Übung GOT - JDBC - Repository

Table of Contents

Version: 1.0

Aufgabenstellung

Lesen Sie diese Angabe genau durch

Für diese Aufgabe brauchen Sie keine main()-Methode erstellen, da die Funktionalität über (bereits vorgegebene) Unit-Tests geprüft wird. Es wird empfohlen, das Programm in kleinen Schritten zu erstellen, dh ein Test nach dem anderen soll funktionieren, bei jedem Test werden die von Ihnen erstellten Methoden - falls notwendig - überarbeitet. **Die Unit-Tests werden nicht verändert**

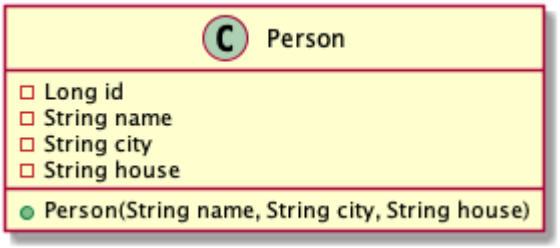
Achten Sie darauf, dass Ihre Methoden nicht zu lang werden und Sie eventuell eigene Methoden schreiben (zB für INSERT und UPDATE)

Entity Person

Erstellen sie im package `Entity` eine Klasse `Person` mit folgenden Attributen. Ergänzen Sie die notwendigen Methoden.

Die Instanzvariable `ID` wird für die Persistierung in der Datenbank benötigt, da die `ID` der Primärschlüssel der Tabelle ist

Achten Sie darauf, dass die Datenbank automatisch sicherstellt, dass ein Name einer Person nicht doppelt vorkommen darf, indem Sie eine Unique-Constraint für die Spalten `NAME`,`CITY` und `HOUSE` verwenden.

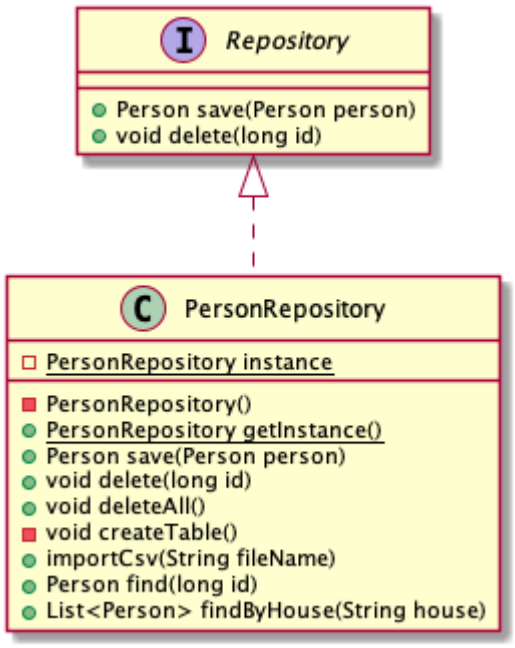


equals

Zwei Personen sind dann gleich, wenn Sie in name, city und house übereinstimmen.

PersonRepository

Das `PersonRepository` implementiert das Interface `Repository`. Beachten Sie, daß das PersonRepository ein Singleton ist. Außerdem wird beim Erstellen eines `PersonRepository`- Objekts immer eine Tabelle erstellt. Existiert bereits eine Tabelle, so wird eine Exception geworfen. Diese Exception wird gefangen und "nur" die Exception-Message auf `System.err` ausgegeben.



Singleton

Für das Singleton verwenden Sie eine statische Variable instance sowie die statische Methode getInstance()

Methode save(Person person)

- Beim Speichern einer Person wird (von der Datenbank mittels eines UNIQUE Constraints) überprüft, ob nicht bereits eine Person doppelt gespeichert wird (name, city und house sind ident). Tritt ein Fehler auf wird eine Exception geworfen und innerhalb der Methode abgefangen. Auf der Console wird die entsprechende Fehlermeldung (`e.getMessage()`) ausgegeben: zB `The statement was aborted because it would have caused a duplicate key value in a unique or primary key constraint or unique index identified by 'PERSON_UQ' defined on 'PERSON'.`
  - Beispiel:
    - bestehende Person: ID:23, name:"Loras", city:"White Haven", house:"Tully"
    - neue Person: ID:null, name:"Loras", city:"White Haven", house:"Tully"
      → die neue Person wird **nicht** gespeichert
- Wenn eine Person bereits eine id hat, wird nicht das Speichern verhindert, sondern die Werte der Instanzvariablen der Person werden geändert
  - Beispiel:
    - bestehende Person: ID:23, name:"Loras", city:"White Haven", house:"Tully"

- neue Person: ID:23, name="Arya", city:"Winterfell", house:"Stark"
  → die neue Person wird gespeichert

Wird eine neue id von der Datenbank generiert, kann man diese auslesen. Siehe dazu: https://stackoverflow.com/a/1915197

delete(long id)

Die Person mit der angegebenen Id wird in der Tabelle PERSON gelöscht.

deleteAll()

Löschen sämtlicher Zeilen in der Tabelle PERSON

createTable()

Erstellen der Tabelle Person. Falls ein Fehler auftritt, wird nur die entsprechende Message des Fehlers auf System.err ausgegeben (e.getMessage())

find(long id)

Eine Person mit angegebenen Id wird zurückgegeben

findByHouse(String house)

Eine Liste von Personen des gesuchten Hauses wird zurückgegeben