

Erstellen und Benützung von JDBC.

INSY - JDBC

Wolfgang Mair

19.11.2013

Inhaltsangabe

<i>Aufgabenstellung:</i>	2
<i>Arbeitsschritte:</i>	3
<i>Zeitaufwand Geschätzt:</i>	3
<i>Arbeitsprotokoll:</i>	3
<i>Arbeitsdurchführung:</i>	4
<i>Technologiebeschreibung:</i>	4
<i>Testdurchläufe:</i>	5
<i>Quellenangaben:</i>	6

Aufgabenstellung:

JDBC: Daten abfragen

Erstelle eine Anwendung in Java im Package "deinnachname" (bitte durch Nachnamen ersetzen), die es Benutzern erlaubt, beliebige Datentabellen zu durchsuchen.

Dein Programm soll Kommandozeilenparameter für den Server, den Benutzernamen, das Passwort und die zu verwendende Datenbank verarbeiten - d.h. dein Programm **muss** sich wie folgt aufrufen lassen:

```
java deinnachname.jdbcuel -h projekte.tgm.ac.at -u benutzer -p  
passwort -d datenbankname
```

Daraufhin soll eine Swing-Eingabemaske angezeigt werden. Diese enthält die gezeigten Eingabefelder vorausgefüllt mit den Werten aus den Parametern, sowie für einen SQL-Befehl und eine noch leere Tabelle (siehe dazu auch das Interface TableModel und seine implementierenden Klassen). Der Befehl wird von deinem Programm an das DBMS geschickt werden und die Ergebnisse in einer Tabelle angezeigt.

Achtung: damit eine Datenmanipulation ausgeschlossen wird, muss dein Programm den eingegebenen SQL-Befehl prüfen:

- es darf nur EIN Befehl sein
- es dürfen nur SELECT, SHOW oder DESCRIBE-Befehle verwendet werden INSERT, DELETE, CREATE ... müssen von deinem Programm mit einer sinnvollen Fehlermeldung abgelehnt werden.
- Strichpunkt vorhanden?

Liefert das RDBMS eine Fehlermeldung zurück, muss diese sorgfältig verarbeitet und dem Benutzer eine entsprechende Meldung gezeigt werden.

Alle, die diese (oder eine ähnliche) Aufgabe im letzten Jahr schon gemacht haben:

Erstelle eine GUI, die zusätzlich zu den Verbindungsparametern (Kommandozeile, Felder wie oben) ein Feld für eine Combobox für alle in der DB vorhandenen Tabellen zeigt. Wird eine vom Benutzer ausgewählt, zeigt euer Interface jeweils ein Feld für alle vorhandenen Attribute in dieser Tabelle. Gibt der Benutzer dort Daten ein, werden diese in einem neuen Datensatz in die Datenbank gespeichert.

Arbeitsschritte:

GUI

Ich möchte möglichst eine einfache Grafische Oberfläche generieren mit dem nötigsten(Eingabefelder und Buttons). Die Buttons sollen es einfacher machen einen genauen Zeitpunkt der Übernahme der Daten zu gewährleisten. Dabei ist die Größe der Textfelder ziemlich irrelevant.

CLI

Das Programm muss mit CLI gestartet werden können. Allerdings will ich nicht das es ein Muss ist, deswegen ermöglicht ich es den User Teile auszulassen somit wird dann nur ein Teil der Textfelder im Vorhinein ausgefüllt, dass CLI startet nicht automatisch die Verbindung mit der Datenbank.

JDBC

Die GUI soll sich durch bestimmte Eingaben mit einer Datenbank verbinden und Befehle ausführen. Es soll möglich sein sich zu trennen und in eine andere Datenbank zu verbinden.

Eingabe Kontrolle

Die Befehle die über die GUI eingegeben werden sollen überprüft werden. Es sollen nur SELECT, SHOW und DESCRIBE Befehle erlaubt werden. Dabei sollen auf Fehler wie; vergessen spezifisch aufmerksam gemacht werden.

Zeitaufwand Geschätzt:

Aufgabe	Geschätzt	Tatsächlich
GUI	1h	2 h
CLI	1h	¾ h
JDBC	3h	2h
Eingabe Kontrolle	3h	1 h
Debugging	4h	½ h
Gesamtzeit	12 h	6 ¼ h

Arbeitsprotokoll:

Aufgabe	Datum	Arbeitszeit
GUI	16.11.2013	1 ½ h
CLI	18.11.2013	¾ h
Debugging	18.11.2013	¼ h
JDBC	19.11.2013	2 h
Eingabe Kontrolle	20.11.2013	1 h
Debugging	20.11.2013	¼ h
GUI	20.11.2013	½ h
Gesamtzeit		6 ¼ h

Arbeitsdurchführung:

Erfolge:

- CLI War schnell erledigt durch die „Vorlage“ von Roboterfabrik war es schneller erledigt als erwartet (auch ohne Fehler bis auf Änderungen in der Panel Klasse).
- JDBC war einfacher und verständlicher als erwartet bis auf die Implementierung des Jar-Files (das habe ich erst herausfinden müssen).
- Eine Eingabe Kontrolle zu erzeugen war ziemlich einfach obwohl so eine IF Verschachtelung nicht sehr schön ist.
- Debuggen hat mir dieses Mal viel weniger Zeit gekostet

Niederlagen:

- Die GUI sieht nicht so aus wie ich wollte (Die Positionierung hat mir Probleme bereitet), Notdürftig genügt aber um weiter zu machen!
- Die Tabelle zu erzeugen ist sehr schwierig gewesen und hat mir noch einmal viel Zeit gekostet.[24]

Technologiebeschreibung:

CLI:

Ermöglicht das Starten des Programmes mit Anfangswerten. Man kann bestimmen ob diese Anfangswerte angegeben werden müssen oder nicht.

JDBC:

JDBC (Java Database Connector) ermöglicht das verbinden eines Programmes mit einer Datenbank. Es ist außerdem möglich mittels JDBC Veränderungen auf der Datenbank zu übermitteln.

Testdurchläufe:

Aufruf mittels CLI Eingabe: `-h projekte.tgm.ac.at -u benutzer -p passwort -d datenbankname` funktioniert.

Aufruf mittels CLI Eingabe: `-h projekte.tgm.ac.at -u benutzer -d datenbankname` funktioniert.

Aufruf mittels CLI Eingabe: `-h projekte.tgm.ac.at` funktioniert.

Aufruf mittels CLI Eingabe: `[LEER]` funktioniert.

Verbinden zur Datenbank mittels eingaben in dem TextFeld und klicken auf Verbinden ... funktioniert

Verbinden zur Datenbank mit falschen Eingaben ... Fehlermeldung taucht in einem Popup Fenster auf.

Ausführen von Befehlen mit richtigen Angaben: `Select * from [tabelle];` Funktioniert.

Ausführen von Befehlen mit falschen Angaben: `Select * from; [tabelle];` Fehlermeldung.

Ausführen von Befehlen mit falschen Angaben: `Select * from [tabelle]` Fehlermeldung.

Ausführen von Befehlen mit falschen Angaben: `Drop table [tabelle];` Fehlermeldung.

Ausführen von Befehlen mit richtigen Angaben: `desc [tabelle];` Funktioniert.

Ausführen von Befehlen mit richtigen Angaben: `show tables;` Funktioniert.

Ausführen von Befehlen mit fast richtigen Angaben: `Select * from [tabelle]; drop table [tabelle];` Fehlermeldung.

Quellenangaben:

[24] mKorbel, zuletzt bearbeitet: 14.10.2013, zuletzt geöffnet: 20.11.2013

<http://stackoverflow.com/questions/19365113/defaulttablemodel-in-a-jframe-only-creating-on-row-java>