TP Hibernate
Approche Buttom-up

Le précédent TP vous a permis de tester l'approche dite *top-down* ou descendante, le point de départ étant un modèle objet. L'objectif de ce TP est de vous montrer l'approche *bottom-up* ou montante, le point de départ étant cette fois-ci une base de données existante.

Les documents et fichiers nécessaires au bon déroulement de ce TP sont disponibles à l'adresse : `http://www.lamsade.dauphine.fr/~manouvri/HIBERNATE/TP_HIBERNATE/TP_BottumUp.html`

1. Création de la base de données

La base de données utilisée pendant ce TP va être la base de données que vous aviez créée lors du TP JDBC.

Penser à supprimer les relations créées suite à l'exécution du TP sur le tutorial Hibernate

2. Création des POJO et des DAO par Hibernate à partir d'une base de données existante

Hibernate peut générer les POJO, DAO et fichiers de correspondance à partir d'une base de données existante. Les étapes à suivre sont les suivantes. Elles sont reprises du document en ligne *Introduction à Hibernate Tools* de Thibault HUET (Elève-Ingénieur Supinfo Paris) – disponible à l'adresse `http://supinfo-projects.com/fr/2006/hibernate%5Ftools`

1. Créer un nouveau projet Java sous Eclipse.
2. Ajouter dans l'onglet Librairies (*Add External JARs*) les fichiers .jar suivants :
   - Les fichiers `.jar` contenu dans le répertoire `/Volumes/Space/Staff/mmanouvrier/TP_HIBERNATE/lib/hibernate/`
   - Le fichier `.jar` correspondant au pilote JDBC du SGBD avec lequel vous allez travailler (ex. `postgresql-8.4-701.jdbc3.jar` pour PostgreSQL ou cf. `http://jdbc.postgresql.org/download.html#jars`).
3. Ne pas ouvrir la perspective Java.
4. Cliquer sur Control N et sélectionner *Hibernate Configuration File (cfg.xml)* et cliquer sur le bouton *Next* deux fois ou cliquer avec le bouton droit de la souris sur votre projet, puis *New* et sélectionner *Hibernate Configuration File (cfg.xml)* et cliquer sur le bouton *Next* une seule fois.

   Rappel : le fichier hibernate.cfg.xml correspond au fichier de configuration qui va permettre de se connecter à la source de données.
5. Configurer la connexion en remplissant les champs suivants avec les valeurs indiquées :
   a. `Database dialect` : sélectionner `PostgreSQL`.
   b. `Driver class` : sélectionner `org.postgresql.Driver`
   c. `Connection url` : indiquer l'adresse de la base de données
   d. `Username` : indiquer votre login de connexion à la base de données
   e. `Password` : indiquer votre mot de passe de connexion à la base de données
6. Cochez la case *Create a console configuration* et cliquer sur *Next*.
7. Dans `Classpath`, ajouter le `.jar` du pilote JDBC et cliquer sur Finish.

8. Pour vérifier que la connexion se fait correctement dans l'onglet *Hibernate Configurations*, cliquer sur la console de configuration qui a été créée si vous avez bien coché *Create a console configuration* à l'étape 6. *Si tous les paramètres ont été parfaitement saisis vous pouvez y lister les relations de la base de données.*
9. Dans le menu *Run*, sélectionner *Hibernate Code Generation* …et créer une nouvelle configuration.
10. Choisir la `Console configuration` qui vient d'être créée.
11. Choisir un répertoire de destination qui va contenir les fichiers générés par Hibernate Tools.
12. Cochez la case *Reverse engineer from JDBC Connection* et définir un nom de package.
13. Passer à l'onglet *Exporters* et cocher les informations suivantes :
    a. `Domain code` : pour qu'Hibernate génère les POJO,
    b. `Hibernate XML Mappings` : pour qu'Hibernate génère les fichiers de correspondance de POJO
    c. `DAO code` : pour qu'Hibernate génère les classes de DAO (nommée `Home`)
    d. `Hibernate XML Configuration` : pour qu'Hibernate mette à jour le fichier de configuration `hibernate.cfg.xml`.

Si tout c'est bien passé, les fichiers correspondant ont été créés[1].

3. Travail à effectuer

Penser à consulter l'aide en ligne de Hibernate :
`http://www.hibernate.org/hib_docs/v3/reference/fr/` ainsi que les transparents de cours (`http://www.lamsade.dauphine.fr/~manouvri/HIBERNATE/SLIDES/`)

Choisir 3 classes associées parmi les classes du modèle et créer un programme permettant de tester leurs méthodes ainsi que les méthodes de leur DAO.

4. Annexe – Outils nécessaires pour réaliser ce TP chez vous

Pour réaliser ce TP sur votre propre machine, vous devez avoir installé les logiciels et *plugins* suivants :

- Eclipse 3.2 (`http://www.eclipse.org/`)
- Hibernate 3.2 – package *Hibernate Core* (`http://www.hibernate.org`)
- *Hibernate Tools* (`http://www.hibernate.org`)
- Un SGBD (ex. PostgreSQL - `http://www.postgresql.org/` et son interface PgAdmin3 - `http://www.pgadmin.org/`)
- Un pilote JDBC (ex. `postgresql-8.1-408.jdbc3.jar` `http://jdbc.postgresql.org/`)

Pour installer et utiliser le *plugin Hibernate Tools*, vous pouvez consulter :
`http://www.hibernate.org/hib_docs/tools/reference/en/html/` ou bien
`http://supinfo-projects.com/fr/2006/hibernate%5Ftools/1/`

[1] La version de Java disponible au CRIO UNIX étant la 1.4, L'*autoboxing* (nouveauté de Java 5.0 permettant de transformer un type de variable de type primitif en son type objet correspondant et ceci automatiquement ~ cast automatique) ne sera pas disponible. Vous aurez donc sans doute des erreurs pour les méthodes `findById()` des DAO qui attendent en paramètre un `Serializable` et reçoivent un `int`. Vous devrez donc dans ces méthodes remplacer le type `int` du paramètre par `java.lang.Integer`.