

Projet Java Paradise

Nous allons développer un logiciel de gestion pour une Agence de voyage. Ce projet se nommera *Java Paradise*.

Nous utiliserons la console pour interagir avec la solution et la base de données MySQL pour sauvegarder les données.

Le projet devra proposer un menu en Anglais.

Sachez que le client insiste pour l'utilisation des bonnes pratiques Java, et l'utilisation exclusive de l'Anglais.

La console devra ressembler à cela :

```
Welcome aboard !

What do you want to do ?
1 - Add a place
2 - Find a place
3 - Edit a place
4 - Remove a place
5 - Add a trip
6 - Find a trip
7 - Remove a trip
8 - Quit

1
Name: Tokyo
Place added with the ID=1.
```

Roadmap

I. DÉMARRAGE DU PROJET

Vous avez 2h pour poser toutes vos questions, m'envoyer en MP **le lien de votre Github avec un diagramme de classe** réalisé sous un format consultable directement depuis le Github (png, jpg, pdf, html)

Ensuite, vous aurez la possibilité de véritablement commencer la partie dev !

DEADLINES :

- 4 juin 11h : lien du projet sur Github avec un diagramme de classe
- 4 juin 23h59 : lien du projet sur Github finalisé (v1 beta ou finale)
- 6 juin 23h59 : lien du projet sur Github finalisé (v1 finale)

II. SUITE DU PROJET

Vous pouvez mettre en place le projet indiqué dans la partie III.

La partie IV est un BONUS ! Réaliser l'interface avec Swing ou JavaFX

```
What do you want to do ?
1 - Add a place
2 - Find a place
3 - Edit a place
4 - Remove a place
5 - Add a trip
6 - Find a trip
7 - Remove a trip
8 - Quit

1
Name: Paris
Place added with the ID=2.
```

```
What do you want to do ?
1 - Add a place
2 - Find a place
3 - Edit a place
4 - Remove a place
5 - Add a trip
6 - Find a trip
7 - Remove a trip
8 - Quit

5
Departure: Please enter the id of the place : 2
Place : Paris
Destination: Please enter the id of the place : 1
Place : Tokyo
Price: 999
Trip added with the ID=2.
```

```
What do you want to do ?
1 - Add a place
2 - Find a place
3 - Edit a place
4 - Remove a place
5 - Add a trip
6 - Find a trip
7 - Remove a trip
8 - Quit

4
Please enter the id of the place : 1
Place : Tokyo
All trips with this Place will be removed with it.
Are you sure you want to remove it ? [yes/no]
yes
Place removed !
```

III. VERSION PROJET CONSOLE

1) Créer un package **com.formation.app**

Créer une classe Launcher qui aura la méthode main (point de lancement de votre application Java)

2) Créer un sous-package **model**

Créer deux classes dans ce package : **Place** et **Trip**

- **Place** devra avoir les propriétés suivantes : un id (Long) et un nom (String).
- **Trip** devra avoir les propriétés suivantes : un id (Long), un lieu de départ, un lieu d'arrivée et un prix (Float)

Ces deux classes devront respecter la convention JavaBeans.

3) Créer un sous-package **util**

- Créer une classe ConnectionManager
- Créer un attribut statique de type Connection
- Créer une méthode *static getConnection()* qui renverra la valeur de l'attribut Connection si sa valeur n'est pas nulle, sinon, si Connection a une valeur nulle, assignez lui comme valeur l'instance de votre connexion créée avec le Driver MySQL, puis retourner la valeur de Connection
- Créer une méthode *static close()* qui fermera la connexion

4) Créer un sous-package **dao**

Créer deux interfaces : PlaceDao et TripDao

Ces deux interfaces devront avoir 5 signatures de méthodes : (exemple pour PlaceDao)

- createPlace(Place place) retournera un Long
- findPlaceById(Long id) retournera une Place
- updatePlace(Place place) retourne un booléen
- removePlace(Place place) retournera un booléen
- findAllPlace() retournera une liste de Place

5) Toujours dans ce package, Créer une classe **DaoFactory**

- DaoFactory aura un constructeur privé vide
- DaoFactory aura deux méthodes :
 - GetTripDao() : statique et retourne une instance de JdbcTripDao()
 - GetPlaceDao() : statique et retourne une instance de JdbcPlaceDao()

6) Créer un sous-package **jdbc** dans **dao**

- Créer trois classes : JdbcDao, JdbcPlaceDao, JdbcTripDao
- JdbcDao est une classe abstraite
- JdbcTripDao et JdbcPlaceDao sont les implémentations de vos interfaces respectives et sont également des classes filles de JdbcDao
- JdbcDao aura un attribut Connection en protected. Il faudra utiliser la classe utilitaire (située dans le package com.intifomation.app.util) pour l'instancier
- Les classes filles JdbcTripDao et JdbcPlaceDao utiliseront l'attribut de la classe mère

Vous pouvez désormais créer une base de données MySQL qui contiendra deux tables :

- Place
- Trip

Ces deux tables auront la même structure que vos JavaBeans, et vos id seront les clés primaires et auto incrémentés. Ils seront également liés par une relation, à vous de la trouver et de la définir.
(Relation one-to-one, one-to-many, many-to-many...)

7) La classe Launcher

- Vous pourrez utiliser le scanner pour créer un Place en base de données
- Vous pourrez utiliser le scanner pour créer un Trip en base de données

8) Les 5 Méthodes de vos interfaces

- L'implémentation de vos méthodes sont des requêtes SQL exécutées avec Java, pour faire des SELECT, UPDATE, DELETE....
- N'oubliez donc pas d'utiliser :
 - PreparedStatement
 - ResultSet

Bonne réussite !

IV. PROJET VERSION GRAPHIQUE

Une fois votre projet mis en place en console, vous pouvez être fier de vous !!

Désormais, vous pouvez mettre en avant vos talents à travers la mise en place d'une interface graphique, afin de rendre votre logiciel utilisable à travers une fenêtre.

Ressources :

<https://www.tutorialspoint.com/swing/index.htm>

<https://www.javatpoint.com/java-swing>

<https://www.jmdoudoux.fr/java/dej/chap-swing.htm>

<https://www.guru99.com/java-swing-gui.html>

Encore...une Bonne réussite !