

Entwicklung webbasierter Anwendungen

Klausur vom 10.07.2017

Prof. Dr. R. Hahn, Prof. Dr. B. Kreling

Name	Vorname	Matrikelnummer
------	---------	----------------

Das Skript, Beispielcode und persönliche Notizen auf Papier sind als Hilfsmittel erlaubt, aber keine elektronischen Geräte. Schreiben Sie auf jedes Lösungsblatt zuerst Ihren Namen! Lösungsblätter ohne Namen werden von der Aufsicht konfisziert.

Bearbeitungszeit 100 Minuten (14:15 bis 15:55 Uhr);
insgesamt 60 Punkte in 4 Aufgaben.

Aufgabe:

Schreiben Sie eine Webanwendung für ein einfaches Online-Quizspiel. Dabei werden Fragen gestellt und 3 mögliche Antworten angeboten. Ziel des Spiels ist es, möglichst viele Fragen hintereinander korrekt zu beantworten. Das Spiel ist beendet, wenn eine Frage falsch beantwortet wird. Die Anzahl der korrekten Antworten kann über eine andere Webseite abgefragt werden, die aber nicht Bestandteil der Aufgabe ist.

Die Quizfragen sind gegeben in der Datenbanktabelle **fragen**. Jede Tabellenzeile beinhaltet eine Frage und 3 Antworten. Die korrekte Antwort ist in der Spalte 'richtig' vermerkt.

In der Tabelle **spiele** sollen Sie für jedes Spiel einen Datensatz mit dem Namen des Spielers anlegen und die Anzahl der richtigen Antworten in diesem Datensatz hochzählen.

Die Datenbank hat diese Struktur und Inhalte dieser Art:

```
CREATE TABLE fragen (  
  id int AUTO_INCREMENT, PRIMARY KEY (id),  
  frage varchar(50) NOT NULL,  
  antwort1 varchar(50) NOT NULL,  
  antwort2 varchar(50) NOT NULL,  
  antwort3 varchar(50) NOT NULL,  
  richtig enum('1', '2', '3') NOT NULL);
```

```
CREATE TABLE spiele (  
  id int AUTO_INCREMENT, PRIMARY KEY (id),  
  name varchar(50) NOT NULL,  
  antworten int NOT NULL);
```

id	frage	antwort1	antwort2	antwort3	richtig
1	Wie viele Einwohner hat Darmstadt?	155.000	205.000	75.000	1
2	Was bedeutet das html-Tag <td>?	test data	test dummy	table data	3
3	Wie viele Einwohner hat Mainz?	155.000	210.000	295.000	2

id	name	antworten
1	Ralf	3
2	John Doe	1
3	Hurz	23

'2' bedeutet:
'Antwort 2 ist richtig'

Lesen Sie zuerst die gesamte Aufgabenstellung. Schreiben Sie dann die fehlenden Methoden der bereits teilweise vorgegebenen Seitenklasse **MEISTER** (siehe Anhang). Die bekannte Klasse **Page** ist ebenfalls vorgegeben.

Der Code muss vollständig und sicher sein und alle gegebenen Daten korrekt darstellen. Er muss Exception Handling einsetzen und den Syntaxregeln der Veranstaltung entsprechen.

1. HTML / PHP / MySQL

21 Punkte

Generieren Sie die oben abgebildete HTML-Seite gemäß der Seitenklassenarchitektur. Der Titel der Webseite soll "Quizmeister" lauten. Die Quizfragen wählen Sie zufällig aus der Tabelle **fragen**: `SELECT xxx FROM yyy ORDER BY RAND() LIMIT 1` liefert eine zufällige Tabellenzeile. Ignorieren Sie das Risiko, dass die gleiche Frage mehrfach gestellt wird. Beim ersten Aufruf der Seite wird zusätzlich zur ersten Frage ein Block angezeigt, der den Namen des Spielers abfragt, damit das neue Spiel unter seinem Namen abgespeichert werden kann.

Tipp: realisieren Sie die Zuordnung der Antworten zum richtigen Spiel mit einer Session.

Wer wird h_da Quizmeister?

Neues Spiel - bitte geben Sie Ihren Namen ein:

Ihr Name?

Frage:

Was bedeutet das html-Tag <td>?

test data

test dummy

table data

2. PHP / MySQL

20 Punkte

Die Seite schickt ihre Daten (d.h. die gewählte Antwort und beim ersten Aufruf den Spielernamen) an sich selbst. Implementieren Sie die Verarbeitung der übermittelten Daten:

- Legen Sie beim ersten Aufruf einen Datensatz in der Tabelle **spiele** an. Speichern Sie den Namen (er darf nicht leer sein!) und initialisieren Sie den Zähler 'antworten'.

Tipp: die Eigenschaft `$mysqli->insert_id` liefert den letzten Wert eines `AUTO_INCREMENT`.

- Prüfen Sie, ob die gewählte Antwort richtig ist:
 - falls ja, inkrementieren Sie den Zähler 'antworten' im Datensatz und zeigen Sie eine weitere Frage an.
 - falls nein, zeigen Sie nur den Text "leider falsch" an und beenden das Spiel (der nächste Aufruf startet ein neues Spiel).

3. CSS

8 Punkte

Die Styles sollen in der Datei "MEISTER.css" definiert werden (siehe Einbindung in der class Page im Anhang).

Legen Sie folgende Merkmale über Styles fest (andere Eigenschaften müssen Sie nicht setzen):

- die Schrift ist Times New Roman
- die Fett- oder Kursivschrift und die Textausrichtung
- die Schriftgröße abhängig von der Browserschriftgröße
- den Rahmen um die beiden Blöcke
- den grauen Hintergrund
- das Aussehen der Knöpfe mit den Antworten (je nach Breite des Fensters sollen die Buttons nebeneinander oder untereinander dargestellt werden, damit das Quiz auch auf einem Smartphone funktioniert)

4. JavaScript

11 Punkte

Realisieren Sie das Skript als separate Datei "MEISTER.js" (der Link zur Einbindung ist schon vorhanden; siehe auch `<body ...>`).

- Der Spieler soll nur 30 Sekunden zur Beantwortung der Frage haben. Dazu wird am unteren Ende der Seite ein Fortschrittsbalken eingebunden: `<progress id="myProgress" value="0" max="30"></progress>` (Sie müssen den Code nicht abschreiben). Ein value von 0 ergibt einen leeren Balken. Schreiben Sie den Code, der den Progressbar in 30 Schritten füllt.
- Wenn innerhalb der 30 Sekunden keine Antwort gewählt wurde, wird die Seite abgeschickt. Das Spiel ist dann beendet.

Vorgegebener Code: PHP class Page und ein Teil von class MEISTER

Der folgende Code ist etwas abgewandelt gegenüber dem Praktikum. Ihre Lösung muss ihn unverändert verwenden; Sie brauchen ihn natürlich nicht abzuschreiben:

```
abstract class Page
```

```
{
    protected $db;

    protected function __construct() {
        $this->db = new MySQLi("localhost", "usr", "pwd", "MEISTER");
        if (mysqli_connect_errno())
            throw new Exception(mysqli_connect_error());
        if (!$this->db->set_charset("utf8"))
            throw new Exception($this->db->error);
    }
}
```

```
protected function __destruct() {
    $this->db->close();
}
```

```
protected function generatePageHeader($headline = "") {
    header("Content-type: text/html; charset=UTF-8");
    $headline = htmlspecialchars($headline);
    echo <<<EOT
    <!DOCTYPE html>
    <html lang="de" >
    <head>
    <meta charset="UTF-8" />
    <title>$headline</title>
    <link rel="stylesheet" type="text/css" href="MEISTER.css"/>
    <script type="text/javascript" src="MEISTER.js"></script>
    </head>
    <body onload="start();">
    EOT;
```

```
}
protected function generatePageFooter() {
    echo <<<EOT
    </body>
    </html>
    EOT;
}

protected function processReceivedData() { }
```

```
require_once 'Page.php';
class MEISTER extends Page
{
    public static function main()
    {
        try {
            $page = new MEISTER();
            $page->processReceivedData();
            $page->generateView();
        }
        catch (Exception $e) {
            header("Content-type: text/plain");
            echo $e->getMessage();
        }
    }
}
MEISTER::main();
```