

Datenbanksysteme II

Wintersemester 2008/2009

5. Übungsblatt

Aufgabe 1: 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)

Aufgabe 2: JSP-Programmierung

Ein Einzelhandelsunternehmen möchte seinen Mitarbeitern Kataloge von Herstellern im Internet zur Verfügung stellen. Dabei sollen diese Unterlagen vor unberechtigtem Zugriff geschützt sein.

- a) Entwickeln Sie mit Hilfe einer Java Server Page einen Authentifizierungsmechanismus, bei dem ein Login-String und ein Passwort mit Hilfe eines HTML-Formulars ('form'-Element) abgefragt werden. Für die Überprüfung der Eingaben wird eine Java-Bean herangezogen, die Zugang zu den korrekten Daten hat. Überlegen Sie sich dabei einen Mechanismus für die Parameterübergabe. Wenn die Eingaben nicht korrekt sind, soll der Anwender einen entsprechenden Hinweis erhalten und erneut das Eingabe-Formular zusammen mit der Anzahl falscher Eingaben erscheinen.

Hinweis: Für diese Aufgabe kann das Beispiel *NumGuess* des Tomcat Servers hilfreich sein. Eine ausführliche Beschreibung der JSP/Servlet Technologie ist unter <http://pdf.coreservlets.com/> zu finden. (5 Punkte)

- b) Um die Kataloge, die z.B. im .pdf oder .ps-Format vorliegen, herunterladen zu können, bietet es sich an, das Programm *Download4J* zu verwenden (unter <http://www.javazoom.net/jzservlets/download4j/download4j.html> bzw. auf unserer Web-Seite). Installieren Sie diese Java-Komponente in Ihrem Tomcat-Pfad und testen Sie die Beispiele.

Hinweis: Es gibt Installationshinweise in der mitgelieferten Dokumentation. Es sollte genügen, den Verzeichnis-Baum, der nach dem Auspacken der .gz-Datei entsteht, unter \$TOMCAT_ROOT/webapps/ in einem eigenen Verzeichnis zu speichern.

- c) Führen Sie für eine Lösung des Ausgangsproblems Ihren Authentifizierungs- und den Download-Mechanismus zusammen, indem Sie die entsprechenden Java Server Pages kombinieren. In der endgültigen Fassung soll dem Anwender nach einer Authentifizierung die Liste der Kataloge präsentiert werden, die er dann herunterladen kann.

(5 Punkte)

Senden Sie alle Lösungen gepackt als .tar.gz oder .zip per Email an Ihren Übungsgruppenleiter.

Abgabetermin: Montag, den 24. November, vor der Vorlesung.