

Mit Java und SQL auf Datenbanken zugreifen

Endlich wird die Brücke zwischen zwei Welten geschlagen!!!

Vorüberlegungen

Es soll eine Java-Methode entwickelt werden, die die Räume zu einem Fach zurückgibt. Den Raum soll man als Parameter angeben können; die Rückgabeliste soll Raum-Nummer und -Etage enthalten

D.h. gesucht ist eine Methode

```
public List<String> raeumeFuerFach(String pFach)
```

Die **Datenbank-Abfrage**, die das ermöglicht, ist die folgende - hier aufgezeigt für das Fach "Deutsch":

```
SELECT r.nummer AS nummer, r.etape AS etape  
FROM raum r JOIN unterricht u  
ON r.id = u.raum_id  
WHERE u.fach LIKE '%Deutsch%'
```

Strategie

Die möglichen Fehlerbehandlungen werden in der Strategie nicht berücksichtigt.

1. Es gibt ein Attribut connector vom Typ DatabaseConnector
2. Im Konstruktor wird der connector erzeugt.
3. In der Methode raeumeFuerFach(String **pFach**) geht man so vor:
 1. SQL-Statement "zusammenbauen":
String sqlStatement = " SELECT ..."
Dabei den Parameter geeignet einbauen: "... LIKE '%" + **pFach** + "%' ...";
Das wird dann z.B. zu: "... LIKE '%Deutsch%' ..."
Doppelte Anführungszeichen: Java; einfache Anführungszeichen: SQL
 2. Das SQL-Statement mit System.out.println(sqlStatement); ausgeben:
Das ist wichtig zur Kontrolle!
 3. SQL-Statement ausführen:
connector.executeStatement(sqlStatement);
 4. Das Ergebnis der SQL-Abfrage ermitteln:
QueryResult queryResult =
connector.getCurrentQueryResult()
 5. Die Daten des Ergebnisses ermitteln:
String[][] data = queryResult.getData();
Man erhält ein 2-dimensionales Array.
Dabei ist die erste Dimension für die Zeilen, die zweite Dimension für die Spalten.
 6. Die Zeilen werden mit einer for-Schleife durchlaufen:
for (int i = 0; i < data.length; i++)
 7. Jetzt liest man die einzelnen Werte nach Spaltennummer aus, z.B. für die Oscars
String nummer = data[i][0];
 8. Man erhält grundsätzlich Objekte vom Typ String.
Evtl. muss man sie noch in einen richtigen Datentyp konvertieren, z.B. mit
int ganzeZahl = Integer.parseInt(ganzeZahlString);

Fertiger Quelltext

```
public class DatenbankTest {
    private DatabaseConnector connector;

    public DatenbankTest() {
        // ip port database user password
        connector =
            new DatabaseConnector("localhost", 3306, "schule", "root", "");
        String errorMessage = connector.getErrorMessage();
        if(errorMessage != null) System.err.println(errorMessage);
    }

    public ListWithViewer<String> raeumeFuerFach(String pFach){
        ListWithViewer<String> ergebnis = new ListWithViewer<String>();

        String sqlStatement =
            " SELECT r.nummer AS nummer, r.etape AS etape "+
            " FROM raum r JOIN unterricht u "+
            " ON r.id = u.raum_id "+
            " WHERE u.fach LIKE '%" + pFach + "%' ";

        // zur Kontrolle ausgeben
        System.out.println(sqlStatement);

        connector.executeStatement(sqlStatement);

        String errorMessage = connector.getErrorMessage();
        if(errorMessage != null)
        {
            System.err.println(errorMessage);
            return null;
        }

        QueryResult queryResult = connector.getCurrentQueryResult();

        String[][] data = queryResult.getData();

        for (int i = 0; i < data.length; i++) {
            String nummer = data[i][0];
            String etape = data[i][1];
            String zeile = nummer+": "+etape;
            System.out.println(zeile);
            ergebnis.append(zeile);
        }

        return ergebnis;
    }
}
```