

# Programmation Objet Avancée - TP7

—o000o—o000o—

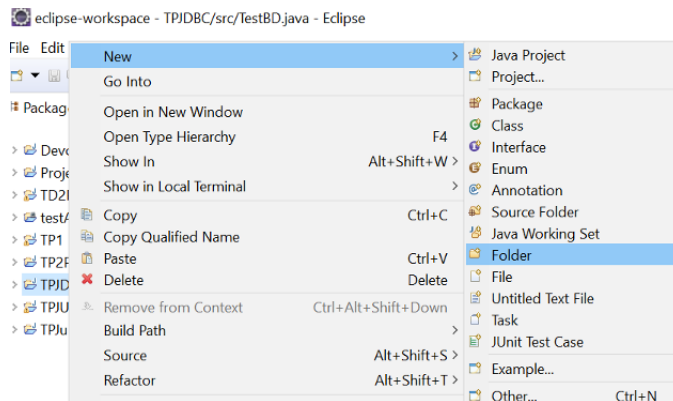
## Accès aux bases de données par JDBC

Le but de ce TP est de manipuler une base de données MySQL à travers un programme java en utilisant l'interface de programmation JDBC. Les premières questions vous aident à configurer votre environnement de travail. Les dernières questions demandent la programmation d'un accès à une base pour la recherche de points d'intérêts (POI).

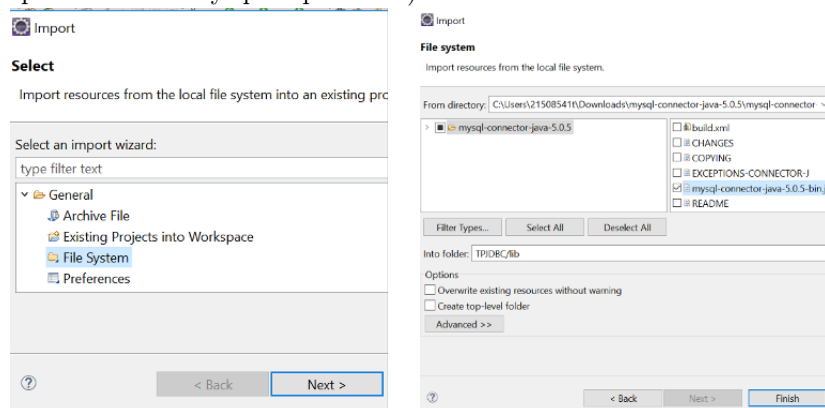
## 1 Configuration de l'environnement de travail

### 1.1 Ajout du driver JDBC

1. Créez un projet java POIProject
2. Créez un dossier lib au niveau du projet comme suit



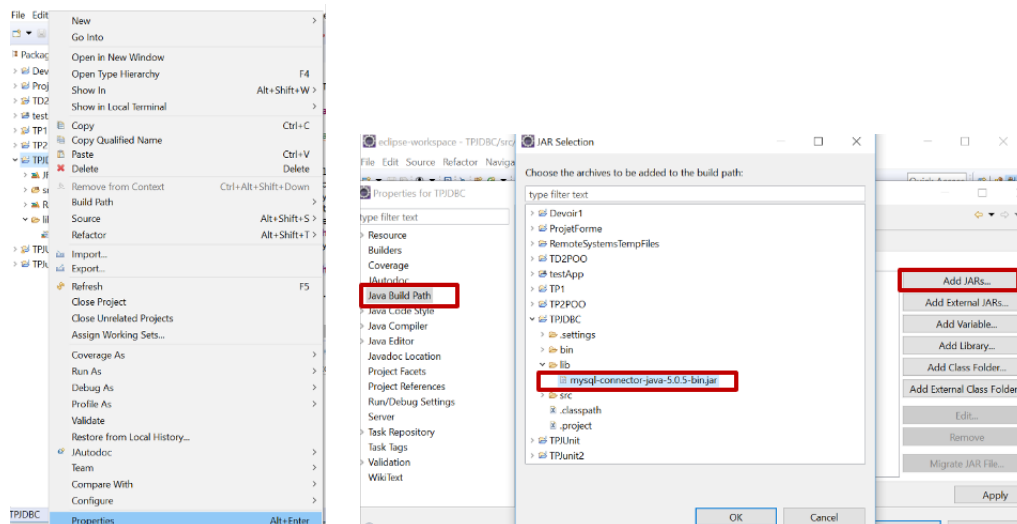
3. Puis, importez dans le dossier lib le driver jdbc mysql comme suit (vous devez l'avoir téléchargé depuis le site de mysql au préalable) :



4. Le résultat de l'importation est le suivant :



5. Informez votre projet Eclipse de ce driver en déclarant la nouvelle librairie dans le chemin ou build path de votre projet :



## 1.2 Création et gestion de la base de données

Nous allons pour cela utiliser le logiciel MySQL Workbench qui doit être installé sur vos machines. Si ce n'est pas le cas, ou si vous faites le TP depuis chez vous, téléchargez et installez le logiciel au préalable<sup>1</sup>.

Une fois dans MySQL Workbench

1. Démarrez le serveur MySQL : Menu démarrer → Dossier MySQL → MySQL server start
2. Importez le fichier SQL contenant la base de points d'intérêt (File > Run SQL Script...). POSITIONNEZ "Default Character Set" à "utf8". Le fichier `poi.sql` est à télécharger au préalable sur Celene

## 2 Implémentation de l'application

### 2.1 Quelques indications

1. Votre programme doit fournir les fonctionnalités suivantes :

---

```

1 findById(Long id) throws SQLException;
2 List<Place> findByName(String name) throws SQLException;
3 List<Place> findAll() throws SQLException;
4 void save(Place place) throws SQLException;
5 void update(Place place) throws SQLException;

```

---

2. L'établissement de la connexion à la base de données s'effectuera dans votre classe de test
3. Il est très simple de trouver les coordonnées d'un lieu depuis Google Map (bouton droit sur la carte, puis il suffit de cliquer sur les coordonnées pour les copier dans le presse-papier)

### 2.2 Exploitation des données géographiques

Nous souhaitons calculer la distance entre 2 points sur la terre et cela, indépendamment de la version de MySQL. Le but est de proposer les POI se trouvant à un certain rayon de votre position.

Vous pouvez vous aider de cette implémentation : <https://www.plumislandmedia.net/mysql/stored-function-haversine-distance-computation/>

1. Faites évoluer votre programme pour prendre en compte cette méthode :

<sup>1</sup><https://www.mysql.com/fr/products/workbench/>

---

```
1 List<Place> near(double latitude, double longitude, int radius) throws SQLException
    ;
```

---

2. Astuce : utilisez le mot clé HAVING dans votre requête SQL

## 3 Pour aller beaucoup plus loin

### 3.1 Implémentation

1. Affichez vos POI sur une carte embarquée dans une interface JavaFX
2. Prévoyez également un champ de recherche permettant de trouver un POI par son nom
3. ...

### 3.2 Quelques pistes

1. Installer JavaFX
2. (Optionnel) Configurer JavaFX et SceneBuilder dans votre IDE préféré (Eclipse, IntelliJ, ...)
3. Plusieurs librairies à votre disposition :
  - (a) mapjfx (<https://github.com/sothawo/mapjfx>)
  - (b) Gluon Maps : <https://gluonhq.com/labs/maps/>
  - (c) ...