Université Moulay Ismaïl
École Normale Supérieure de Meknès
Département des sciences

Programmation oriontée objet en JAVA
CH6 (JDBC)

Pr. Mehdi Moukhafi

JDBC

Java DataBase Connectivity

Qu'est-ce que la JDBC ?

- Une API (Application Programming Interface) qui permet d'exécuter des instructions SQL JDBC fait partie du JDK (Java Development Kit)

- Toutes les classes et interfaces sont dans le package java.sql :

```
import java.sql.*;
```

JDBC : structure d'un programme

1. On établit une connexion avec une source de données
2. On effectue des requêtes
3. On utilise les données obtenues pour des affichages, des traitements statistiques etc.
4. On met à jour les informations de la source de données
5. On termine la connexion
6. Eventuellement, on recommence en 1

Connexion

Pour établir une connexion à une base de données, il faut :

1. Connaître son nom
2. Lui associer un pilote (ou driver)

Déclaration du driver

Pour charger un driver, on peut utiliser la méthode

```
Class.forName(String);
```

Pilote utilisé pour MySql :

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

• Pour charger les archives sous Eclipse :

Avec l'onglet « Libraries », choisir « Add external jars »

Déclaration du driver

Pour charger un driver, on peut utiliser la méthode

```
Class.forName(String);
```

Pilote utilisé pour MySql :

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

• Pour charger les archives sous Eclipse :

Avec l'onglet « Libraries », choisir « Add external jars »

Exemple de test du driver

```
package DAO;


import java.sql.*;
public class premiereConnexion {
public static void main(String[ ] args){
try{
Class.forName("com.mysql.cj.jdbc.Driver");
}
catch (ClassNotFoundException e){
System.out.println("Impossible de charger le pilote");
System.exit(1);
}
System.out.println("Pilote chargé");


} /*du main*/
} /* de la classe premiereConnexion*/
```

Nommage des bases de données : les URL

<protocole> : <sous-protocole>://<adresse IP>:<Port>/<nom de la base de données>

- Protocole : jdbc
- Sous-protocole : le driver.
- Adresse IP du serveur BD
- le port : par defaut de mysql c'est 3306

Etablir la connexion

On utilise la méthode getConnection( ) de la classe DriverManager avec l'URL en argument :

```
Connection conn = DriverManager.getConnection(url, login, mdp);
```

On definit aussi le nom de l'utilisateur et le mot de passe

exemple de connexion MySql :

```
Connection conn = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/test1", "root", "");
```

Déclaration et exécution de la requête

Il faut tout d'abord demander la création du statement

```
Statement stmt = conn.createStatement();
```

Ensuite il faut déclarer le code SQL de la requête

```
 ResultSet resultat = stmt.executeQuery( »ma
                      requette");
```

Exemple :

```
ResultSet resultat = stmt.executeQuery("SELECT *
                     FROM Persons");
```

Exploitation des résultats

On peut parcourir les lignes de l'objet ResultSet avec la méthode next( ). Cette méthode renvoie VRAI s'il reste des lignes à lire et FAUX sinon :

```
 while(resultat.next()) {
//traitement des données récupérées
}
```

Exemple :

```
   while(resultat.next()) {
   System.out.println(resultat.getInt("ID")+"  "+result
   at.getString("LastName")+"  "+resultat.getString("Fi
   rstName")+"  "+resultat.getString("Age"));

      }
```

Exploitation des résultats

On peut récupérer les données selon leur type grâce au méthodes suivantes :

```
x=resultat.getInt("xBD");   // recupérer un int
y=resultat.getFloat ("yBD"); // recupérer un float
z= resultat.getDouble ("zBD"); // recupérer un double
w= resultat.getBoolean ("wBD"); // recupérer un Boolean
```

Correspondances entre les types SQL et les types JAVA

CHAR, VARCHAR, LONGCHAR	String
NUMERIC, DECIMAL	java.math.BigDecimal
BIT	boolean
TINYINT	byte
SMALLINT	short
INTEGER	int
BIGINT	long
REAL	float
FLOAT, DOUBLE	double
BINARY, VARBINARY, LONGVARBINARY	byte[ ]
DATE	java.sql.Date
TIME	java.sql.Time
TIMESTAMP	java.sql.Timestamp

Accès en mise à jour

En outre de l'exécution d'un `select` avec executeQuery, on peut aussi exécuter un `update`, un `insert` ou un `delete` :

```
Statement stmt = conn.createStatement();
//insert
stmt.executeUpdate("INSERT INTO `Persons`(`ID`, `LastName`,
`FirstName`, `Age`) VALUES ('11','sfhgfgh','sdfhgdfghd','12')");

//update
stmt.executeUpdate("UPDATE `Persons` SET `LastName`='mehdi',
`FirstName`='moukhafi',`Age`='35' WHERE  ID='1' »);

//delete
stmt.executeUpdate("DELETE FROM `Persons` WHERE  ID='18'");
```

Déconnexion

Libérer les objets Resultat et Statement :

```
stmt.close();
resultat.close();
```

Fermer la connexion :

```
conn.close();
```