

Datenbankzugriffe mit Java

1)

In diesem Beispiel soll der Zugriff auf eine MS-Access Datenbank realisiert werden. Hierbei wird nicht auf Design Wert gelegt, sondern nur die prinzipielle Funktionsweise demonstriert.

2)

Die Tabelle "Artikel" der Datenbank sieht so aus:

Feldname	Felddatentyp
Nr	Text
Bezeichnung	Text
Bestand	Zahl (Long Integer)
Preis	Zahl (Double)

Bemerkung:

In MSAccess kann man den Datentyp "Zahl" wählen.

Dann kann man im Fenster "Feldeigenschaften" im Feld "Feldgröße" den Datentyp Double oder Integer wählen.

3)

JDBC: Java DataBase Connectivity

ODBC: open DataBase Connectivity

Mit einem ODBC-Treiber kann man in einer Windows-Systemumgebung auf eine MS-Access Datenbank zugreifen. Um von Java aus über diesen Treiber auf eine MS-Access Datenbank zuzugreifen, braucht man einen JDBC-Treiber.

Dazu braucht man einen Treiber für die sogenannte JDBC-ODBC-Brücke.

Diese wird mit

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

geladen.

Nach dem Laden einer Klasse durch den Klassenlader werden die Anweisungen des sogenannten statischen Blocks (der mit dem Bezeichner static beginnt) ausgeführt. Dabei wird der Treiber initialisiert und registriert.

```

import java.sql.*;

public class MainDb10 {
    public static void main(String[] args) {
        String datenQuelle;
        Connection demoDB1Connection;
        Statement statementsSQL;
        ResultSet resultSet;
        String AbfrageSQL;
        Boolean b;
        String nr;
        String bezeichnung;
        int bestand;
        double preis;
        try{
            // Lädt die Klasse mit dem Namen "sun.jdbc.odbc.JdbcOdbcDriver"
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            // Zugriff auf die DB (Deklaration und Wertzuweisung) wird über
            // eine Datenquelle gemacht.
            // eine Datenquelle ist wie folgt aufgebaut:
            // jdbc:Subprotokoll:Datenquellennamen
            // Für ODBC-Datenquellen ist das Subprotokoll obcd
            // ACHTUNG: Den String nach datenquelle = in der Java-Datei
            // in eine Zeile schreiben !!!
            datenQuelle = "jdbc:odbc:Driver={Microsoft Access Driver
                (*.mdb)};DBQ=H:/Daten_Austausch/MEINE_SKRIPTE/
                ProgJava/22_Datenbank/PROG/db10/demoDB1.mdb";
            //obenstehender Pfad muss jeweils angepasst werden!
            // getConnection(...) liefert die Verbindung zur Datenbank zurück.
            // Dazu wird benötigt:
            // eine Datenquelle, einen User, ein Passwort
            demoDB1Connection = DriverManager.getConnection(datenQuelle, "", "");
            // Um Anfragen an die Datenbank stellen zu können benötigt man ein
            // Objekt der Klasse Statement
            statementsSQL = demoDB1Connection.createStatement();
            // Liefere von allen Datensätzen mit Nr = 1000 die
            // Nr, Bezeichnung, Bestand, Preis.
            // Wichtig: Zeichenketten müssen in SQL-Befehlen in einfachen
            // Hochkommata eingeschlossen werden
            AbfrageSQL = "SELECT Nr, Bezeichnung, Bestand, Preis FROM Artikel ";
            AbfrageSQL = AbfrageSQL + "WHERE Nr = '1000'";
            // AbfrageSQL ist also:
            // SELECT Nr, Bezeichnung, Bestand, Preis FROM Artikel WHERE Nr = '1000';
            System.out.println("AbfrageSQL="+AbfrageSQL);
            // Das Ergebnis der Abfrage befindet sich in resultSet.
            // Dieses Ergebnis kann man sich als Tabelle vorstellen, in der
            // die gefundenen Datensätze als Zeilen dargestellt werden.
            // Direkt nachdem das Ergebnis in resultSet abgeliefert wird,
            // steht der "Zeilenzeiger" _vor_ der ersten gefundenen Zeile.
            // (mit next() wird der Zeilenzeiger dann auf die erste gefundene
            // Zeile positioniert).
            resultSet = statementsSQL.executeQuery(AbfrageSQL);
            // Bei jedem Aufruf von next() wird der Zeilenzeiger um eine Zeile
            // nach unten bewegt. Bei ersten Aufruf von next() zeigt der
            // Zeilenzeiger also auf die erste gefundene Zeile.
            // Gibt es keine gefundene Zeile, liefert next() den Wert
            // false zurück.
            b = resultSet.next();
            if(b==true){
                nr = resultSet.getString("Nr");
                bezeichnung = resultSet.getString("Bezeichnung");
                bestand = resultSet.getInt("Bestand");
                preis = resultSet.getDouble("Preis");
                System.out.println("nr="+nr);
                System.out.println("bezeichnung="+bezeichnung);
                System.out.println("bestand="+bestand);
                System.out.println("Preis="+preis);
            }
        }
    }
}

```

```
        else{
            System.out.println("Datensatz wurde nicht gefunden");
        }

        resultSet.close();
        StatementSQL.close();
        demoDB1Connection.close();
    }
    catch (ClassNotFoundException fehler){
        System.out.println("Treiber existiert nicht");
        fehler.printStackTrace();
    }
    catch (SQLException fehler){
        System.out.println("DB konnte nicht geöffnet werden");
        fehler.printStackTrace();
    }
}
}
```