Praktikum Datenbanken / DB2
Woche 9: GUI-Anbindung über JDBC

Betreuer: Gudrun Fischer, Tobias Tuttas, Camille Pieume
Raum: LF 230
Bearbeitung: 10., 11. und 13. Juli 2006

Datum	
Team (Account)	
Vorbereitung	
Präsenz	

Aktuelle Informationen, Ansprechpartner und Material unter:
`http://www.is.informatik.uni-duisburg.de/courses/dbp_ss06/index.html`

Hinweise:

- Scheine und Teilnahmebestätigungen werden für alle ausgestellt, die mindestens 8 Wochenziele erreicht haben. Als Ersatz für eine Woche kann auch die aktive Teilnahme an der Zusatzveranstaltung am Mittwoch, dem 12. Juli, von 10 bis 12 Uhr in LF 230 zum Thema SSicherheitängerechnet werden.
- AOS-Studierende können durch aktive Teilnahme an der Zusatzveranstaltung den 10. Bonuspunkt erreichen.
- Die Scheine und Teilnahmebestätigungen können voraussichtlich ab Mitte August im Sekretariat (LF 227) abgeholt werden. Der genaue Termin wird auf der Webseite bekannt gegeben.

Wochenziele

In dieser Woche verbinden wir erste Funktionalitäten der graphischen Benutzeroberfläche (GUI) über JDBC mit der Datenbank.

Aufgaben

Vorbereitung (Hausaufgaben)

Betrachtet zur Vorbereitung den folgenden Auszug aus dem GUI-Code:

Listing 1: DatabaseConnector.java

```
1  protected int createNewMedium(Connection conn, Medium medium)
2                  throws SQLException {
3
4          Statement stmt = null;
5          ResultSet rs = null;
6          int maxId = 0;
7
8          try {
9                  stmt=conn.createStatement();
10                 rs=stmt.executeQuery(
11                         "SELECT MAX(MediumID) FROM Medium");
12                 if (rs.next()) {
13                         maxId=rs.getInt(1);
14                 }
15         } finally {
16                 if (rs!=null)
17                         try {
18                                 rs.close();
19                         } catch (SQLException e) {
20                                 e.printStackTrace();
21                         }
22                 if (stmt!=null)
23                         try {
24                                 stmt.close();
25                         } catch (SQLException e) {
26                                 e.printStackTrace();
27                         }
28         }
29
30         maxId++;
31
32         PreparedStatement createMedium = null;
33         try {
34                 createMedium = conn.prepareStatement(
35                         "INSERT INTO Medium("+
36                         "MediumId,MediumTitel,Art,BesitzerId) "+
37                         "VALUES(?,?,?,?)");
38                 createMedium.setInt(1,maxId);
39                 createMedium.setString(2,medium.getTitel());
40                 createMedium.setString(3,medium.getArt());
41                 createMedium.setInt(4,getUserId());
42                 createMedium.execute();
43         } finally {
44                 if (createMedium!=null)
45                         try {
46                                 createMedium.close();
47                         } catch (SQLException e) {
48                                 e.printStackTrace();
49                         }
50         }
51         return maxId;
52 }
```

V1: Java-Klassen

(a) Welche verschiedenen Java-Klassen kommen in dem obigen Code-Fragment vor?

(b) Schlagt diese Klassen in der Java-API-Dokumentation nach: `http://java.sun.com/j2se/1.4.2/docs/api/index.html`
Schreibt Euch zu jeder Klasse außer `Medium` kurz auf, was sie repräsentiert, und wie man sie verwendet. (Die Klasse `Medium` ist nicht Teil der Java-API.)

(c) Was könnte eine `SQLException` sein?

(d) Was ist der Unterschied zwischen `Statement` und `PreparedStatement`?

(e) Wie erzeugt man ein Objekt vom Typ `PreparedStatement`?

(f) Was bewirken die Methoden `setString` und `setInt` in der Klasse `PreparedStatement`?

V2: Semantik der Methode `createNewMedium`

(a) Was geschieht in den Zeilen 4 bis 28?

(b) Warum wird die Variable `maxId` in Zeile 30 um 1 erhöht?

(c) Die Klasse `Medium` repräsentiert ein Medium in unserem Datenmodell. Über die Methoden `getTitel()` und `getArt()` kann man auf den optionalen Mediumstitel und die Art des Mediums (DVD oder VHS) zugreifen.

Was geschieht demnach in den Zeilen 32 bis 51?

(d) Warum ist es sinnvoll, der Methode `createNewMedium` eine bestehende Datenbank-Verbindung (`conn`) zu übergeben? Warum wird die Verbindung nicht einfach innerhalb der Methode aufgebaut und auch dort wieder geschlossen?

(e) Wie müsstet Ihr die Methode `createNewMedium` ändern, um sie an *Eure* Datenbank und *Euer* Datenmodell anzupassen?

Präsenz

P1: GUI-Code kopieren und kompilieren

Den Code für die graphische Benutzeroberfläche findet Ihr ab Montag, dem 10. Juli im Verzeichnis `/home/dbprak/java`. Kopiert alle Klassen in Euer eigenes `home`-Verzeichnis und kompiliert sie. Die Klassen gehören alle zur Package `dbprak06`

Hinweise:

- Die Klassen befinden sich im Unterverzeichnis `dbprak06`. *Das Verzeichnis muss mitkopiert werden.*

  Beispiel (Kopieren und Kompilieren in der Kommandozeile):

  ```
  cd
  cp -r /home/dbprak/java/dbprak06 ./dbprak06
  javac dbprak06/*.java
  ```

- Für Projekte, die aus mehreren Klassen bestehen, bietet sich die Entwicklungsumgebung `eclipse` an. Ihr könnt sie mit dem Befehl `eclipse` direkt aus Eurer Konsole heraus starten.

P2: Datenbank-Schnittstellenklasse anpassen

Die folgenden Teilaufgaben beziehen sich auf die Klasse `DatabaseConnector`, die die Schnittstelle zwischen GUI und Datenbank realisiert.

(a) Passt die Verbindungsdaten in der Klasse `DatabaseConnector` (ganz oben) so an, dass sich die Klasse mit `Eurer` Datenbank verbindet.

(b) Ergänzt die Methode `login`, um Anwender-ID, Passwort, Adresse und E-Mail zu einem gegebenen Nicknamen aus der Datenbank zu holen.

Im Code der Methode `login` ist die betreffende Stelle mit einem „TODO"-Kommentar markiert.

(c) Kompiliert die angepasste GUI und führt die Klasse `dbprak06.PrideVid` aus. Es sollte ein Login-Fenster erscheinen. Falls nicht, findet und behebt etwaige Fehler.

(d) Meldet Euch als Eva Mustermann (Nick „Ophelia", Passwort „rotkaepchen") an. Funktioniert die Anmeldung?

(e) Lasst Euch in der GUI all „Eure" Medien anzeigen.

(f) Gebt ein neues Medium ein.

Freiwillige Aufgaben zum Weiterdenken

F1: UML-Klassendiagramm zu den GUI-Klassen

Schaut Euch den Code der GUI-Klassen an. Zeichnet ein UML-Klassendiagramm mit den Vererbungs-, Aggregations- und Assoziationsbeziehungen. Welche Klassen entsprechen Entities in Eurem Datenmodell? Wo gibt es Unterschiede?

F2: Weiteren Code anpassen

Passt die restlichen SQL-Anweisungen und -Anfragen in der Klasse `DatabaseConnector` an Eure Datenbank und Euer Datenmodell an. Verändert dabei nicht die anderen Klassen. (Jede Veränderung der anderen Klassen entspräche einer Schnittstellenveränderung, die mit einem hypothetischen GUI-Entwickler-Team abgesprochen werden müsste.)

F3: Weiterentwicklung der GUI

Betrachtet die GUI kritisch:

(a) Welche Funktionalitäten sind zwar im Menü vorgesehen, aber noch nicht implementiert?

(b) Welche Funktionalitäten fehlen ganz, d.h. welche zusätzlichen Menüpunkte haltet Ihr für sinnvoll?

(c) Wie könnte man das Eintragen von Medien benutzerfreundlicher gestalten?

(d) Wenn der Anwender nicht den genauen Titel eines Films kennt, wird derzeit einfach ein neuer Film mit dem falsch eingegebenen Titel angelegt.

Beispiel:

Statt „Pirates of the Caribbean: The Curse of the Black Pearl" gibt der Benutzer „pirates of the caribbean" ein.

Dass hier dann unkritisch ein neuer Film angelegt wird, ist nicht wünschenswert. Wie könnte man dieses Problem lösen?

Zum Schluss ...
stoppt bitte Eure Datenbank-Manager-Instanz auf `salz`,
loggt Euch von `salz` und vom lokalen Rechner aus,
aber schaltet den Rechner nicht ab.