

Matura-Vorbereitung - Webprogrammierung: Online Einkaufsliste

Eine Webanwendung für eine einfache, datenbank-basierte Online-Einkaufsliste soll implementiert werden. In dieser Anwendung können Einkäufe eingetragen und Online (z.B. mit dem Smartphone) abgerufen werden.

Aufgabe 2.1.: Erweitere die folgende Klasse *DBManager* um 2 Methoden (Bitte soweit als möglich bzw. sinnvoll *PreparedStatements* verwenden):

- *void loescheProdukte(int shopId)*
 - Es sollen alle Produkte eines Shops mit der *shopId* gelöscht werden.
- *List<Produkt> holeProdukte(String s)*: Die Produkte, die im Namen den String *s* beinhalten sollen aus der Datenbank gelesen und in einer Liste gespeichert werden.

Die entsprechenden Klassen sind folgendermaßen aufgebaut:

```
public class Produkt
{
    int id;
    String name;
    String amount;
    Shop shop;

    public Produkt() {}

    public Produkt(int id, String name, String amount, Shop shop)
    {
        this.id = id;
        this.name = name;
        this.amount = amount;
        this.shop = shop;
    }

    //zu jedem Attribut existieren Getter- und Setter-Methoden
}

public class Shop
{
    int id;
    String name;

    public Shop() { }

    public Shop(int id, String name)
    {
        this.id = id;
        this.name = name;
    }
}
```

Aufgabe 2.2.: Erstelle eine JSP-Seite *aendern.jsp*, welche das Ändern eines Produktes ermöglicht. Es müssen also alle wesentlichen Daten für die Tabelle *Produkt* über dieses Formular änderbar sein. Die Werte eines bestimmten Produktes werden im Servlet unter Aufgabe 2.3 geladen und in der Session gespeichert.

Hinweise:

- Ein Dropdown-Menue für eine Auswahl der Shops wird in HTML wie in diesem Beispiel zu sehen erstellt:

```
<select>
  <option value="rot">Rot</option>
  <option value="gruen">Grün</option>
  <option value="blau">Blau</option>
</select>
```

- Die möglichen Shops sind bereits in der Session unter dem Namen "shops" als Typ *List<Shop>* gespeichert.
- Der Prolog und die Doctype-Definition ist bereits gegeben:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Aufgabe 2.3.: Die Daten, die in Aufgabe 2.2 benötigt werden sollen im Servlet *AendernServlet* aus der Datenbank geladen und in die Session gespeichert werden. Die id des Produktes wird als request-Parameter übergeben. Für das Laden des Produktes kann DIE BEREITS BESTEHENDE Methode *Produkt holeProdukt(int id)* der Klasse *DBManager* genutzt werden:

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
    // TODO Hier bitte die Funktionalität einfügen
}
```

Aufgabe 2.4.: Jetzt fehlt noch die Darstellung der einzelnen Produkte. Dafür sind 2 Teile notwendig:

- Ein Servlet *AnzeigeServlet*, welches alle Vorbereitungen für die Anzeige trifft. Hier bitte wieder folgende Methode implementieren:

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
    // TODO Hier bitte die Funktionalität einfügen
}
```

- Eine JSP-Seite *anzeige.jsp*, welche die Anzeige selbst vornimmt.

Die einzelnen Produkte sollen als Tabelle dargestellt werden. Es soll in jeder Zeile ein Produkt stehen. Eine Spaltenüberschrift soll anzeigen, um welchen Inhalt es sich handelt. Erzeuge eine weitere Spalte mit einer leeren div-Box mit eindeutiger id zusätzlich für *Aufgabe 2.5.*

Aufgabe 2.5.: Bei einem Klick auf ein Produkt sollen über die Produkt-Id aktuelle Preise in den einzelnen Shops abgefragt werden. Dazu wird ein Servlet aufgerufen (welches intern Webservices

der Shops aufruft), das die Preisinformationen einzelner Geschäfte als JSON in folgender Form zurückgibt:

```
[
  {
    "shop": "Hofer",
    "preis": 4.55
  },
  {
    "shop": "Merkur",
    "preis": 5.05
  }
]
```

Erweitere die Seite *anzeige.jsp* jetzt mittels *AJAX/JSON* so, dass diese Inhalte in einer Liste direkt beim Produkt angezeigt wird (also die leere *div*-Box bei der Anzeige des Produktes befüllt wird).

Der Kerninhalt der Funktion könnte so aussehen (Es reicht, den mit *TODO* markierten Bereich zu befüllen):

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200)
  {
    // TODO Hier bitte die Funktionalität einfügen
  }
};

xhttp.open("GET", "Servletname", true);
xhttp.send();
```

Aufgabe 2.6.: Wie würde das Servlet aussehen, dass die JSON-Texte von Aufgabe 2.5 erstellt?

Datenbank: “einkauf”

Diese benötigte Datenbank besteht bereits und wurde folgendermaßen erstellt:

```
CREATE TABLE Produkt
(
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(128),
  amount VARCHAR(64), /* Die Menge samt Einheit als Text */
  shop_id INT
);

CREATE TABLE Shop
(
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(128)
);

ALTER TABLE Produkt ADD FOREIGN KEY (shop_id) REFERENCES Shop(id);

INSERT INTO Shop(name) VALUES ('Hofer');
INSERT INTO Shop(name) VALUES ('Müller');
INSERT INTO Shop(name) VALUES ('MPreis');
```