**Atelier 08 : Connexion à la base de données avec Java**

2ème Année Licence TI – Parcours DSI – MDW / Module Optionnel – Semestre I

**Ateliers Java**
**Mme Mouna KORT**
**ISET Djerba**

**Objectifs :**

- Présenter un moyen qui permet à un programme Java de se connecter à une BD avec JDBC
- S'entrainer aux opérations d'ajout, modification et suppression des données

**Notes de cours**

Pour se connecter à la base de données, on utilise L'API JDBC définie dans le package java.sql. Le processus de la connexion à la base de données passe par les étapes suivantes.

1. Charger le driver JDBC
2. Établir la connexion à la base de données
3. Créer une variable de description de la requête SQL
4. Exécuter la requête
5. Traiter les données retournées
6. Fermer la connexion établie

Dans ce qui suit, on procède à la réalisation de ces étapes à travers un exemple.

**Partie A : Exemple de connexion à la base de données**

1. En utilisant l'outil Xampp, ou autre outil, ouvrir le PhpMyAdmin et créer une nouvelle base de données intitulée javaDB comportant une table Etudiant qui possède les champs : cin, nom et prenom. Insérer quelques valeurs d'étudiants selon votre choix.
2. Dans l'environnement de développement Eclipse, suivre les étapes suivantes.
   a. Créer un nouveau projet Java de nom ConnexionDB
   b. Télécharger la librairie mysql-connector (driver de connexion Java/MySql) et l'ajouter au projet à partir de "Java Build Path → librairies → Add External JARs…
   c. Créer un nouveau package Atelier08 et créer une nouvelle classe DemoDB
   d. Ajouter le code suivant comme exemple de connexion à la base de données et récupération de données

```java
import java.sql.*;

public class DemoDB {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
	try {
		//Paramètres de connexion
		//Chemin d'accès à la BD
		String url = "jdbc:mysql://localhost/javadb";

		//Paramètres de l'admin mysql
		String login = "root";
		String passwd = "";

		//Instance de la classe Connection
		//utilisée pour ses atts et ses meths. d'accès à la BD
		Connection cnx=null;

		// Chargement du driver : connecteur mysql
		Class.forName("com.mysql.cj.jdbc.Driver");

		// Récupération de la connexion
		cnx=DriverManager.getConnection(url,login, passwd);

		// Création d'un statement : Ensemble de méthodes
		// et d'attributs pour envoyer des commandes sql à la BD
		Statement st=cnx.createStatement();

		// Ecriture et exécution de la requête
		// Récupération des résultats
		ResultSet rs=st.executeQuery("select * from etudiant");

		//Affichage des données récupérées
		while(rs.next()) {
			//getString(arg) possède un argument q ui désigne
			// nom ou n° de la colonne de la table dans la BD
			System.out.println(rs.getString("cin")
			+ " " + rs.getString(2) + " " + rs.getString(3));
		}
		//Libération des ressources
		cnx.close();
		st.close();
		rs.close();
	}
	//En cas d'erreur,SQL ou Driver, affichage du message d'erreur
	catch(SQLException e| ClassNotFoundException e){
		System.out.println(e.getMessage());
		}
	}
}
```

**Remarque**

Comme l'interface Statement, il est possible d'utiliser l'interface PreparedStatement pour définir une requête SQL précompilée. Cette requête peut être paramétrée et exécutée plusieurs fois. En voici un exemple.

```
PreparedStatement pst = cn.prepareStatement("select * from
Etudiant where nom=? and prenom=?");
pst.setString(1, "Ben Salah"); // 1 remplace le 1er ? dans la requête
pst.setString(2, "Salah");     // 2 remplace le 2ème ? dans la requête
rs=pst.executeQuery();
```

**Partie B : Optimisation de la connexion**

Afin d'optimiser la connexion à la base de données, on se propose de créer une classe dédiée à cet objectif. Pour ce faire, créer la classe MaConnexion ayant les attributs et les méthodes suivants.

- url, login et passwd de type String
- cnx de type Connection
- st de type Statement
- rs de type ResultSet
- un constructeur MaConnexion() initialisant les attributs
- les getters getStatement() et getResultSet( )
- la méthode seConnecter() qui permet de se connecter à la BD
- la méthode liberer() qui permet de libérer les ressources utilisées

**Partie C : Gestion des étudiants**

On souhaite créer une application pour la gestion des étudiants. Pour ce faire, on propose de suivre les étapes suivantes.

1. Créer la classe Etudiant caractérisée par les attributs : cin, nom et le prénom
2. Créer une classe GestionEtudiants qui renferme les méthodes suivantes
   - void inserer (Etudiant e) : ajoute l'étudiant e à la table etudiant
   - void supprimer (int cin) : supprime l'étudiant ayant le n°cin passé en paramètre
   - void modifier (int cin, String nom) : modifie le nom de l'étudiant dont le n°cin est passé en paramètre
   - Etudiant trouverParCin (int cin) : renvoie l'étudiant de la table etudiant dont le n°cin est passé en paramètre

- void listerEtudiants ( ) : permet d'afficher les étudiants qui existent dans la table etudiant.

Il importe de rappeler qu'il faut utiliser la classe MaConnexion pour établir la connexion à la BD.

**Partie D : Test de la Gestion des étudiants**

Dans une classe de TestGestionEtudiants, afficher le menu suivant.

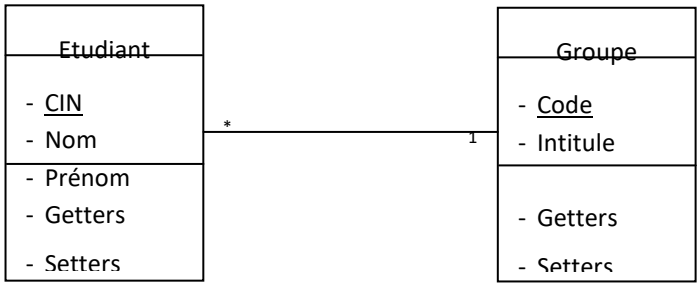
```
1. Ajouter étudiant
2. Modifier étudiant
3. Chercher étudiant
4. Lister les étudiants
5. Supprimer étudiant
6. Quitter

Entrer votre choix : .....
```

Selon le choix saisi, appeler la méthode concernée et réafficher de nouveau le menu.

**Partie E : Relation entre tables**

On se propose d'ajouter l'implémentation de la relation suivante entre Etudiant et Classe.




1. Mettre à jour la classe et la table Etudiant
2. Ajouter la classe et la table Groupe
3. Dans la classe GestionEtudiants, ajouter la méthode getSesEtudiants(String code), qui renvoie la liste des étudiants du groupe dont le code est passé en paramètre.
4. Dans le menu de l'application ajouter l'option "Afficher Registre" qui affiche le code et intitulé du groupe ainsi que tous ses étudiants

   CIN |Non |Prénom
   CIN |Non |Prénom
   CIN |Non |Prénom

Remarques

- La méthode toString() de la classe Etudiant renvoie <cin> →|<nom>→|<prenom>
- La méthode toString() de la classe Groupe renvoie <code> <intitule>

*Bon travail...*