Inhaltsverzeichnis

1. Datenbanken.JDBC

Java-Anwendungen greifen auf SQL-Datenbanken über einen JDBC-Treiber zu (Java DataBase Connectivity). Dadurch kann der Java-Source-Code weitgehend datenbankunabhängig gehalten werden, so dass ein späterer Wechsel der SQL-Datenbank leicht möglich ist. Genaueres zu JDBC erfahren Sie unter http://java.sun.com/products/jdbc.

In den meisten Fällen sind JDBC-Type-4-Treiber optimal. Sie sind sehr schnell und sehr einfach zu installieren. Die Unterschiede zwischen den JDBC-Typen sind erklärt unter http://java.sun.com/products/jdbc/driverdesc.html.

Einen zu Ihrer Datenbank passenden JDBC-Treiber finden Sie am leichtesten unter http://industry.java.sun.com/products/jdbc/drivers.

Quelle: http://www.torsten-horn.de/techdocs/java-sql.htm#JDBC (10.5.2007)

1.1. Ein erstes Beispiel

1. MySQL-JDBC-Type-4-Treiber (z.B. 'mysql-connector-java-5.0.3-bin.jar' aus 'mysql-connector-java-5.0.3.zip') downloaden von: http://www.mysql.com.
2. CLASSPATH muss JDBC-Treiber beinhalten (eventuell reicht Kopieren nach '%JAVA_HOME%\jre\lib\ext').
3. Connection (siehe unten 'Programmierbeispiele'):

```
Class.forName( "com.mysql.jdbc.Driver" );
cn = DriverManager.getConnection(
"jdbc:mysql://MyDbComputerNameOrIP:3306/myDatabaseName", sUsr, sPwd );
```

MySQLConnector in den jre-Pfad kopieren
zB: ??????????????????????????????\ext\mysql-connector-java-5.0.3-bin.jar

jdbc_mysql_bsp1.java

```
import java.sql.*;

public class jdbc_mysql_bsp1 {
	public static void main(String[] args) {
		// TODO Auto-generated method stub

	   try {
		  String sDbDrv="??????????????????????";

		  String sDbUrl="??????????????????????";
		  String sUsr="advent";
		  String sPwd="tiger";
		  String sTable="is_professoren";
```

```
            ????????????? conn;
            ????????????? stmt;
            ????????????? rslt;

            Class.?????????????( sDbDrv );

            conn = ??????????.?????????????( sDbUrl, sUsr, sPwd );
            stmt = conn.???????????????();
            rslt = stmt.????????????????("select persnr,name from " +
                                        sTable);

            System.out.printf("%6s:%-15s\n", "PERSNR","NAME");
           while( rslt.????????????() ) {
                System.out.printf("%6d:%-15s\n",
                     rslt.?????????(1),  rslt.?????????(2));
            }
            ??????????.?????????????();
        }
        catch( Exception e ) {
        }
    }

}
```

1.2. Ein zweites Beispiel: Zugriff auf Metadaten

Wenn man ein SQL-Statement der Art

select * from Tabelle

absetzt, möchte man wissen, wieviele Felder in der Resultset vorhanden sind. Dazu gibt es die Metadaten. Hier ein Beispiel dazu:

```
/*
 * MySQLConnector in den jre-Pfad kopieren
 * zB: C:\Programme\Java\jdk1.5.0\jre\lib\ext\mysql-connector-java-
5.0.3-bin.jar
 *
 * siehe auch:
 * http://www.torsten-horn.de/techdocs/java-sql.htm
 */

import java.io.*;
import java.sql.*;

public class jdbc_mysql_advent {

     public static void main(String[] args) {
          // TODO Auto-generated method stub
          String sDbDrv=null, sDbUrl=null, sTable=null,
          sUsr="", sPwd="";

          try {
               BufferedReader in = new BufferedReader(
               new InputStreamReader( System.in ) );
```

```
            System.out.println( "Name des Datenbanktreibers eingeben
                 (z.B. com.mysql.jdbc.Driver):" );
            sDbDrv = in.readLine();

            System.out.println( "Url der Datenbank eingeben (z.B.
                 jdbc:mysql://localhost:3306/MeineDb):" );
            sDbUrl = in.readLine();

            System.out.println( "Name der Tabelle eingeben (z.B.
                 MeineTestTabelle):" );
            sTable = in.readLine();

            System.out.println( "Benutzername (z.B. root):" );
            sUsr = in.readLine();

            System.out.println( "Passwort (z.B. mysqlpwd):" );
            sPwd = in.readLine();
        } catch( IOException ex ) {
            System.out.println( ex );
        }

        if( null != sDbDrv && 0 < sDbDrv.length() &&
            null != sDbUrl && 0 < sDbUrl.length() &&
            null != sTable && 0 < sTable.length() ) {

          java.sql.Connection cn = null;
          java.sql.Statement  st = null;
          java.sql.ResultSet  rs = null;

          try {
            // Select fitting database driver and connect:
            Class.forName( sDbDrv );

            cn = DriverManager.getConnection( sDbUrl, sUsr, sPwd );
            st = cn.createStatement();
            rs = st.executeQuery( "select * from " + sTable );


            // Get meta data:
            ResultSetMetaData rsmd = rs.getMetaData();
            int i;
            int n = rsmd.getColumnCount();

            // Print table content:
            for( i=0; i<n; i++ ){
              System.out.print( "+---------------" );
            }
            System.out.println( "+" );

             // Attention: first column with 1 instead of 0
            for( i=1; i<=n; i++ )
              System.out.printf( "| %14s", rsmd.getColumnName( i ));

            System.out.println( "|" );

            for( i=0; i<n; i++ )
              System.out.print( "+---------------" );
```

```
                System.out.println( "+" );

                while( rs.next() ) {
                      // Attention: first column with 1 instead of 0
                  for( i=1; i<=n; i++ )
                    System.out.printf( "| %14s", rs.getString( i ) ) ;
                  System.out.println( "|" );
                }

                for( i=0; i<n; i++ )
                  System.out.print( "+---------------" );
                System.out.println( "+" );

              } catch( Exception ex ) {
                System.out.println( ex );
              } finally {
                try { if( null != rs ) rs.close(); }
                 catch( Exception ex ) {}
                try { if( null != st ) st.close(); }
                 catch( Exception ex ) {}
                try { if( null != cn ) cn.close(); }
                 catch( Exception ex ) {}
              }
            }
          }
}
```

1.3. Aufgaben

Aufgabe: mysql2html.java
Erstellen Sie ein Programm, das das Speichern einer Tabelle als HTML-Datei ermöglicht. Die Connect-Informationen sollen von der Konsole eingebbar sein. (s.o.)

1.4. Zusammenfassung

Gib eine Aufstellung der Klassen und Methoden (jdbc) und eine Zuordnung.

A) java.sql.Connection;
B) ResultSet
C) Statement
D) DriverManager
E) Class

1) createStatement();
2) next()
3) getConnection();
4) executeQuery();
5) forName()
6) getInt()
7) getString()

8) close()

Zuordnung:
A-1
?????????

1.5. Ausblick

- ODBC