung von großen Programmabschnitten in anderen Sprachen kann durch die Einteilung in PHP-Klassen deutlich übersichtlicher gestaltet werden. Damit soll vor allem auch erreicht werden, dass in einer späteren `index.php` Datei hauptsächlich eine Sprache benutzt wird (HTML). Nur die Initialisierung der `page`-Klasse und kurze Aufrufe tauchen so im HTML-Code auf. Der Javascript-Code wird ebenfalls in eine separate Datei verschoben und von der `index`-Datei eingebunden.

Die Modellierung der Funktionen in den einzelnen Klassen ist in folgendem Klassendiagramm angedeutet:



In den angedeuteten Klassen sind einige Funktionen, die beim Implementieren noch entstehen würden, noch nicht enthalten. Das Diagramm soll nur zur Veranschaulichung der Beseitigung der Symptome gelten.

Im Folgenden sind nun die einzelnen Dateien/Klassen entworfen:

- Durch die `index`-Datei wird die Webseite aufgerufen:

```

1 <!DOCTYPE html>
2 <html lang="de">
3 <?php
4 // Erzeuge die Verwaltung für den Seiteninhalt:
5 include_once 'page.class.php';
6 $page = new Page ();
7 ?>
8 <head>
9   <!-- Binde die Stylesheet- und Javascript-Datei ein: -->
10  <link rel="stylesheet" href="style.css" />
11  <script src="script.js"></script>
12 </head>
13 <body>
14  <?php
15  // Generiere Inhalt des Headers:
16  $page->printHeader();
17  ?>
18  <main>
19    <?php
20    // Generiere Main-Seiteninhalt:
21    $page->printMainContent ();
22    ?>
  
```

```

23     </main>
24 <?php
25     // Generiere Inhalt des Footers:
26     $page->printFooter();
27     ?>
28 </body>
29
30 </html>

```

- In der darin eingebundenen Javascript-Datei wird das dynamische Wechseln der Sprache realisiert:

```

1 function switchLanguage() {
2     /*
3     * Ändert die Abfragerichtung der Sprache, indem die
4     *   entsprechenden Felder ausgetauscht werden und die
5     *   Adresszeile korrigiert wird.
6     */
7 }

```

- Die page-Klasse realisiert die Logik der Webseite:

```

1 <?php
2 class page {
3     // Potentielle Variablen.
4
5     public function __construct() {
6         // Potentielle Variablen mit Start Werten initialisieren.
7     }
8
9     public function printHeader() {
10        // Erstellen des Kontents im Header der Webseite.
11    }
12
13    public function printFooter() {
14        // Erstellen des Kontents im Footer der Webseite.
15    }
16
17    public function readPostVars() {
18        // Einlesen der mit Post übergebenen Variablen.
19    }
20
21    public function printMainContent(var PostVar) {
22        // Erstellen des Kontents im Body der Webseite mit Hilfe
23        //   der übergebenen Variablen.
24    }
25 }

```

- Die lesson-Klasse kümmert sich um die Lektion (falls die entsprechende Seite aufgerufen wird):

```

1 <?php

```

```

2 class Lesson {
3     // potentielle globale Variablen (wie die Lektion o.ä.)
4
5     public function __construct() {
6         // Potentielle Variablen mit Start Werten initialisieren.
7     }
8
9     public function getVocabulary() {
10        // Holt die entsprechenden Vokabeln aus der Datei.
11    }
12 }

```

- Die filehandler-Klasse ist für den Umgang mit der Datei, in der die Vokabel abgespeichert sind, verantwortlich:

```

1 <?php
2 /* Einbinden potentieller Bibliotheken. */
3
4 class filehandler {
5     /* Potentielle Variablen. */
6     private $languageFile
7     private $statFile
8
9     public function __construct() {
10        /* Initialisieren von $languageFile und statFile. */
11    }
12
13    public function openStatFile() {
14        /* Öffnen der Statistik Datei. */
15    }
16
17    public function uploadStatFile() {
18        /* Hochladen der Statistik Datei. */
19    }
20
21    public function checkExistingFile() {
22        /* Dateien auf Existenz prüfen. */
23    }
24
25    public function uploadLanguageFile() {
26        /* Hochladen der angegebenen Sprachdatei. */
27    }
28 }

```

Durch diese Überlegungen können alle Symptome beseitigt werden.