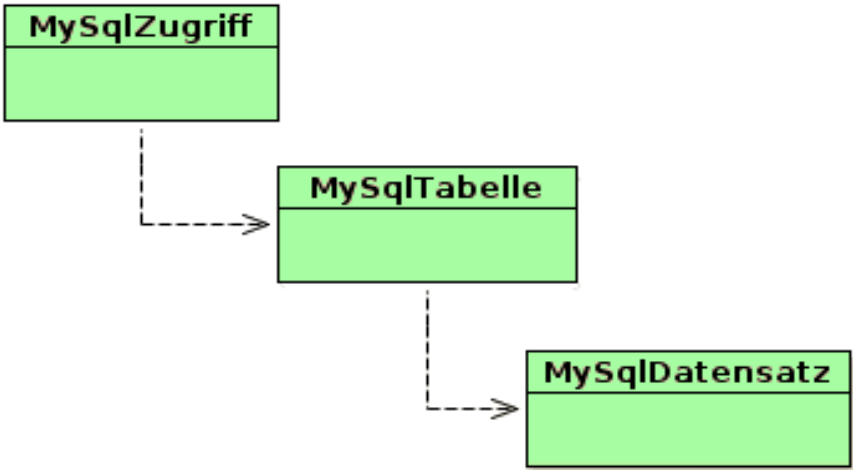
Datenbanken mit JAVA

Es ist nicht ganz trivial, mit JAVA an eine Datenbank anzukoppeln. Insbesondere braucht man für jedes Datenbank-System einen speziellen Treiber. Danach bleibt immer noch die Mühe, die Ergebnisse in einer GUI darzustellen.

Hier wird exemplarisch mit ***MySql*** demonstriert, wie eine Datenbank-Anbindung erfolgen kann.

Die ***Ergebnis-Tabellen*** kommen dabei als ***2D-Array von Strings*** zurück – unabhängig vom Datentyp innerhalb der Datenbank. Damit lässt sich das Ergebnis sehr leicht in eine JAVA-GUI einfügen.

Klassen-Modell



MySqlZugriff

Repräsentiert das DBMS. Regelt den Zugriff auf die Inhalte der Datenbank.

MySqlTabelle

Repräsentiert eine Tabelle in der Datenbank oder eine Ergebnis-Tabelle.

MySqlTabelle

Repräsentiert einen Datensatz in einer Tabelle.

Die Klasse MySqlZugriff

MySqlZugriff
private Connection connection private MySqlTabelle letzteErgebnisTabelle private String[] letzteMetaDaten private String letzteSqlFehlerMeldung private String letztesSqlStatement private boolean letztesSqlStatementFehlerfrei private PreparedStatement preparedStatement private ResultSet resultSet
public MySqlZugriff() public MySqlTabelle nenneLetzteErgebnisTabelle() public String nenneLetzteSqlFehlerMeldung() public String nenneLetztesSqlStatement() public boolean nenneLetztesSqlStatementFehlerfrei() public int sql_ohne_select(String sql) public boolean sql_select(String sql) public boolean verbindeMitServer(String ip, String datenbank, String benutzer, String passwort) public void verbinden_test() public void verbindungBeenden() private boolean datenbankTreiberLaden() private String[] metaDataAusgeben(ResultSet resultSet)

Verbindung aufbauen

`verbindeMitServer ( String IP , String DB , String user , String PW )`

Verbindet einen Benutzer mit einer Datenbank.
Gibt *`true`* zurück, wenn die Verbindung erfolgreich war, sonst *`false`*.

SQL-Anfragen

`sql_ohne_select ( String sql )`

Schickt ein SQL-Statement ab, das keine Ergebnis-Tabelle zurück gibt.
Gibt die Anzahl an betroffenen Datensätzen zurück.
Wird intern gespeichert. → *`nenneLetztesSqlStatement()`*
SQL-(Fehler-)Meldung → *`nenneLetzteSqlFehlerMeldung()`*
War sie erfolgreich? → *`nenneLetztesSqlStatementFehlerfrei()`*

`sql_select ( String sql )`

Schickt ein SQL-SELECT-Statement ab.
Gibt die Ergebnis-Tabell vom Typ *`MySqlTabelle`* zurück.
Wird intern gespeichert. → *`nenneLetztesSqlStaement()`*
SQL-(Fehler-)Meldung → *`nenneLetzteSqlFehlerMeldung()`*
War sie erfolgreich? → *`nenneLetztesSqlStatementFehlerfrei()`*

Wertvolle Hilfs-Methoden

`nenneLetztesSqlStatement ( )`

Gibt das letzte, abgesandte SQL-Statement zurück. *(z.B. für Verbesserung nach Fehler)*

`nenneLetzteSqlFehlerMeldung ( )`

Gibt die Rückmeldung des DBMS auf die aktuelle SQL-Anfrage zurück.

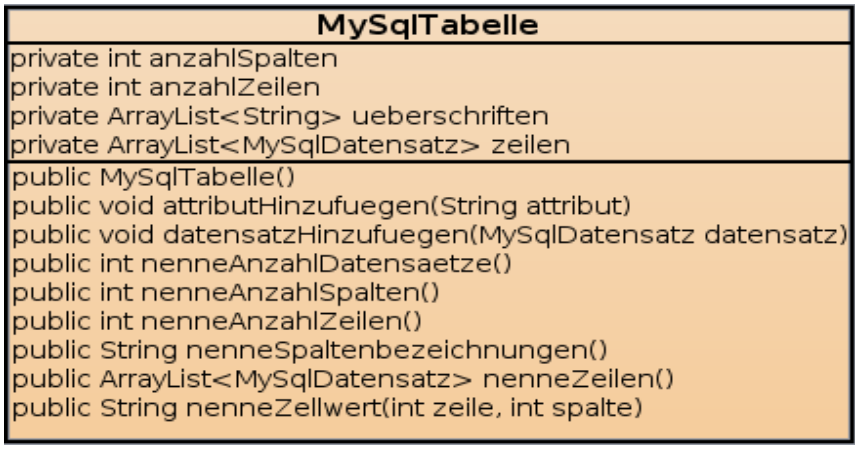
`nenneLetztesSqlStatementFehlerfrei ( )`

Gibt *`true`* zurück, wenn das letzte SQL-Statement fehlerfrei war, sonst *`false`*.

`verbindungBeenden ( )`

Trennt die Verbindung zum DB-Server.

Die Klasse MySQL-Tabelle



Meta-Informationen einer Tabelle

`nenneAnzahlDatensätze ( )`

Gibt die Anzahl an beinhalteten Datensätzen zurück.

`nenneAnzahlSpalten ( )`

Gibt die Anzahl an beinhalteten Spalten zurück.

`nenneAnzahlZeilen ( )`

Gibt die Anzahl an beinhalteten Zeilen zurück.

Daten aus Tabellen

`nenneSpaltenbezeichnungen ( )`

Gibt die Bezeichner aller Spalten als einen *(TAB-separierten)* `String` zurück.

`nenneZeilen ( )`

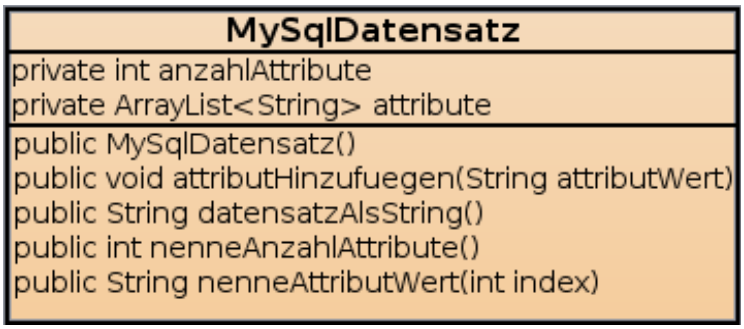
Gibt eine `java.util.ArrayList` vom Typ `MySqlDatensatz` zurück, die dann iteriert werden kann.

`nenneZellwert ( int zeile , int spalte )`

Gibt den Inhalt der entsprechenden Zelle der Tabelle als `String` zurück.

Die Methode `datensatzHinzufuegen(...)` wird eher Framework-intern verwendet.

Die Klasse MySqlDatensatz



Meta-Informationen eines Datensätzes

```
nenneAnzahl ( int zeile , int spalte )
```

Gibt die Anzahl der Attribute des Datensatzes zurück.

Daten aus Datensätzen

```
nenneAttributwert ( int index )
```

Gibt den entsprechenden Attributwert als `String` zurück.

```
datensatzAlsString ( )
```

Gibt den gesamten Datensatz als TAB-separierten String zurück.

Die Methode `attributHinzufuegen(...)` wird eher Framework-intern verwendet.