

|                      |              |                    |                           |
|----------------------|--------------|--------------------|---------------------------|
| Verteile Systeme     |              | Name               |                           |
| xx                   |              | Semester           |                           |
| Prof. Dr. S. Kruse   |              | Matrikel-Nr.       |                           |
| <i>Prüfungsart</i>   | Klausur      | <i>Hilfsmittel</i> | Code aus den Laborübungen |
| <i>Prüfungsdatum</i> | xx, xx.xx.xx | <i>Max. Punkte</i> | 100                       |
| <i>Prüfungsdauer</i> | 90 min.      | <i>Punkte</i>      |                           |
|                      |              | <i>Note</i>        |                           |

#### Hinweise:

- Die Klausur umfasst 5 Seiten mit 6 Aufgaben. Prüfen Sie bitte zuerst, ob Sie alle Blätter erhalten haben.
- Geben Sie auf allen Lösungsblättern Ihren Namen, das Semester und die Matrikelnummer an.
- Falls Sie zusätzliches Papier benötigen, melden Sie sich bitte.
- Schreiben Sie bitte deutlich, sauber und leserlich und mit einem dokumentenechten Schreibgerät, also nicht mit Bleistift.
- Benutzen Sie keine rote Farbe.
- Streichen Sie Ungültiges deutlich und eindeutig durch. Verwenden Sie kein Tipp-Ex, Korrekturband u. ä.
- Bieten Sie zu einer Fragestellung nicht mehrere Lösungen an, sondern entscheiden Sie sich für eine und streichen Sie die anderen durch.
- Lösungswege müssen nachvollziehbar sein. Die Angabe von Endergebnissen genügt nicht.
- Die zu verwendenden Programmiersprachen sind PHP und SQL. Es sind nur solche Lösungen erlaubt und erforderlich, die im Unterricht direkt behandelt wurden.
- Viel Erfolg!

**Achtung: Die Klausur umfasst 140 Punkte und damit mehr Umfang als eine reguläre Klausur mit 100 Punkten.**

## 1. Aufgabe (15 Punkte): Datenbankentwurf

Gegeben sei eine Relation Rennfahrer (Name, Team, Firmen) zur Verwaltung der Daten von Rennfahrern. Gegeben sei ebenfalls folgende Tabelle mit den Daten zu einigen Rennfahrern:

| <u>Name</u>          | Team                    | Firmen                                           |
|----------------------|-------------------------|--------------------------------------------------|
| Sebastian Nickel, G  | Bison Racing, A         | Bison, Ronault (Motor), Minelli (Reifen)         |
| Mark Reber, AUS      | Bison Racing, A         | Bison, Ronault (Motor), Minelli (Reifen)         |
| Richard Hamilton, UK | McLintock, UK           | McLintock, SMW (Motor), Minelli (Reifen)         |
| Torsten Rock, G      | Falling Star Racing, UK | Falling Star, Rosworth (Motor), Minelli (Reifen) |

Hinter den Rennfahrer- und Teamnamen ist jeweils noch die Abkürzung des Herkunftslands notiert. Unter den am Rennstall beteiligten Firmen neben dem Teambesitzer noch die Lieferanten für den Motor und die Reifen. Rennställe haben üblicherweise mehrere Fahrer unter Vertrag.

Überführen Sie die Relation schrittweise in die 1., 2. und 3. Normalform. Erläutern Sie dabei jeweils ihre Überführungsmaßnahmen. Die Tabelleneinträge mit den Daten zu einigen Rennfahrern brauchen nicht in ihre Lösung übernommen zu werden.

## 2. Aufgabe (25 Punkte): Schleifen, Felder, Zeichenketten

1. Gegeben seien die beiden folgenden assoziativen Felder:

```
$verein["Vorsitzender"] = "Müller";  
$verein["Kassenwart"] = "Meier";  
$verein["Mitglied"] = "Schmidt";
```

```
$alter["Müller"] = 52;  
$alter["Meier"] = 30;  
$alter["Schmidt"] = 24;
```

- Schreiben Sie eine `foreach`-Schleife, die aus diesen beiden Feldern das folgende zweidimensionale assoziative Feld `$gesamt` generiert:

```
$gesamt["Müller"]["Alter"] = 52;  
$gesamt["Müller"]["Funktion"] = "Vorsitzender";  
$gesamt["Meier"]["Alter"] = 30;  
$gesamt["Meier"]["Funktion"] = "Kassenwart";  
$gesamt["Schmidt"]["Alter"] = 24;  
$gesamt["Schmidt"]["Funktion"] = "Mitglied";
```

- Schreiben Sie eine geschachtelte `foreach`-Schleife, die das obige Feld `$gesamt` mit `echo` in das Browserfenster ausgibt.

2. Gegeben sei der folgende PHP-Code: `$kunde = "Müller, Peter";` mit exemplarischer Zeichenkette, die Informationen zu Vor- und Nachnamen enthält, wobei der Vorname in diesen Zeichenketten stets nach dem Nachnamen steht. Schreiben Sie PHP-Code nieder, der durch Auswerten der Zeichenkette in `$kunde` mit `echo` die folgende Ausgabe produziert:

Vorname: Peter , Nachname: Müller

Berücksichtigen Sie dabei, dass Vor- und Nachnamen in dieser Art von Zeichenketten unterschiedlich lang sein können.

|                    |              |  |
|--------------------|--------------|--|
| Verteile Systeme   | Name         |  |
| xx                 | Semester     |  |
| Prof. Dr. S. Kruse | Matrikel-Nr. |  |

### 3. Aufgabe (25 Punkte): Templates

1. Konstruieren Sie für die folgende Beispiel-HTML-Seite ein XTemplate mit iterierten Kontexten. Berücksichtigen Sie dabei, dass sowohl die Anzahl der Listen, als auch die Anzahl der jeweiligen Listeneinträge variabel sein kann.

```
<html>
  <body>
    <p>Liste 1:</p>
    <ul>
      <li>1a</li>
      <li>1b</li>
    </ul>

    <p>Liste 2:</p>
    <ul>
      <li>2a</li>
      <li>2b</li>
      <li>2c</li>
    </ul>

    <p>Liste 3:</p>
    <ul>
      <li>3a</li>
      <li>3b</li>
    </ul>
  </body>
</html>
```

2. Schreiben sie ebenfalls ein PHP-Script, welches das von ihnen konstruierte XTemplate mit den Daten der Beispielseite oben füllt. Die Daten werden direkt in der PHP-Datei eingetragen, d. h. es sind keine Datenbankzugriffe erforderlich.

## 4. Aufgabe (35 Punkte): Dynamischer Webauftritt

Sie entwickeln eine Online-Börse für Umzugsbeiladungen. Umzugsunternehmen können für Umzüge, bei denen die Transportfahrzeuge nicht ausgelastet sind, den Start- und Zielort sowie den Umzugstag und die Anzahl freier Quadratmeter in die Börse einstellen. Interessenten können in der Börse nach den verfügbaren Umzügen nach Start- und Zielort suchen.

1. Schreiben Sie die folgenden SQL-Befehle nieder:

(a) Benutzen einer Datenbank `boerse`.

(b) Erstellen einer Tabelle `umzuege` mit den Spalten `ID`, `start`, `ziel`, `tag` und `qm`.

Die Tabelleneinträge für `start`, `ziel` und `tag` dürfen maximal 50 Zeichen lang sein. Die Quadratmeter `qm` sowie die `ID` sind INT-Zahlenwerte.

(c) Füllen der Tabelle `umzuege` mit den folgenden Daten für Start, Ziel, Tag und Quadratmeter:

Ulm, Stuttgart, 04.08.11, 10

München, Ulm, 01.09.11, 20

2. Schreiben Sie die unten angegebene Webanwendung, die Eingabeformular und zugehörendes PHP-Script in einer Datei vereinigt.

Das Formular gestattet die Eingabe des Abfahrtsorts und/oder des Zielorts und der Mindestzahl erforderlicher Quadratmeter. Dabei können auch Teile der Ortsnamen eingegeben werden.

Es sind im Formular keine Tabelle und allgemein keine PHP-Templates, CSS oder Angaben zu Fonts, Ausrichtungen und Maßen sowie keine Fehlerbehandlung im PHP erforderlich.

Die Einträge im Formular in Abb. 1 unten sind nur Beispiele und keine Vorbelegungswerte. Der Vorbelegungswert des `qm`-Eintrags ist 0.

Die Ausgabe der Ergebnisse kann ohne CSS durch ein einfaches Fenster, das automatisch vom Browser geöffnet wird, erfolgen.

Ihr Benutzeraccount sei: Benutzername: `user`, Passwort: `pw`. MySQL-Server und Webanwendung seien auf dem gleichen Rechner installiert.

Der Fall, dass zu einer Anfrage keine Ergebnisse gefunden wurden, muss bei der Ausgabe nicht gesondert berücksichtigt werden, d. h. in diesem Fall kann ein Tabellenkopf wie in Abb. 2 rechts ausgegeben werden.

Beispiele:

start:

ziel:

qm:

Abbildung 1: Eingabeformular mit Beispielwerten

| Resultat(e): |       |           |          |    | Resultat(e): |       |      |     |    |
|--------------|-------|-----------|----------|----|--------------|-------|------|-----|----|
| Nr.          | Start | Ziel      | Tag      | qm | Nr.          | Start | Ziel | Tag | qm |
| 1.           | Ulm   | Stuttgart | 04.08.11 | 10 |              |       |      |     |    |

Abbildung 2: Ausgabe für die Eingabewerte aus Abb. 1 (links) und für das leere Suchergebnis (rechts)

## 5. Aufgabe (20 Punkte): Objektorientierte Programmierung

Schreiben Sie die folgenden PHP-Klassen:

1. Die Klasse `Roboter`. Roboter besitzen einen Namen, der beim Instanzieren der Klasse übergeben wird. Dieser Vorgang wird durch eine Ausgabe in den Webbrowser quittiert. Daneben besitzen Roboter die Methode `fahren()`, die beim Aufruf zusammen mit dem Namen des Roboters die Zeichenkette "fährt" in das Webbrowserfenster ausgibt. Siehe hierzu die Scriptausgabe unten.
2. Die Klasse `Staubsauger`. Staubsauger besitzen die Methode `saugen()`, die beim Aufruf die Zeichenkette "saugt" in das Webbrowserfenster ausgibt. Siehe hierzu die Scriptausgabe unten.
3. Definieren Sie auf der Grundlage von `Roboter` die Klasse `Saugroboter`. Saugroboter besitzen einen Staubsauger. Daneben besitzen sie die Methode `fahren()`. Zur Aktion Fahren gehört das eigentliche Fahren sowie das Saugen.
4. Erzeugen Sie einen Saugroboter mit dem Namen "Sam". und lassen Sie Sam danach fahren.

Vermeiden Sie Coderedundanz, indem Sie in übergeordneten Klassen die Eigenschaften der untergeordneten Klassen nutzen.

Ausgabe des Scripts:

Sam erzeugt

Sam fährt saugt

## 6. Aufgabe (20 Punkte): Cookies

Schreiben Sie eine PHP-Webseite `s1.php`, die bei Erstaufruf den Namen des Besuchers mit Hilfe eines Webformulars erfragt:

Bitte Namen eingeben:

Name:

Abschicken des Namens führt zum Aufruf einer Seite `s2.php`, die ein Cookie mit diesen Daten setzt und das Senden der Daten mit einer Dankesmeldung quittiert:

Danke für die Information!

Bei späteren Aufrufen von `s1.php` wird der Besucher jeweils mit seinem Namen begrüßt, sofern seit dem letzten Besuch nicht mehr als 30 Tage vergangen sind:

Hallo Erika Mustermann!

Anderenfalls wird der Seitenaufruf als Erstaufruf behandelt und die Daten werden erneut erfragt. Lösen Sie das Problem mit Hilfe eines Cookies.

Der im Formular der Abbildung oben dargestellte Name ist kein Vorbelegungswert, sondern nur eine Testeingabe.