

# Aufgabenblatt 1 V0.92

Im ersten Praktikumstermin sollen Sie sich schrittweise in die Arbeit mit Java und relationalen Datenbanken einarbeiten. Im AIL finden Sie folgenden Softwarestand vor, den ich auch bei mir installiert habe:

- Eclipse Java EE IDE for Web Developers. Version: Luna Service Release 1a (4.4.1)  
Build id: 20150109-0600
- Java 8\_0\_31 (hier bin ich auf 8\_0\_41, aber das ist nicht entscheidend)
- JUnit 4
- Hibernate 4.3.9 Final
- slf4j 1.7.12
- activemq 5.11.1
- jms 4.4.2

Für jeden Schritt finden Sie die Projekte gezippt im pub unter in/praktikum/projekte. Das Projekt *Libs* brauchen Sie für alle anderen Projekte, deshalb ist es in jedem enthalten. Auf Grund der Größe werde ich in Zukunft darauf verzichten. Dort finden Sie in den Verzeichnissen *lib/hibernate4*, *lib/oracle* und *lib/slf4j* auch alle benötigten .jar-Dateien. Es fehlen noch einige Javadoc und Source-Archive. Wegen der Ausgaben auf *System.err* empfehle ich, diese Ausgabe umzuleiten.

1. JDBC: Hier geht es um den Zugriff mit dem geringsten „Rüstaufwand“. Es geht auch nur um eine Klasse *Customer* und die zugehörige Tabelle. *Task Tags* weisen Sie auf die zu bearbeitenden Stellen hin. Achten Sie bitte auch auf die weiteren Hinweise im Code.
  - 1.1. Denken Sie bitte daran, vor den Testfällen das sql-Skript auszuführen!
  - 1.2. Vervollständigen Sie bitte die Klasse *CustomerDAOImpl*!

- 1.3. Passen Sie bitte die notwendigen Parameter in der Klasse *JDBCTest* an! Die Auslagerung der Strings in eine *messages.properties* Datei ist nur ein erster Schritt. Benutzerkennungen und Passworte haben in Code nichts verloren und sind nur so lange bedingt tolerabel wie Sie alleine testen. Für Eingaben, die sichtbar sein dürfen, geht das mit einfachen Mitteln besser (siehe z.B. *.java.swing.JOptionsPane*). Weitere Möglichkeiten bietet z. B. *controlsfx*. Die Klasse *Messages* und die Datei *messages.properties* finden Sie im Projekt *Libs* zwecks Verwendung in den anderen Projekten.
- 1.4. Nach Abschluss Ihrer Arbeiten müssen die Testfälle in *JDBCTest* bestanden werden.
2. JPA\_XML\_Hibernate: Auch hier geht es nur um eine Klasse *Customer* und die zugehörige Tabelle. Die Konfiguration erfolgt über eine Datei *persistence.xml* oder eine Datei *orm.xml*.
  - 2.1. Denken Sie bitte daran, vor den Testfällen das sql-Skript auszuführen!
  - 2.2. Passen Sie bitte die *persistence.xml* Datei und/oder eine *orm.xml*-Datei für die Anbindung von *Customer* an!
  - 2.3. Nach Abschluss Ihrer Arbeiten müssen die Testfälle in *JPAXMLTest* bestanden werden.
3. JPA\_Annotations\_Hibernate: Auch hier geht es nur um eine Klasse *Customer* und die zugehörige Tabelle. Nun erfolgt die Konfiguration über Annotationen.
  - 3.1. Denken Sie bitte daran, vor den Testfällen das sql-Skript auszuführen!
  - 3.2. Konfigurieren Sie bitte den OR-Mapper in der *persistence.xml*!
  - 3.3. Ergänzen Sie bitte die Klasse *Customer.java* mit Hibernate-Annotationen!
  - 3.4. Ergänzen Sie die Klasse *JPAAnnotationsTest.java*
  - 3.5. Nach Abschluss Ihrer Arbeiten müssen die Testfälle in *JPAXMLTest* bestanden werden.
4. JPA\_Relationships\_Hibernate: Hier soll die Konfiguration von Tabellen und Verbindungstabellen über Annotationen erfolgen.
  - 4.1. Zeichnen Sie bitte ein UML-Klassendiagramm, das die Beziehungen zwischen den Klassen *Customer*, *Bank*, *Bank* und *Address* darstellt!
  - 4.2. Bilden Sie die Assoziationen durch Hibernate-Annotationen in den Java-Klassen ab! Achten Sie darauf, ob die Assoziationen in den Klassen wirklich ausreichend beschrieben sind!
  - 4.3. Konfigurieren Sie bitte den OR-Mapper in der *persistence.xml*
  - 4.4. Denken Sie bitte daran, vor den Testfällen das sql-Skript auszuführen!
  - 4.5. Ergänzen Sie bitte aussagefähige Testfälle in *JPARelationshipsTest*!

- 4.6. Nach Abschluss Ihrer Arbeiten müssen die Testfälle in *JPAAnnotationsTest* in Ihrer und meiner Version bestanden werden.
5. JPA\_Locking\_Hibernate: Hier sollen Sie das Arbeiten mit Locking üben. Es geht nur um eine Klasse/Tabelle *Customer*.
  - 5.1. Konfigurieren Sie bitte den OR-Mapper in der *persistence.xml*! (Das sollten Sie aus anderen Aufgaben einfach übernehmen können.)
  - 5.2. Ergänzen Sie bitte eine Spalte für optimistische Sperren in der Klasse *Customer* und in *createdb.sql*!
  - 5.3. Annotieren Sie bitte die ergänzte Klasse *Customer* um die notwendigen Hibernate Annotationen!
  - 5.4. Schreiben Sie bitte in der Klasse *JPALockingTest* aussagefähige Testfälle!
  - 5.5. Denken Sie bitte daran, vor den Testfällen das *sql*-Skript auszuführen!
  - 5.6. Nach Abschluss Ihrer Arbeiten müssen die Testfälle in *JPALockingTest* in Ihrer und meiner Version bestanden werden.