

Übungsblatt 3

Besprechung in der Vorlesung am 28.09.2006

0 Anlegen der Datenbank

Mit der Installation des Java EE 5 SDK haben Sie bereits eine Derby-Datenbank (<http://db.apache.org/derby/>) installiert. Bevor Sie dort allerdings Daten ablegen können, müssen sie zunächst einen entsprechenden Datenspeicher anlegen.

Starten Sie die Datenbank indem Sie den Eintrag „Start Java DB“ ausführen, den Sie in Ihrem Startmenü finden.

Compilieren und führen Sie anschl. die vorgegebene Klasse `CreateDB` aus; beachten Sie dabei, dass Sie die Bibliothek `derbyclient.jar` im Classpath verfügbar machen müssen.

1 Anlegen der Datenbanktabellen

Legen Sie die notwendigen Tabellen für die Klassen der Bibliotheksverwaltung an. Implementieren Sie dazu in den vorgegebenen Klassen (`Benutzer`, `Buch` und `Vormerkung`) die `createTable()`-Methoden, sodass diese eine Verbindung zur Datenbank aufbauen und die nötigen SQL-Statements absetzen.

Schreiben Sie eine Klasse `DBSetup`, die die `createTable()`-Methoden der genannten Klassen aufruft, um die Datenbanktabellen zu erzeugen.

Sollten Sie Tabellen nicht ganz korrekt angelegt haben und/oder es erneut versuchen wollen, sollten Sie die Tabellen zunächst mit `DROP TABLE` löschen, bevor Sie sie erneut anlegen können. Implementieren Sie diese Funktionalität am besten in den jeweiligen `dropTable()`-Methoden.

2 Anlegen von Objekten

1. Als Nächstes sollen Sie Objekte in der Datenbank ablegen. Erzeugen Sie dazu zunächst ein neues Benutzer-Objekt mit allen Attributen und speichern Sie es durch Aufruf der `insert()`-Methode ab.

Merken Sie sich das Passwort, das Sie für dieses Benutzer-Objekt gesetzt haben, Sie werden es später brauchen.

2. Wie in der Vorlesung beschrieben, ist es sehr mühselig, für viele ähnliche Objekte ähnliche SQL-Statements zu entwickeln. Daher benutzen wir für `insert()`-Methoden der verschiedenen Medien-Klassen Vererbung. Bereits in der Oberklasse `Medium` ist die `insert()`-Methode definiert. Sie ruft in ihrem Innern die Methode `protected abstract String[] constructKeyValues(String keys, String vals)` auf. Das SQL-Kommando `INSERT` kann in zwei Teile gegliedert werden: Im ersten Teil werden die betroffenen Spaltennamen genannt, im zweiten Teil die tatsächlichen Werte. Diese beiden Teile werden hier `keys` bzw. `values` genannt.

Ergänzen Sie die `insert()`-Methode der Klasse `Medium` so, dass sie vor dem Aufruf von `constructKeyValues(String keys, String vals)` in den Strings `keys` und `values` die notwendigen Spaltennamen bzw. Werte der Attribute der `Medium`-Klasse zusammenstellen. Nach dem Aufruf der Methode `constructKeyValues(String keys, String vals)` soll das Rückgabe-Array an Position 0 die `keys` enthalten und an Position 1 die `values`. Lesen Sie diese Variablen also nach dem Aufruf der Methode wieder ein und bauen Sie aus den beiden Strings ein SQL-Statement, das Sie über eine Datenbankverbindung absetzen.

Nun müssen Sie in den Unterklassen die Methode `constructKeyValues(String keys, String vals)` überschreiben. Aus der Oberklasse `Medium` erhalten Sie bereits die Daten für ein `Medium` als Parameter übergeben. Rufen Sie ggf. trotzdem zunächst diese Methode auch in der Oberklasse auf (Stichwort `super`), um alle Attribute der Vererbungshierarchie zu erhalten. Aus dem Ergebnis (String-Array) entnehmen Sie nun wieder die `keys` und `values` und fügen die für diese Klasse spezifischen Attribute hinzu. Anschließend verpacken Sie die beiden Strings in der richtigen Reihenfolge in ein String-Array und geben dieses als Rückgabewert zurück.

Beachten Sie, dass Strings von einzelnen Anführungszeichen (`'...'`) umschlossen werden müssen. Attribute, die nicht gesetzt sind (`null`-Referenzen), dürfen entweder gar nicht in die Datenbank geschrieben werden oder müssen mit dem Wert `NULL` (ohne Anführungsstriche) beschrieben werden.

3. Erweitern Sie die entsprechenden Klassen so, dass man bestehende Objekte ändern und die Änderungen auch in der Datenbank ablegen kann. Benutzen Sie dazu den SQL-Befehl `UPDATE`. Leider hat der `UPDATE`-Befehl eine andere

Syntax als `INSERT`, daher müssen Sie neben der Methode `update()` in der Klasse `Medium` auch noch die Methode `String constructUpdateSql(String sql)` in den Unterklassen implementieren. Hier können Spaltennamen und Werte allerdings „am Stück“ dem String `sql` hinzugefügt werden, sodass das umständliche Hantieren mit `keys` und `values` entfällt.

3 XML ⇔ DB Konverter

1. Lassen Sie die Buchobjekte aus der XML-Datei `Biblio.xml` einlesen und in der Datenbank abspeichern. Sie können dazu den Menüpunkt „Datei | importieren“ aufrufen. Dabei werden die in Aufgabe 2 von Ihnen implementierten `insert()`-Methoden benutzt.

Beim Programmstart müssen Sie sich dann mit der Benutzernummer und dem Passwort authentifizieren, die Sie in Aufgabe 2.1 vergeben haben.

2. Lesen die gespeicherten Objekte aus der Datenbank aus und lassen Sie Ihre Daten anzeigen.

Erweitern Sie dazu die entsprechenden Klassen um die notwendigen Methoden, ihre Attribute aus einem `ResultSet` der Datenbank zu lesen.

4 Suchfunktionalität

Schreiben Sie eine Methode (`findByStichwort(String stichwort)` in der Klasse `Medium`), mit der Sie nach einem Teil eines Buchtitels (Stichwort) suchen können.

Diese Methode kann dann über den Menüpunkt „Bestand | Durchsuchen“ aufgerufen werden, die Ergebnisse werden anschließend angezeigt.

5 Ausleih- und Rückgabefunktionalität

Implementieren Sie die Methoden in der Klasse `AusleihbaresMedium`, mit denen Bücher ausgeliehen, verlängert, zurückgegeben und vorgemerkt werden können. Wenn sich der Leihstatus eines Buches ändert, soll dieser auch in der Datenbank aktualisiert werden.

Nur Bücher, die ausgeliehen sind, können vorgemerkt werden. Ein Benutzer kann mehrere Bücher vormerken, ein Buch kann von mehreren Benutzern vorgemerkt werden. Leiht ein Benutzer ein von ihm vorgemerktes Buch aus, so wird seine Vormerkung gelöscht.

Leiht ein Benutzer ein Buch aus, so ist das Rückgabedatum zu berechnen und mit dem Leihdatum zu speichern. Die Standardleihdauer beträgt 14 Tage.

Gibt ein Benutzer ein ausgeliehenes Buch zurück, so wird das Rückgabedatum gespeichert und das Ausleiher-Attribut gelöscht.

6 Leihübersicht

Implementieren Sie die Methode `findByAusleiher(Benutzer ausleiher)` in der Klasse `AusleihbaresMedium`, mit der es möglich ist, alle Bücher zu finden, die ein Benutzer ausgeliehen hat.

Mit dem Menüpunkt „Konto | Ausleihen einsehen“ können Sie Ihre Methode für den Benutzer aus Aufgabe 2.1 überprüfen.