

Programmieren 2

Aufwärmübung JDBC

1. JDBC – Ergänzende Literatur

- <http://docs.oracle.com/javase/tutorial/jdbc/index.html>
- <http://openbook.galileocomputing.de/javainsel/index.html> (Lehrbuch, Kapitel 17)

2. JDBC – Zugriff auf Beispieldatenbank (Java DB)

Im JDK ist die von Oracle unterstützte Apache Derby Opensource Datenbank enthalten. Nachfolgend greifen wir mit dem Programm `SimpleDBApp` (siehe Moodle) auf Tabelleninhalte zu.

- Importieren Sie `SimpleDBApp.zip` über `File -> Open Projects from File System -> Archive`.
- ODER: Legen Sie ein neues Projekt `JDBC_Uebung` an. Danach fügen Sie unter `src` ein Package `jdbc_uebung` und darunter eine Klasse `SimpleDBApp` ein. Kopieren Sie den im Moodle verfügbaren Programmcode in die vorbereitete Klasse `SimpleDBApp`.
- Führen Sie das Programm aus. Es erscheint eine Fehlermeldung, dass die Treiber-Klasse fehlt.
- Öffnen Sie das Kontextmenü für das Eclipse-Projekt `JDBC_Uebung` (Rechtsklick) und wählen Sie `Properties`. Anschließend öffnen Sie `Java Build Path` und dort den Reiter `Libraries`. Betätigen Sie die Schaltfläche `Add Library` und wählen Sie den Eintrag `Connectivity Driver Definition` aus.
 - Klicken Sie auf `New Driver Definition` (Hinweis: Erster Button rechts von der Drop-Down-Box für verfügbare Treiber).
 - Wählen Sie den Treibernamen `Derby`
 - Markieren Sie den `Derby Embedded JDBC Driver` mit der höchsten Versionsnummer.
 - Klicken Sie jetzt auf den Reiter `JAR List`, betätigen Sie `Add JAR/ZIP...` und navigieren Sie zum Verzeichnis `jdk...\db\lib\derby.jar`
 - Hinweis zur Fehlersuche: Wenn Sie bereits eine `derby.jar` in Eclipse angezeigt bekommen, die aber nicht funktioniert, müssen Sie sie löschen.
- Legen Sie die Datenbank an:
 - Öffnen Sie die Eclipse Perspektive `Database Development` (`Window -> Open Perspective -> Other`).

3. Anlegen einer Datenbank

Legen Sie eine neue Datenbankverbindung an. Wählen Sie `Derby` als `Connection Profile` aus.

- Geben Sie als „Database Location“ den Pfad „<...>\MyDB\sample“ an. Als Username und Password wählen Sie „app“. Das Verzeichnis MyDB und das Verzeichnis sample dürfen nicht bereits existieren.
- Testen Sie die Verbindung.

- Es sind in der neuen Datenbank noch keine Tabellen angelegt. Klicken Sie deshalb in der Java-Perspektive rechts auf die Datei `customer.sql` im Projekt SimpleDBApp. Führen Sie die SQL-Befehle über „Execute SQL“ aus.
- Über den Data Source Explorer können Sie jetzt den Inhalt der Tabelle CUSTOMER im Schema APP betrachten.
- Beachten Sie, dass bei der in dieser Übung verwendeten Systemkonfiguration nur ein Programm eine Verbindung zur Datenbank aufbauen kann. Sie müssen jetzt die vom Data Source Explorer aufgebaute Datenbankverbindung lösen (Rechtsklick auf das Datenbanksymbol), damit das Java Programm eine Verbindung aufbauen kann.
- Ersetzen Sie in SimpleDBApp.java den Pfad zur Datenbank mit den korrekten Werten für die DB, die Sie eben erstellt haben. Führen Sie das Programm SimpleDBApp.java erneut aus.

4. Interagieren mit einer Datenbank

Ermitteln Sie nun die nötigen SQL-Kommandos für die folgenden Aktionen:

- Anzeigen aller Kunden in der DB (SQL-SELECT)
- Anzeigen Kunden mit einer bestimmten Eigenschaft (z.B. Name beginnt mit „B“, Customer_ID größer als 12345, beide Eigenschaften gleichzeitig, ...)
- Ändern von bestimmten Kunden-Einträgen (SQL-UPDATE)
- Löschen der Kunden-Daten (SQL-DELETE)