

L3 MIAGE – Java avancée – TD3

Exercice 1

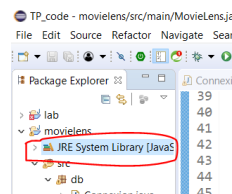
L'objectif de l'exercice est d'utiliser **Java** pour se connecter à une base de données **MySQL**, et afficher son contenu dans une interface Swing.

La base de données utilisée est une extraction de la base du site movielens.org enrichi par des informations du site imdb.com, elle contient un seul tableau **movie** qui contient les champs suivants :

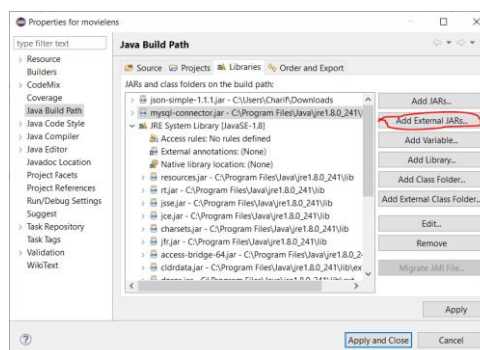
- **mov_id** *int(11) NOT NULL* : clef primaire
- **imdb_id** *varchar(255) DEFAULT NULL* : l'identifiant du film chez imdb
- **title** *varchar(256) NOT NULL* : le titre du film
- **poster** *varchar(256) DEFAULT NULL* : un lien vers l'affiche du film
- **genres** *varchar(256) DEFAULT NULL* : les genres du film (comédie, action, ...)
- **release_date** *varchar(256) DEFAULT NULL* : la date exacte (par jour) de sortie du film
- **type** *varchar(256) DEFAULT NULL* : le type du film (long, ...)
- **synopsis** *mediumtext* : un résumé de l'histoire du film
- **rating** *double DEFAULT NULL* : la note moyenne selon imdb
- **year** *bigint(20) DEFAULT NULL* : l'année de sortie du film
- **active** *bit(1) DEFAULT b'1'* : si le film est actif dans la base (1 pour tous)

Importer la librairie de connexion entre Java et MySQL

1. Télécharger le fichier **mysql-connecteur** depuis [archive](http://archive.apache.org/dist/mysql/connector-java/)
2. Cliquez droit sur **JRE system Library (JavaSE-1.8)** de votre projet dans le **package explorer** à gauche de l'écran.



3. Choisissez **Build Path**, puis **configure Build Path**.
4. Dans la liste à gauche allez vers **Java Build Path** (si ce n'est pas déjà le cas).
5. Dans le **java build path** allez vers la page **Libraries** (3ème page après Source et Projects).



6. Cliquez le bouton **Add External JARs**, puis naviguez vers le fichier que vous avez téléchargé, et ajoutez-le.

7. Cliquez sur **Apply and Close** pour terminer

La classe Movie :

1. Créez la classe Movie qui contient les paramètres suivants :

```
protected int id = 0;
protected String title = "";
protected String poster = "";
protected int year = 1995;
```

2. Ajoutez un constructeur qui prend deux paramètres en entrée (id, title).
3. Ajoutez une méthode toString() qui rend le titre du film et l'année de sortie
4. Ajoutez des *getters* pour les trois paramètres **id**, **title** et **year**. Ajoutez deux *setters* pour les paramètres **poster** et **year**.
5. Ajoutez le *getter* du paramètre poster qui doit être particulier car il rend un objet image au lieu de String. Ce getter va essayer de récupérer l'image depuis internet par le lien de poster, et rend null en cas d'échec (lien erroné par exemple) :

```
import java.awt.Image;
import java.io.IOException;
import java.net.*;
import javax.imageio.ImageIO;

public Image getPoster() {
    Image image = null;
    try {
        URL url = new URL(this.poster);
        image = ImageIO.read(url);
    } catch (IOException e) {
        return image;
    }
    return image;
}
```

Classe Connexion

Cette classe gère la connexion avec la base de données.

1. Créez la classe Connexion qui contient les paramètres suivants :
 - a. host : un lien vers le serveur de la base de données (localhost si la base est sur le même ordinateur, ce qui n'est pas le cas ici).
 - b. port : le port de la connexion
 - c. user : le nom d'utilisateur de la base de données
 - d. pass : le mot de passe de l'utilisateur
 - e. connected : un paramètre qui sert à savoir si on est bien connecté à la base (optionnel).
 - f. connexion : le paramètre qui gère la connexion
 - g. sql : un paramètre de type statement pour passer de commande SQL pour interroger la base.
 - h. MoviesId : une liste d'identifiants de tous les films de la base ; afin d'éviter de stocker tous les films dans la mémoire pour ne pas trop la charger.
 - i. currentId : l'index du film actuel de la liste MoviesId

```

public String host = "sql7.freemysqlhosting.net"; //localhost
public int port = 3306;
public String db = "sql7372020";
public String user = "sql7372020";
public String pass = "N75js3bs4H";
private Boolean connected = false;
private Connection connexion;
private Statement sql;
private List<Integer> MoviesId = new ArrayList<Integer>();
private int currentId = 0;

```

2. Créez le constructeur de la classe qui doit permettre de se connecter à la base et de récupérer les identifiants des films pour les stocker dans la liste :

```

public Connexion() {
    try{
        Class.forName("com.mysql.jdbc.Driver");
        String lien = "jdbc:mysql://" + host + ":" + port + "/" + db;
        System.out.println(lien);
        this.connexion=DriverManager.getConnection(lien,this.user,this.pass);
        this.connected = true;

        // interroger la base et récupérer tous les films
        sql = this.connexion.createStatement();
        ResultSet rs=sql.executeQuery("select * from movie");
        MoviesId.clear();
        // currentId pointe sur le premier film
        currentId = 0;
        while(rs.next()) {
            // l'identifiant est le 1er champ, et il est de type int
            MoviesId.add(rs.getInt(1));
        }

        //con.close(); //cette comande permet de fermer la connexion avec la base
        System.out.println("connected ...");
        this.connected = true;
    }catch(Exception e){
        System.out.println(e);
    }
}

```

3. Ajoutez la méthode getMovie qui récupère un film de la base données, et stocke ses informations dans un objet de type Movie avant de le rendre :

```

public Movie getMovie() {
    Movie movie = new Movie(0, "");

    try {
        sql = this.connexion.createStatement();
        ResultSet rs=sql.executeQuery("select * from movie WHERE mov_id = " +
            this.MoviesId.get(this.currentId));
        while(rs.next()) {
            // On sait que le 1er argument est l'id et que le 3ème est le titre
            movie = new Movie(rs.getInt(1), rs.getString(3));
            // On sait que le 4ème argument est le poster
            movie.setPoster(rs.getString(4));
            // On peut aussi appeler l'argument par son nom
            movie.setYear(rs.getInt("year"));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return movie;
}

```

4. Ajoutez une méthode de se déconnecter :

```

public void desconnecte() {
    try {
        this.connexion.close();
        connected = false;
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

La classe principale

Il est temps maintenant de créer notre interface. La classe principale a trois arguments :

```

public static JLabel lbl_image;
public static JLabel lbl_title;
public static Movie currentMovie;

```

1. Ajoutez la méthode **main** qui va créer une connexion vers la base et charger le premier film dans le paramètre **currentMovie**, puis ajouter le titre et l’affiche dans les deux labels :

```

public static void main(String[] args) {
    Connexion connexion = new Connexion();
    currentMovie = connexion.getMovie();

    lbl_title = new JLabel(currentMovie.getTitle());
    lbl_image = new JLabel(new ImageIcon(currentMovie.getPoster()));

    ...
}

```

2. Par la suite, la méthode main va créer un **frame** et y ajouter les deux labels :

```

JFrame frame = new JFrame();
frame.setSize(600, 600);

lbl_title.setBounds(150, 10, 300, 30);
frame.add(lbl_title);
frame.add(lbl_image);

frame.setVisible(true);
frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

```

3. Au milieu, on va ajouter le bouton **suivant** dont le rôle est de charger le film suivant dans l’interface. Avant d’ajouter le bouton, on va aller à la classe connexion pour ajouter la méthode **nextMovie** :

```

public Movie nextMovie() {
    this.currentId = (this.currentId + 1) % this.MoviesId.size();
    return this.getMovie();
}

```

Maintenant, on modifie la méthode main pour insérer le code du bouton **suivant** :

```

lbl_image = new JLabel(new ImageIcon(currentMovie.getPoster()));

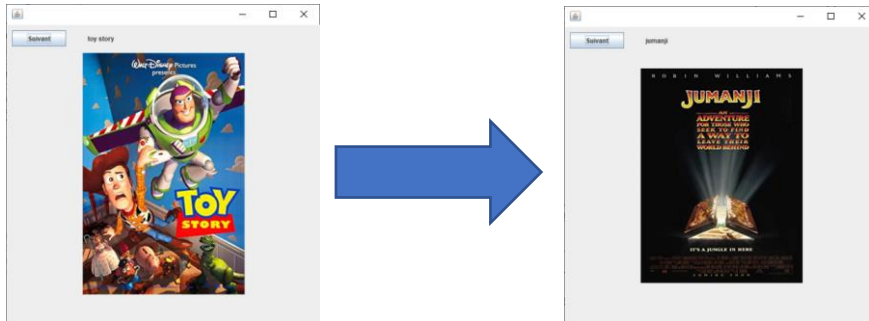
JButton btn_next = new JButton("Suivant");
btn_next.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        currentMovie = connexion.nextMovie();
        lbl_title.setText(currentMovie.getTitle());
        if(currentMovie.getPoster() != null)
            lbl_image.setIcon(new ImageIcon(currentMovie.getPoster()));
        else
            lbl_image.setIcon(null);
        frame.repaint();
    }
});

btn_next.setBounds(10, 10, 100, 30);
frame.add(btn_next);

lbl_title.setBounds(150, 10, 300, 30);

```

Voilà à quoi le résultat doit ressembler avant et après avoir cliqué sur le bouton « suivant » :



Exercice 2

Reprenez le projet de l'exercice 1, et faites le nécessaire pour ajouter les fonctionnalités suivantes :

1. Ajoutez les composants nécessaires à l'interface pour afficher les champs suivants du film (cela comprend aussi la modification de la classe Movie pour stocker leurs valeurs) :
 - a. Genres
 - b. Année de sortie
 - c. Synopsis
 - d. Rating
2. Ajoutez un bouton « précédent » qui permet le retour vers le film précédent.
3. Ajoutez un champ de texte qui affiche l'identifiant du film, et qui permet on le modifiant d'afficher le film correspondant à l'identifiant entré par l'utilisateur.