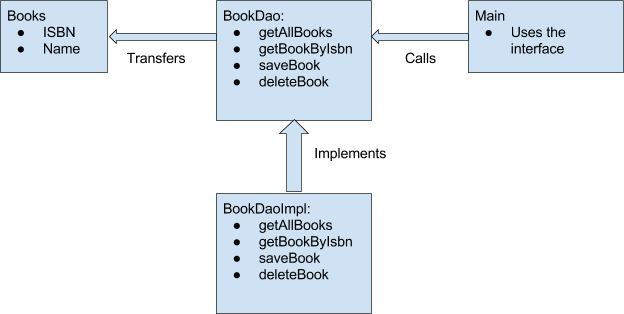
# DAO - Tests unitaires avec Junit & Hamcrest – Accès à la BDD SQL Server

**Objectifs** :

L’objectif est de compléter l’étude de cas Supmaster en étudiant le pattern DAO, déjà abordé lors des projets Android

## Pattern DAO

DAO signifie Data Access Object. Ce pattern est utilisé pour séparer la couche de persistance de données (dans la BDD, dans un fichier externe, dans un objet sérialisé …) de la logique métier codée dans l’application.



Exemple:

Dans le schema ci dessus, on a:

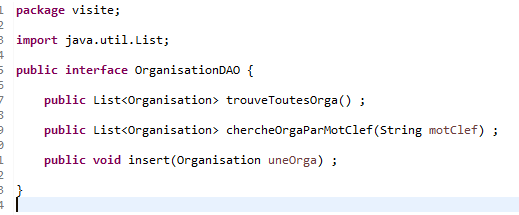
* Un objet “métier”, Books avec une classe Java associée
* Une interface DAO BookDao qui définit les méthodes d’accès à la base de données pour récupérer des objets Book, les stocker, ls supprimer …
* Une classe d’implémentation BookDaoImpl qui se charge concrètement:
  + De définir une connexion à la BDD
  + De définir et utiliser un driver, par exemple JDBC dans le contexte Java
  + D’implémenter les méthodes de l’interface DAO pour accéder aux données, les restituer, les enregistrer , les modifier et les supprimer

## Implémentation pour Supmaster

Nous allons implémenter le pattern DAO pour la classe Organisation.

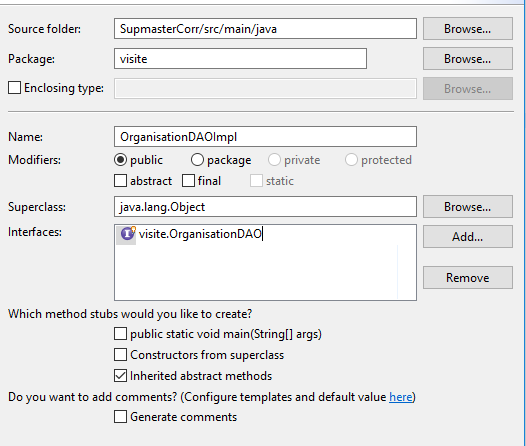
Etape 1 : interface DAO

Ajouter une interface ***OrganisationDAO*** comme ci-dessous :



Etape 2 : implémentation de l’interface DAO

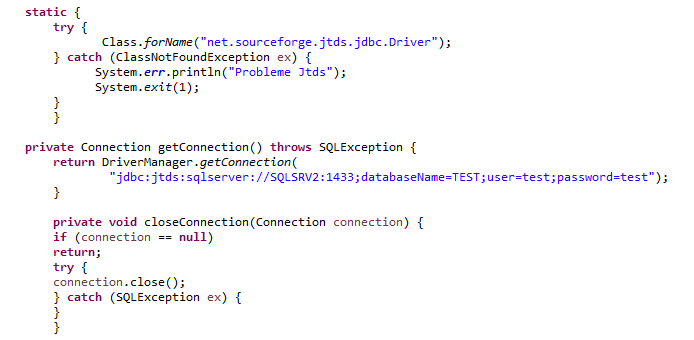
1. Créez une nouvelle classe nommée OrganisationDAOImpl , comme ci-dessous :



Puis choisissez d’implémenter toutes les méthodes.

1. Ensuite, on écrit le code de l’implémentation

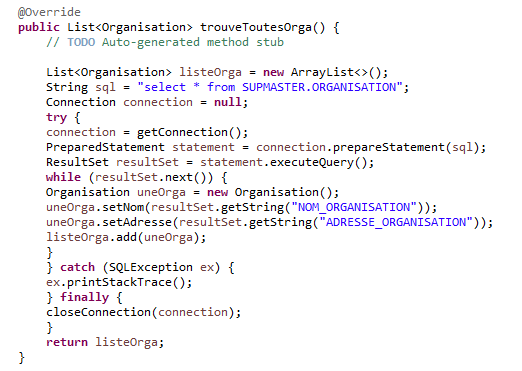
Le driver et la connexion :



Question 1 : effectuez une recherche

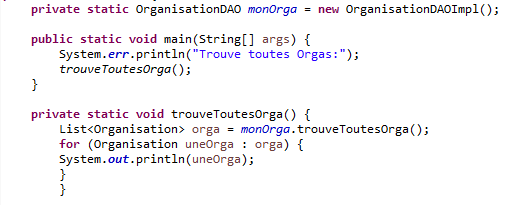
Quel driver utilise t’on ? Comment se fait il qu’il soit disponible dans notre projet ? Que devriez vous faire comme modification pour utiliser votre propre base de données ?

1. On implémente la méthode trouveToutesOrga()



Etape 3 : on développe une classe d’utilisation de notre DAO

Créez une classe UtilisationDAO avec le code suivant :



Question 2 : vérifiez que ce code marche. Que fait il ?

Exercice 1 : Ecrire un test unitaire de la méthode trouveToutesOrga, en utilisant Hamcrest et les matchers de Collection

Exercice 2 : Ecrire le code de la méthode chercheOrgaParMotClef dans la classe OrganisationDAOImpl

Exercice 3 : Ecrire le code de la méthode insert dans la classe OrganisationDAOImpl

Exercice 4 : Ecrire un test unitaire de la méthode insert dans la classe OrganisationDAOImpl

## POO, classes métier et collection HashMap

On revient maintenant sur l’étude de cas DEMGUIV / Bretagne afin d’illustrer un nouveau type de collection, l’interface Map, avec une classe implémentant cette interface : HashMap.

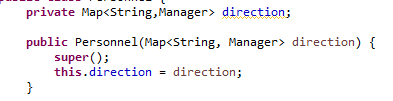
Exercice 5 : effectuez un diagramme de classe UML métier des classes Manager et Salarie

Que peut on dire de l’association entre Manager et Salarie ?

Etape 4 : on va créer une nouvelle classe nommée **Personnel**

Cette classe a comme unique attribut direction, qui est une collection de services, avec pour chaque service un Manager.

* Quel est l’intérêt de la collection Map pour représenter cette structure ?
* Ecrire le code correspondant



Exercice 5 : écrire un autre constructeur qui initialise la collection ***direction***

Exercice 6 : écrire une méthode ***ajoutePersonnel***

En paramètre elle prend le nom d’une direction et un Manager, et elle ajoute à la collection direction.

Créez une classe de test unitaire et un test hamcrest pour tester cette méthode : on repartira par exemple du setup() de **TestManager**, on testera dans un 1er temps que la collection (laquelle ?) est vide, puis on ajoutera une direction avec comme Manager theBigPatron et on testera alors que cette collection a une taille de 1.

Exercice 7 : méthode listePersonnel

Ecrire une méthode listePersonnel qui se contente d’écrire pour chaque direction, le nom du Manager et la liste de ses salariés séparés par des espaces, entre []

Créez une classe avec une méthode main pour afficher la liste du personnel.

## Annexe 1 : collection Map

|  |  |
| --- | --- |
| **Methods** | |
| **Modifier and Type** | **Method and Description** |
| void | [**clear**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#clear())()  Removes all of the mappings from this map (optional operation). |
| boolean | [**containsKey**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#containsKey(java.lang.Object))([**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) key)  Returns true if this map contains a mapping for the specified key. |
| boolean | [**containsValue**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#containsValue(java.lang.Object))([**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) value)  Returns true if this map maps one or more keys to the specified value. |
| [**Set**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html)<**[Map.Entry](https://docs.oracle.com/javase/7/docs/api/java/util/Map.Entry.html" \o "interface in java.util)**<[**K**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html),[**V**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html)>> | [**entrySet**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#entrySet())()  Returns a [**Set**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html) view of the mappings contained in this map. |
| boolean | [**equals**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#equals(java.lang.Object))([**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) o)  Compares the specified object with this map for equality. |
| [**V**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html) | [**get**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#get(java.lang.Object))([**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) key)  Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key. |
| int | [**hashCode**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#hashCode())()  Returns the hash code value for this map. |
| boolean | [**isEmpty**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#isEmpty())()  Returns true if this map contains no key-value mappings. |
| [**Set**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html)<[**K**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html)> | [**keySet**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#keySet())()  Returns a [**Set**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html) view of the keys contained in this map. |
| [**V**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html) | [**put**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#put(K,%20V))([**K**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html) key, [**V**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html) value)  Associates the specified value with the specified key in this map (optional operation). |
| void | [**putAll**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#putAll(java.util.Map))(**[Map](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html" \o "interface in java.util)**<? extends [**K**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html),? extends [**V**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html)> m)  Copies all of the mappings from the specified map to this map (optional operation). |
| [**V**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html) | [**remove**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#remove(java.lang.Object))([**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) key)  Removes the mapping for a key from this map if it is present (optional operation). |
| int | [**size**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#size())()  Returns the number of key-value mappings in this map. |
| [**Collection**](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)<[**V**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html)> | [**values**](https://docs.oracle.com/javase/7/docs/api/java/util/Map.html#values())()  Returns a [**Collection**](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html) view of the values contained in this map. |