

Klausur webbasierte Anwendungen

SS 2013

Angaben zur Person

Vorname Nachname

Matrikelnummer Unterschrift

Studiengang ggf. Codewort
(zum Wiederfinden der Note in der Notenliste)

Dauer: 90 min

Hilfsmittel: keine

Bewertung

Aufgabe	1	2	3	4	5	6	7	Summe
mögliche Punkte	11	15	5	4	4	10	14	63
erreichte Punkte								

Bemerkung

- Kontrollieren Sie die Klausur auf Vollständigkeit (7 Aufgaben, 10 Seiten) und einwandfreies Druckbild.
- Vergessen Sie nicht, auf diesem Deckblatt die Angaben zur Person zu machen und Ihren Namen auf jedes einzelne Blatt zu schreiben.
- Schreiben Sie Ihre Lösung jeweils auf das Blatt mit der Aufgabenstellung, Sie dürfen auch die Rückseite benutzen.
- Sollten Sie zusätzlichen Platz für Ihre Lösungen benötigen, finden Sie am Ende der Klausur zwei Zusatzseiten. Sollten Sie diese benutzen, so verweisen Sie bitte bei der Aufgabe darauf und schreiben Sie auch auf diese Blätter Ihren Namen.

Viel Erfolg!!

Name:

1. Die Volkshochschule möchte ein Online-Portal zur Kursanmeldung entwickeln. Sie haben die Aufgaben, ein Anmeldedialog zu entwerfen, mit dem sich die Teilnehmer für einen Kurs anmelden können. In dem Formular sollen die Wochentage und die Kurse ausgewählt werden können, für die man sich anmelden möchte. Damit das Formular nicht missbraucht werden kann, ist auch ein Benutzername und ein Passwort mit anzugeben.

Setzen Sie Ihr Formular in HTML5 um. Geben Sie als Lösung nur das Formular an. Die Felder sollen per „post“ an die Datei „vhs.php“ gesendet werden. (11 Punkte)

Name:

2. Sie haben eine MySQL-Datenbank mit folgenden Tabellen:

```
CREATE TABLE `vhsbenutzer` (  
  `benutzerid` varchar(20) NOT NULL,  
  `pwd` varchar(40) NOT NULL,  
  PRIMARY KEY (`benutzerid`)  
);  
CREATE TABLE `vhskurs` (  
  `tag` varchar(2) NOT NULL,  
  `kursname` varchar(20) NOT NULL,  
  `maxteilnehmer` int(2) NOT NULL,  
  PRIMARY KEY (`tag`, `kursname`)  
);  
CREATE TABLE `vhsreservierung` (  
  `tag` varchar(2) NOT NULL,  
  `kursname` varchar(20) NOT NULL,  
  `benutzerid` varchar(20) NOT NULL,  
  PRIMARY KEY (`tag`, `kursname`, `benutzerid`)  
);
```

Die Tabelle `vhskurs` ist mit folgenden Werten gefüllt:

```
INSERT INTO `vhskurs` VALUES ('Mo', 'Nähen', '8'),  
                                ('Mo', 'Englisch', '20');
```

Die Tabelle `vhsbenutzer` ist mit folgenden Werten gefüllt:

```
INSERT INTO `vhsbenutzer` VALUES ('Hans', 'abc'),  
                                    ('Fritz', 'def');
```

Weiterhin existiert die folgende Klasse `VHS`, um Einträge in den Tabellen einzufügen und abzufragen.

```
class VHS {  
  private $db;  
  public function __construct() {  
    // DB-Server verbinden und Datenbank auswählen  
    try {  
      $this->db = new PDO ("mysql:host=localhost;dbname=vhs",  
                          "vhs", "");  
      $this->db->setAttribute(PDO::ATTR_ERRMODE,  
                             PDO::ERRMODE_EXCEPTION);  
    } catch (PDOException $e) {  
      throw new Exception ('Verbindung nicht moeglich: ' .  
                            $e->getMessage());  
    }  
  }  
}
```

Name:

```
public function pruefeLogin ($benutzerid, $pwd) {
    $vhsbenutzer = -1;
    // Ist Benutzer in DB?
    $stmt=$this->db->prepare('SELECT count(*) AS anzahl
        FROM vhsbenutzer WHERE benutzerid=:benutzerid
        AND pwd=:pwd');
    $stmt->bindParam(':benutzerid',$benutzerid);
    $stmt->bindParam(':pwd',$pwd);
    $stmt->execute();
    $vhsbenutzer = $stmt->fetch(PDO::FETCH_OBJ);
    // Ergebnis zurückgeben
    return ($vhsbenutzer->anzahl);
}

public function loescheKursReservierung ($tag, $kursname) {
    // Lösche alle Teilnehmer aus Kurs
    $stmt=$this->db->prepare('DELETE FROM
        vhsreservierung WHERE tag=:tag AND
        kursname=:kursname');
    $stmt->bindParam(':tag',$tag);
    $stmt->bindParam(':kursname',$kursname);
    $stmt->execute();
}
}
```

Entwickeln Sie eine Methode `reserviereKurs ($tag, $kursname, $benutzerid, $pwd)`, die eine Reservierung für einen Teilnehmer vornimmt. Das Rückgabergebnis soll jeweils ein String sein. Folgende Funktionalität ist gefordert: (15 Punkte)

- Prüfen Sie die Anmeldedaten. Ist die Anmeldung falsch, geben Sie „Anmeldedaten falsch“ zurück.
- Prüfen Sie, ob ein Kurs an dem Tag vorhanden ist. Ist dies nicht der Fall, geben Sie „Kurs nicht gefunden“ zurück.
- Prüfen Sie ob noch ein freier Platz im Kurs vorhanden ist. Ist dies nicht der Fall, geben Sie „Der Kurs ist leider schon voll“ zurück.
- Im Erfolgsfall geben Sie „Sie sind erfolgreich angemeldet“ zurück.

Name:

```
public function reserviereKurs ($tag, $kursname,  
                                $benutzerid, $pwd) {
```

```
}
```

Name:

3. Schreiben Sie ein Programm, dass Ihr Formular auswertet und die Methode `reserviereKurs` der Klasse `VHS` aufruft. Die Textmeldungen können einfach mit `echo` ausgegeben werden. (5 Punkte)

4. Zur Erhöhung der Sicherheit von Formularen in Web-Anwendungen können *whitelist* oder *blacklist* Verfahren eingesetzt werden. Beschreiben Sie kurz beide Verfahren. Welches Verfahren ist zu bevorzugen und warum? (4 Punkte)

Name:

5. Für Angreifer ist es u.a. möglich, Sessions zu klauen. Erläutern Sie stichpunktartig ein Verfahren, mit dem Sessions **sinnvoll** geschützt werden können. (4 Punkte)

6. Nennen und erläutern Sie stichpunktartig fünf Regeln, wie Texte im Web **formuliert** (nicht formatiert) werden sollen. (10 Punkte)

Name:

7. Folgendes Formular enthält in Bezug auf die Web-Ergonomie mindestens sieben unterschiedliche Fehler. Nennen Sie die Fehler und erklären Sie stichpunktartig wie es richtig ein muss. Das Feld Geschlecht stellt ein Textfeld dar. (14 Punkte)

Schlafstudie

Geschlecht:

Alter:

0-9
10-19
20-29
30-39
40-49
50-59

▼

Schlafgewohnheit:

Langschläfer
Frühaufsteher
keine Angabe

▼

Name:

Zusatzseite 1

Musterklausur

Name:

Zusatzseite 2

Musterklausur