ESTIA 2ème année
2018/2019

JDBC : TP 1

Connexion à une base de données avec JAVA

Exercice 1.1 *Environnement de travail*

1. Télécharger le fichier `mysql-connector-java-5.1.40-bin.jar` à l'adresse suivante :

`http://www.labri.fr/perso/preuter/POO2018/JDBC/`

- Chaque fichier qui utilise la bibliothèque de JDBC doit faire les importations suivantes:

```
import java.sql.*;

// import java.sql.Connection;
// import java.sql.DriverManager;
// import java.sql.Statement;
```

- Chaque projet qui utilise la bibliothèque de JDBC doit associer le fichier `mysql-connector-java-5.1.40-bin.jar`. Ceci est réalisé par :
  - avec Eclipse : (clique droit sur le nom de votre projet, et ensuite :) *Build-Path* → Add External Archives → `mysql-connector-java-5.1.40-bin.jar`
  - en ligne de commande : rajouter à votre variable d'environnement `CLASSPATH` le fichier `mysql-connector-java-5.1.40-bin.jar`
- Choisissez des noms des classes avec des lettres en majuscule, des noms de variables avec des lettres en miniscules.
- Dans un fichier, vous allez définir une seule classe. Le nom de ce fichier sera le nom de la classe suivi de l'extension `.java`
- N'utilisez jamais des caractères spéciaux ni des accents dans vos noms de fichiers/classes/variables.

Exercice 1.2 *Un menu pour proposer différents choix*

1. Tapez le code suivant dans une classe `MenuRun.java`.

```
import java.util.Scanner;

public class MenuRun
{
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        boolean fin  = false;
        while (!fin)
        {
            System.out.println("Faites votre choix :");
```

```
            System.out.println("---------------------------------------");
            System.out.println("(1) Montrer tous les employés");
            System.out.println("(2) Insérer un employé");
            System.out.println("(3) Vérfier login/mot de passe d'un employé");
            System.out.println("(4) Quitter");
            System.out.println("---------------------------------------");
            System.out.println("");


            // entrée clavier entier
            String choixString = scan.nextLine();
            int choix = Integer.parseInt(choixString);

            switch (choix)
            {
                case 1 :
                    System.out.println("1 - Les employées :");
                    break;
                case 2 :
                    System.out.println("2 - Insérer un employé :");
                    System.out.print("Nom   : ");
                    String nom = scan.nextLine();
                    System.out.print("login : ");
                    String login = scan.nextLine();
                    System.out.print("mdp   : ");
                    String mdp = scan.nextLine();
                    System.out.println("***");
                    // Insérer l'employé dans la table employé
                    break;
                case 3 :
                    System.out.println("3 - Vérification :");
                    // Demander à l'utilisateur de rentre login et mot de passe
                    // Vérifier si :
                    // - le login existe
                    // - le mot de passe est le bon mot de passe
                    break;
                case 4 :
                    fin = true;
                    break;
            }

        }
    }
}
```

2. Exécutez votre code.

Exercice 1.3 *Créer une base de données avec MySQL*

1. Lancez tout d'abord le gestionnaire de base de donnée (SGBD) MySQL grâce au programme WAMP (MAMP sous OSX, LAMP sous Linux). Une fois celui-ci démarré, lancez phpMyAdmin (l'interface de configuration de votre BDD).

2. Créer une base de données sur laquelle vous allez travailler. Pour ce faire, cliquez sur l'onglet "Bases de données" et créez une nouvelle base que vous nommerez `estiajdbc` (laissez les options par défaut). Créez ensuite une nouvelle table (onglet "Structure") que vous nommerez `employe`. Cette table contiendra 5 colonnes.

- un champs de type INT qui s'appelle `id` **avec les attributs clé primaire ainsi que "auto-increment"**,
- trois champs de type TEXT qui s'appellent `nom`, `login`, et `mdp`, et
- un champs de type DATETIME qui s'appelle `date` et qui sera la date d'inscription de l'employés.

3. Insérer deux employés dans votre table `employe` en utilisant l'interface phpMyAdmin. Cryptez votre mot de passe avec la fonction MD5 qui vous est proposé dans la colonne pour insérer le `mdp`. Choisissez une `date` arbitraire dans le passé.

Exercice 1.4 *Classe pour connexion à la base de données via JDBC*

```
// importation des bibliothèques JDBC
import java.sql.*;

public class EmployeModel
{
    private Connection _conn;

    public EmployeModel()
    {
        String nomUser = "root"; // Utilisateur de la BD
        String passwd = "root"; // Password de l'utilisateur de la BD
        String url = "jdbc:mysql://localhost/"; // Serveur de la BD
        String nomBase = "estiajdbc"; // Nom de la BD sur laquelle nous allons acceder

        _conn = null;

        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            _conn=DriverManager.getConnection(url+nomBase, nomUser, passwd);
            System.out.println("Je me connecte a la base de donnees: " + nomBase);
        }

        catch (SQLException ex1)
        {
            System.out.println("J'ai détecté une erreur de type SQL: " + ex1.getMessage());
        }

        catch (Exception ex2)
        {
            System.out.println("J'ai détecté une erreur de type lang: " + ex2.getMessage());
        }
```

```
        System.out.println("Base de donnée connecté");
    }

    // fonction pour montrer
    public void montrer()
    {
        try
        {
            String requete = new String("SELECT id, nom, login FROM employe;");
            Statement stmt = _conn.createStatement();
            ResultSet rs = stmt.executeQuery(requete);

            while ( rs.next() )
            {
                    String nom = rs.getString("login");
                    System.out.println(nom);
                    // à affiner pour montrer les trois champs
            }
        }
        catch (SQLException ex3)
        {
            while (ex3 != null)
            {
                System.out.println(ex3.getSQLState());
                System.out.println(ex3.getMessage());
                System.out.println(ex3.getErrorCode());
                ex3=ex3.getNextException();
            }
        }
    }

    // fonction à remplir : vérifier l'existance de l'utilisateur/mdp, retourner soit son id, soit -1.
    public int verifier(String login, String mdp)
    {
    }

    // fonction à remplir : insérer un nouvel enregistrements avec la date actuelle (utiliser NOW() en MySQL)
    public void inserer(String nom, String login, String mdp)
    {
    }
    public void fermerConnexion()
    {
        try
        {
            if (_conn != null)
                _conn.close();
        }
        catch (SQLException ex1)
        {
            System.out.println("J'ai détecté une erreur de type SQL: " + ex1.getMessage());
        }
    }
}
```

Exercice 1.5 *Lier les deux classes*

L'objectif de ce TD est de remplir les fonctionnalités manquantes dans la classe `MenuRun`. Pour cela, en début de la méthode `main`, vous devriez instancier un objet de la classe `EmployeModel`, et appelez les bonnes méthodes aux bons endroits. Ensuite, il faudra implémenter les méthodes manquantes de la classe `EmployeModel`.