

JDBC Grundübung

Es ist eine JDBC-Anwendung zu entwickeln, mit der Schülerdaten gespeichert und abgefragt werden können. Ausgangsbasis ist eine CSV-Datei mit folgendem Aufbau:

Familienname;Vorname;Geschlecht;Katalognummer;Klasse

Die Datei steht in zwei Versionen zur Verfügung:

1. `schueler_err.csv` enthält Fehler
2. `schueler.csv` ist fehlerfrei

Als Datenbank ist eine PostgreSQL - DB zu verwenden. Definieren Sie zu diesem Zweck eine geeignete Tabelle `schueler`, wobei der zusammengesetzte Primärschlüssel aus Katalognummer und Klasse bestehen soll. Die Tabelle ist durch das Javaprogramm zu erzeugen. Sollte die Tabelle bereits existieren, so sind zunächst alle Datensätze aus der Tabelle zu löschen.

Zur Lösung der Aufgabe ist wie folgt vorzugehen:

- Entwickle eine Domäinklasse `Schueler` und eine Serviceklasse `SchuelerTools` mit folgender Methode:

```
public List<Schueler> readFromCSV(String filename) throws IOException
Liest alle syntaktisch korrekten Zeilen aus der CSV-Datei filename und liefert eine Liste aller erfolgreichen gelesenen Schüler.
```

- Entwickle eine Klasse `JdbcSchuelerRepository`, die das folgende Interface implementiert:

```
public interface SchuelerRepository {
    // persistiert in einer Transaktion alle in lst gespeicherten Schueler.
    // Löscht die Tabelle schueler falls sie existiert und erzeugt sie neu
    int persistSchueler(List<Schueler> lst) throws SQLException;

    // löscht alle in der DB gespeicherten Schueler und liefert die
    // Anzahl der gelöschten Schüler zurück
    int deleteAll() throws SQLException;

    // Liefert eine Liste aller Schüler der Klasse klasse aufsteigend nach
    // Katalognummern sortiert bzw. eine leere Liste wenn die Klasse
    // unbekannt ist
    List<Schueler> findSchuelerByKlasse(String klasse) throws SQLException;

    // Liefert eine Liste aller Schüler mit Geschlecht geschlecht aufsteigend
    // nach Klasse und Katalognummer sortiert. Wirft eine IllegalArgumentException
    // Exception, wenn das Geschlecht nicht 'm' oder 'w' ist.
    List<Schueler> findSchuelerByGeschlecht(char geschlecht) throws SQLException;

    // Liefert eine Map, deren Schlüssel die aufsteigenden Klassenzahlen und
    // deren Werte die entsprechenden Schülerzahlen sind.
    Map<String,Integer> getKlassen() throws SQLException;
}
```

Aufgabe 1: Testgetriebene Implementierung

Entwickle nach oben angegebenem Muster gegen die gegebenen UnitTests `SchuelerToolsTest` und `SchuelerRepositoryTest`.

Ergänze selbst Testmethoden für die Methoden `findSchuelerByGeschlecht()` und `getKlassen()`.

Aufgabe 2: Testapplikation

Entwickle eine FX-Applikation, in der ein Benutzer die Ergebnisse der 3 Abfragemethoden `findSchuelerByKlasse()`, `findSchuelerByGeschlecht()` und `getKlassen()` visualisieren kann.