

Umsetzung JPA, Hibernate Eigenes Projekt



Einleitung:

Sie erstellen Ihr **eigenes Projekt** mit einer **selbst entworfenen Datenbank**. (Keine Gruppenarbeiten, kein Copy Paste aus dem Internet)

Sie werden verschiedene Aufgaben mit dieser Datenbank implementieren.

Die Aufgabe wird bewertet.

Der genaue Umfang wird in der Folge laufen definiert.

Ziele:

- ⇒ Sie können die Funktion des **Persistenz-Frameworks Hibernate** erläutern
- ⇒ Sie können die Funktion des **Interfaces JPA** erläutern.
- ⇒ Sie können die Funktion von **Transaktionen** erläutern.
- ⇒ Sie können Entitäten aus einer Datenbank über das Persistenz-Framework und das Interface JPA mit Objekten der Applikation verwalten.
- ⇒ Sie setzen **CRUD** für die Verwaltung der Daten um.
- ⇒ Sie setzen verschiedene Beziehungen zwischen Entitäten in der Applikation um.

Inhaltsverzeichnis

Einleitung:	1
Ziele:	1
Inhaltsverzeichnis	1
Weitere Informationsquellen	2
Ideen zur Aufgabe	2
Bewertungsraster	3
Aufgabe Entwurf	3
Aufgabe Erste Tabelle	3
Aufgabe Erste Beziehung	3
Aufgabe Erweiterung um weitere Beziehungen	3

Weitere Informationsquellen

Die Seiten von Java Code Geeks zeigt eine Vielzahl von Tutorials zu einer Vielzahl von weiteren Themen rund um Hibernate. Die kann Ihnen bei Ihrem Projekt helfen!

<https://examples.javacodegeeks.com>

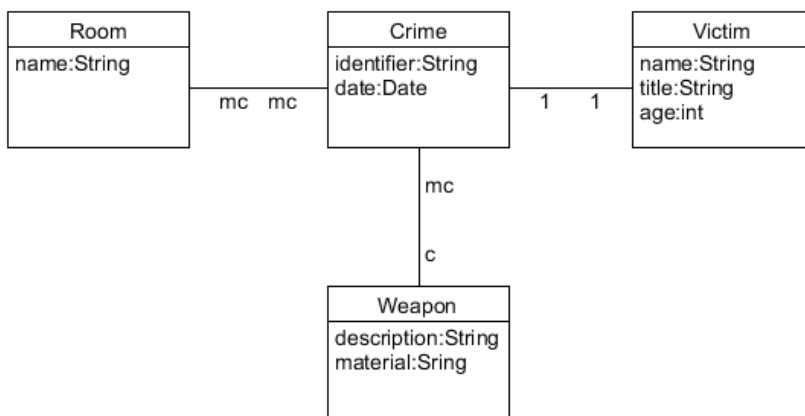
Ideen zur Aufgabe

Eine Datenbank besteht selten aus nur einer Tabelle.

Und zwischen den Tabellen stehen häufig verschiedensten Beziehungen.

Eine erste Idee könnte so aussehen.

Sie müssen für Ihre DB ein anderes Thema wählen.



Es ist absolut sinnvoll mal **mit einer Beziehung zu beginnen** und diese umzusetzen.

Und dann die **Herausforderungen mit den richtigen Annotierungen und das Umsetzen des CRUD** zu meistern.

Was man mit CRUD ausprobieren sollte/kann

- Alle Datensätze auflisten.
- Einen einzelnen Datensatz auflisten.
- Einen weiteren Datensatz hinzufügen, unter Einhaltung der Beziehungen.
- Einen bestehenden Datensatz ändern.
- Einen Datensatz löschen. Dabei die bestehenden Beziehungen richtig behandeln.

Eine Herausforderung dabei ist es, die Datenbank immer wieder in den Ausgangszustand zu führen. Datensätze, die man hinzufügt, auch wieder zu löschen. Und doch den ganzen Vorgang nachvollziehbar zu gestalten.

Was passiert, wenn man einen Datensatz mit den in Beziehung stehenden Datensätzen? Fremdschlüsselwerte dürfen nicht plötzlich auf nicht mehr vorhandene Primärschlüsselwerte verweisen.

Das zu entscheiden ist nicht immer einfach. Und abhängig davon, wie man das Datenmodell interpretiert/versteht.

Das Problem mit dem Cache

Falls Sie mal nicht sehen, was sein sollte, DB und Ausgabe nicht übereinstimmen, so kann das am Cache liegen.

Lesen Sie dazu das Dokument zum Problem mit dem Cache.

Bewertungsraster

Das Bewertungsraster ist in einem separaten Dokument.

Ihre Bewertung ist abhängig davon, was Sie alles Umsetzen,

Aufgabe Entwurf

- Starten Sie mit einem Entwurf von einem Datenmodell.
- Nehmen Sie mindestens 3 Tabelle.
Bei mehr als 6 Tabellen wird es jedoch aufwendig.

Aufgabe Erste Tabelle

- Implementieren Sie in Ihrem Projekt eine CRUD Transaktionen für eine erste Tabelle
- Schaffen Sie die Möglichkeit, das CRUD zu zeigen, ohne ständig den SourceCode ändern zu müssen.

Aufgabe Erste Beziehung

- Implementieren Sie in Ihrem Projekt eine erste Beziehung.
- Implementieren und probieren Sie das CRUD für die beiden beteiligten Entitäten aus.
- Schaffen Sie die Möglichkeit, das CRUD zu zeigen, ohne ständig den SourceCode ändern zu müssen.

Aufgabe Erweiterung um weitere Beziehungen

Mögliche Erweiterungen, die Ihre Note beeinflussen.

- Eine zweite jedoch andere Beziehungsarten implementiert.
- Bidirektionale Beziehung implementiert,
- Eine dritte nochmals andere Beziehungsart implementiert
- Löschen von Entitäten unter Einhaltung der Beziehung für verschiedene Fälle umgesetzt.
Änderungsweitergabe versus Löschweitergabe

Hinweis zu den weiteren Beziehungen

Ergänzen Sie nach Bedarf weitere Entitäten.

Sie wählen die Beziehungen selber aus. zBsp:

- One to Many
- Many to One
- One to One
- Many to Many
- One to Many + Many to One als bidirectional Beziehung
Das heisst in beiden Entitäten sind Annotierungen.
Allerdings muss eine Entität der "Chef" sein.
Das ist gar nicht so einfach zu verstehen.
Lesen Sie dazu diesen Artikel
<https://www.baeldung.com/hibernate-one-to-many>