Aufgaben "Avatar"

Mittels JDBC sollen die Daten aus der Datenbank in entsprechende Daten-Objekte abgefüllt werden.

# Gegeben

## ERD



## Java-Klassen

### Datenklassen

Die Klasse Person ist *'abstract'*.



### Zugriffsklasse und Testing



# Eingesetzte Technologien

* Maven 3.x
* Java JDK 1.8 oder höher
* SubVersion ([SourceForge](https://sourceforge.net))
* [JDBC](https://de.wikipedia.org/wiki/Java_Database_Connectivity) Type-4
* JUnit 4.1x
* Code-Konvention
  + Sun/Oracle Code Conventions
  + <http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>
  + Ausnahme Zeilenlänge max. 150 Zeichen
* RDBMS
  + MySQL oder Apache [Derby](https://db.apache.org/derby/)
* File Encoding: UTF-8
* **Kein** Einsatz von 3rd Party Libraries

## Versionskontrolle

Für die Versionierung wird SubVersion eingesetzt (mit empfohlener Struktur für "Single-Project"). Als Provider wird SourceForge verwendet. Der Verlauf der Arbeiten (commits) sollen ersichtlich sein. Stellen Sie sicher, dass niemand anders als Sie auf das SVN-Repository Zugriff haben.

Nach Abgabe der Arbeit soll das Projekt für jeden einsehbar sein, so, dass auch der Lehrer und die Mitstudenten auf das Projekt zugreifen kann. Der Link soll auf die Startseite des Projekts verweisen.

## Build- und Konfigurationsmanagement

Als Konfigurationsmanagement wird Maven eingesetzt. Alle Maven Files sollen in der Versionsverwaltung abgelegt werden (jedoch keine Settings der verwendeten Entwicklungsumgebung) Als Dokumentation (Minimum) sollen im pom.xml die entsprechenden Entwickler aufgeführt werden. Ebenso soll die URL des Projekts in SourceForge im POM eingetragen werden.

Überprüfen Sie den Link, damit dieser nach Abgabe auch ohne Anmeldung funktioniert.

# Aufgabe

* Erstellen Sie eine Datenbank gemäss ERD
* Erstellen Sie eine DDL (SQL-Script) für die Erstellung der entsprechenden Tabellen und legen Sie diesen ins Verzeichnis '/src/main/resources/sql' ab.
* Erstellen Sie eine DML (SQL-Script) mit dem Sie Testdaten in die Datenbank abspeichern können. Legen Sie diesen im selben Verzeichnis wie der DDL Script ab.
* Erstellen Sie die Entsprechenden Java-Klassen: Datenklassen und Repository.
* Erstellen Sie einen JUnit-Test mit welchem Sie die korrekte Funktionsweise Ihrer Applikation nachweisen.
* Mit Maven lassen sich die SQL-Scripts anwenden
  + Unterschiedliche **Profile** für
    - Datenbank Schema erstellen
    - Tabellen anlegen
    - Tabellen löschen
    - Tabellen mit Testdaten füllen

# Abgabe

Abgegeben werden soll ein ZIP-File mit dem entsprechenden Maven-Projekt. Dieses ZIP muss alle notwendigen Dateien enthalten. Im POM soll u.a. die URL zum Projekt bei SourceForge hinterlegt sein.

# Links

* [Datenbankmanagement mit JDBC](http://openbook.rheinwerk-verlag.de/javainsel9/javainsel_24_001.htm)
* Tutorialspoint: J[DBC Tutorial](http://www.tutorialspoint.com/jdbc/index.htm)
* [MySQL and Java JDBC - Tutorial](http://www.vogella.com/tutorials/MySQLJava/article.html)
* [Oracle Tutorial JDBC Database Access](http://docs.oracle.com/javase/tutorial/jdbc/)
* [MySQL JDBC Type 4 driver](http://mvnrepository.com/artifact/mysql/mysql-connector-java) für Maven