

ÜBUNGSZETTEL „DATENBANKZUGRIFF MIT JAVA, JDBC UND DAO“

AUFGABE 0: VOLLSTÄNDIGE DOKUMENTATION (!)

Dokumentiere alle deine Arbeiten zu allen folgenden Teilaufgaben anhand eines eigenen Git-Repositories für diesen Übungszettel.

- Regelmäßige Commits (jedes Mal, wenn ein Aspekt gelöst wurde)
- Führe im Repo eine Markdown-Datei mit allen wesentlichen Schritten der Codierung, mit eigenen Erklärungen zu den Konzepten, mit Screenshots, mit Codeauszügen.

AUFGABE 1: JDBC INTRO TEIL 1

Arbeite das JDBC-Intro-Videotraining aus dem Kurs vollständig durch.

- Richte die Entwicklungsumgebung ein
- Programmiere alles selbstständig nach und mit (!)
- Spiele ein wenig mit dem Code, probiere aus!
- Du musst alle Codeteile verstehen und syntaktisch sowie semantisch erklären können.
- Du musst neue Konzepte verstehen und erklären können und diese selbst dokumentieren (Markdown), dazu zählen:
 - MySQL-Datenbankserver einrichten
 - DB-Server Adminkonsole einrichten und verwenden können (insb. auch SQL-Statements absetzen können)
 - Java-Maven-Projekt erstellen
 - Dependency für die Verbindung einer MySQL-Datenbank in der pom.xml hinzufügen
 - Verbindung zur Datenbank aufbauen
 - Prepared-Statement für die Abfrage von Daten aus der DB verwenden
 - Prepared-Statement für die Änderung von Daten in der DB verwenden
 - Abgefragte Daten aus der DB mit ResultSet und Schleifen verarbeiten
 - Exceptions verwenden und verarbeiten, die auftreten können
 - Debugging von JDBC-Applikationen (SQL-Statements prüfen, Exceptions richtig interpretieren, Debugger verwenden)

AUFGABE 2: JDBC INTRO TEIL 2

Erstelle neben der Tabelle im Videotraining eine weitere Tabelle mit Primärschlüssel und mindestens 2 weiteren Spalten (eine Textspalte, eine Zahlenspalte) für eine weitere Entität nach Wahl (z.B. Kurs, Adresse, Hobbies, etc.).

Implementiere zu dieser neuen Tabelle den Datenbankzugriff aus Java heraus, analog zu Aufgabe 1.

AUFGABE 3: JDBC UND DAO – KURSE

Lerne anhand des zweiten Videotrainings, wie man mit JAVA auf Daten aus Datenbanken mit dem DAO-Entwurfsmuster zugreift.

- Programmiere alles selbstständig nach und mit!
- Spiele ein wenig mit dem Code, probiere aus!
- Du musst alle Codeteile verstehen und syntaktisch sowie semantisch erklären können
- Du musst neue Konzepte verstehen und erklären können und diese selbst dokumentieren (Markdown). Zu diesen Konzepten zählen:
 - DAO-Pattern zum objektrelationalen Zugriff auf Datenbanken verstehen und anwenden
 - Grundkonzept des objektrelationalen Mappings verstehen
 - Singleton-Pattern zum Aufbau der DB-Verbindung verstehen und anwenden
 - Commandline-Interface (Kommandozeilenmenü) sauber programmieren
 - Exceptions verstehen und verwenden
 - Abstrakte Klassen verstehen und verwenden
 - Interfaces (auch mit Erben für Interfaces) verstehen und verwenden
 - Domänenklassen korrekt aufbauen (Objekte immer im konsistenten Zustand halten, Exceptions verwenden, Setter absichern)
 - CRUD-Operationen mit DAO-Pattern und JDBC umsetzen

AUFGABE 4: JDBC UND DAO – STUDENTEN

Erweitere die fertig nachprogrammierte Applikation mit einem DAO für CRUD für eine neue Domänenklasse „Student“:

- Studenten haben eine Student-ID, einen VN, einen NN, ein Geburtsdatum
- Domänenklasse implementieren (Setter absichern, neue Exceptions definieren, Business-Regeln selbst wählen - z.B. dass Name nicht leer sein darf)
- Eigenes Interface MyStudentRepository von BaseRepository ableiten. MyStudentRepository muss mindestens 3 studentenspezifische Methoden enthalten (z.B. Studentensuche nach Namen, Suche nach ID, Suche nach bestimmtem Geburtsjahr, Suche mit Geburtsdatum zwischen x und y etc.).
- Implementierung von MyStudentRepository durch eine neue Klasse MySQLStudentRepository analog zum MySQLCourseRepository.
- Erweiterung des CLI für die Verarbeitung von Studenten und für spezifische Studenten-Funktionen (z.B. Student nach dem Namen suchen)

AUFGABE 5: JDBC UND DAO – BUCHUNGEN

Gib einen textuellen Vorschlag ab, wie man die bisher programmierte Applikation für die Buchung von Kursen durch Studenten erweitern könnte.

Beschreibe, wie eine neue Buchungs-Domänenklasse ausschauen sollte, wie man ein DAO für Buchungen dazu entwickeln sollte, wie man die CLI anpassen müsste und welche Anwendungsfälle der Benutzer brauchen könnte (wie etwa „Buchung erstellen“).

Verwende zur Illustration insb. auch UML-Diagramme.