

Nonstandard Datenbanken

Wintersemester 2011/2012

4. Übungsblatt

Aufgabe 1: JSP und JDBC-Datenbankanbindung mittels Java Beans

In dieser Aufgabe soll eine Website erstellt werden, welche nach erfolgreicher Eingabe von Zugangsdaten (*Name*, *Passwort*) den Benutzer mit seinem Namen begrüßt und den Inhalt der in HSQLDB hinterlegten Tabelle `Album` von Blatt 1 anzeigt. Achten Sie darauf, dass Ihre Tabelle `Album` folgendem Schema entspricht: `Album(Albumtitel, Interpret, Erscheinungsjahr, Bewertung)`. Zur besseren Trennung von Design und Logik bietet sich der Einsatz von Java Server Pages (JSP) und Java Beans an.

Implementieren Sie eine Java Bean für den Zugang zur Datenbank und zur Überprüfung der Zugangsdaten. Die zusätzlich zu implementierende JSP soll aus einem statischen HTML-Teil bestehen und alle dynamischen Inhalte über die Java Bean beziehen.

Wenn bei der Ausführung Fehler (Exceptions) auftreten, so sollen diese in der HTML-Seite mit ausgegeben werden. Das Schreiben in die Systemkonsole (`System.out`) bietet sich nicht an, da diese nicht vom Benutzer auf Clientseite gelesen werden kann. (10 Punkte)

Technische Hinweise:

- Achten Sie darauf, dass die JSP-Datei (*.jsp) im Ordner `WebContent` Ihres Eclipse-Projektes liegt.
- Zusätzlich ist es wichtig, dass die nach der Übersetzung Ihrer Java Bean-Klasse erzeugte class-Datei im Ordner `WebContent/WEB-INF/classes` abgelegt ist, da sonst die JSP diese Klasse nicht finden kann. Dieser Pfad kann dauerhaft in den Projekteigenschaften unter `Java Build Path -> Source -> Default output folder` festgelegt werden.

- Für den Datenbankzugriff ist es notwendig, dass die Datei `hsqldb.jar` dem Tomcat-Webserver bekannt ist. Eine einfache Möglichkeit besteht darin diese Datei in den Projekteigenschaften unter `Java Build Path -> Libraries` hinzuzufügen.

Aufgabe 2: Objektorientierte Modellierung

Die Musikalben-Datenbank soll erweitert werden, indem Interpreten, Plattenfirmen, Verkäufer (Plattenläden) und Adressen als zusätzliche Entitäten aufgenommen werden. Die Entitäten sind wie folgt beschrieben:

- **Album:** Ein Album hat einen Albumtitel, einen Interpreten, eine Trackliste, ein Erscheinungsjahr, eine Bewertung, wird unter einem Plattenlabel vermarktet und wird von Verkäufern verkauft.
- **Plattenlabel:** Ein Plattenlabel kann eine beliebige Anzahl von Alben von Künstlern produziert haben und beliebig viele Verkäufer beliefern. Das Plattenlabel besitzt zudem eine Adresse.
- **Trackliste:** Eine Trackliste besteht aus einer beliebigen Anzahl von Musiktiteln, die wiederum einen Namen, eine Spielzeit und eine eindeutige Tracknummer haben.
- **Interpret:** Ein Interpret kann eine beliebige Anzahl von Alben veröffentlicht haben und besitzt evtl. eine Kontaktadresse für Fans.
- **Verkäufer:** Ein Verkäufer wird von mindestens einem Plattenlabel beliefert und verkauft mindestens ein Album. Der Verkäufer besitzt ebenfalls eine Adresse.
- **Adresse:** Eine Adresse besteht aus einer Straße, einer Postleitzahl und einem Ort.

Plattenlabel, Interpret und Verkäufer besitzen zudem einen Namen. Modellieren Sie den dargestellten Sachverhalt objektorientiert mit Hilfe von Java-Klassen. Dabei sollen auftretende Referenzen zwischen den Klassen *Verkäufer*, *Interpret*, *Album* und *Plattenlabel* in beide Richtungen realisiert werden, d.h. z.B. ein *Album*-Objekt hat eine Komponente, die den Interpreten, der es geschaffen hat, kapselt und analog dazu kapselt ein *Interpret*-Objekt alle aufgenommenen Alben in einer Komponente. Als Listenstruktur soll *LinkedList* oder *Vector* aus dem Package *java.util* verwendet werden. Die Komponenten dürfen `public` deklariert werden, um Zugriffsmethoden zu sparen. (10 Punkte)

Senden Sie Ihre Quelldateien (*.java, *.jsp) und den Bytecode (*.class) per Email an den Übungsleiter.

Abgabetermin: Montag, den 14. November 2011 bis 14 Uhr im IFIS