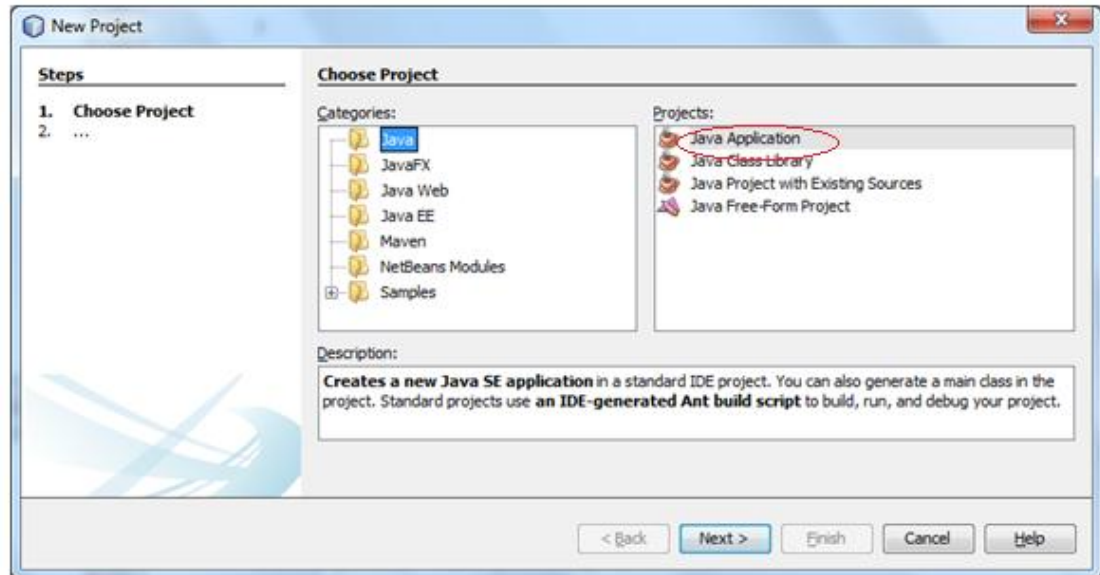


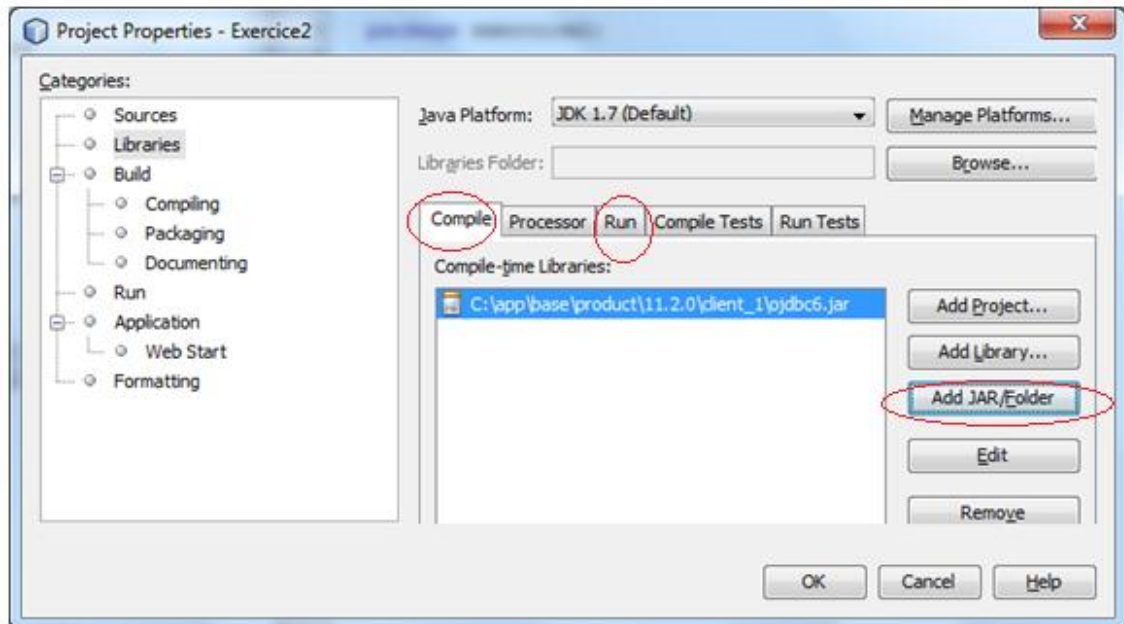
Comment démarrer un projet NetBeans utilisant une interface graphique?

Voici une description rapide du contenu de l'application qui vous a été envoyée.

- 1- Vous créez votre projet normalement (par le menu Fichier nouveau projet)



- 2- une fois créé, n'oubliez pas d'inclure les librairies .JAR



- 3- Par la suite, nous allons créer les classes suivantes :

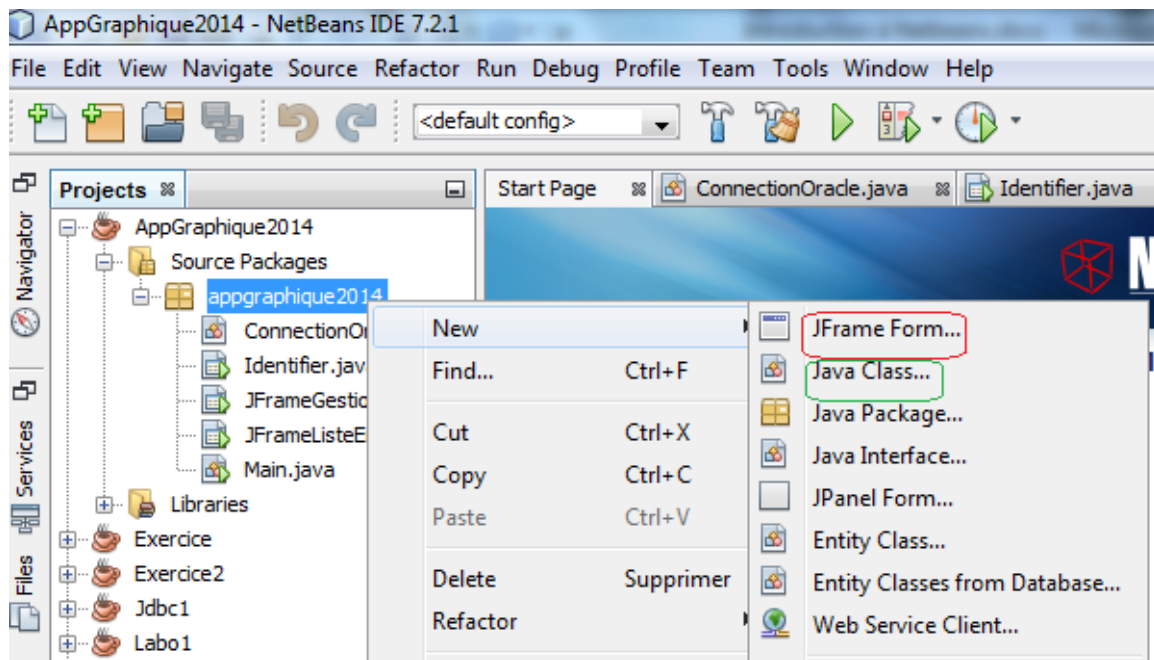
Une classe **ConnectionOracle** de type **Java Class**

Une classe **Main** de type **Java Class**

Une Classe **Identifier** de Type **Jframe Form**

Une Classe **JFrameGestion** de type **JFrame Form**

Une Class **JFrameLisEmp** de type **JFrame Form**



Rôle de la classe ConnectionOracle.

Cette classe a pour rôles :

- Charger le driver s'il y a lieu
- De définir les paramètres de connexion
- De définir deux méthodes connecter() et deconnecter()

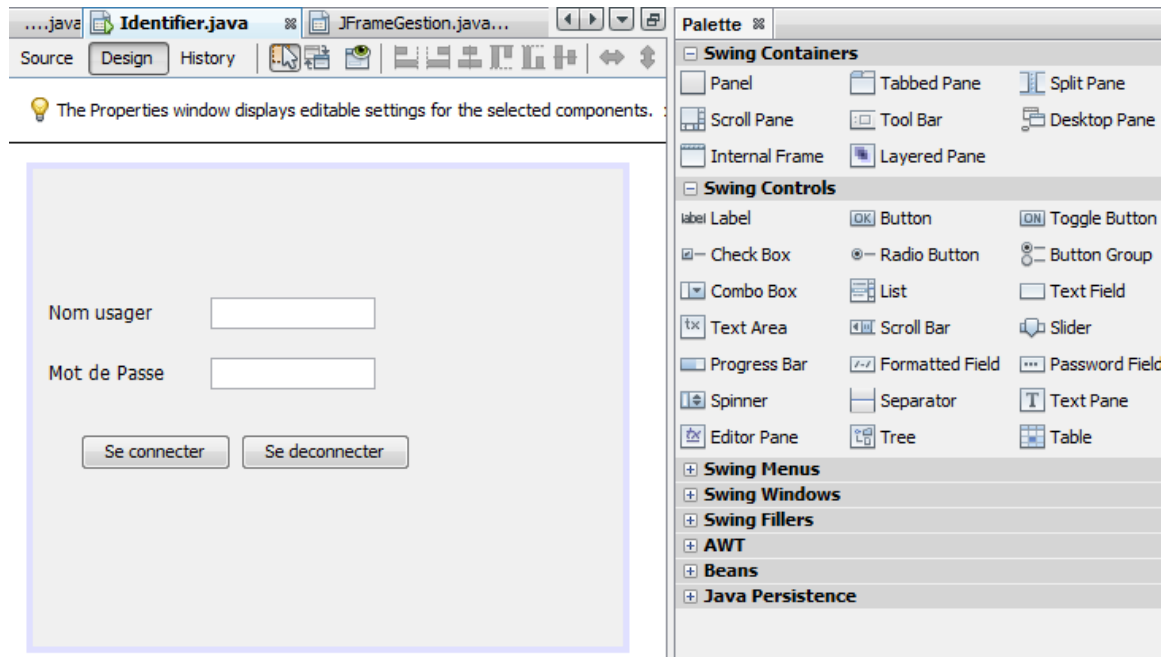
Cette classe doit utiliser les package : `java.sql.*` et `import oracle.jdbc.pool.*`;

```
import java.sql.*;  
  
import oracle.jdbc.pool.*;
```

Rôle de la classe Identifier de type JFrameForm

En mode Design :

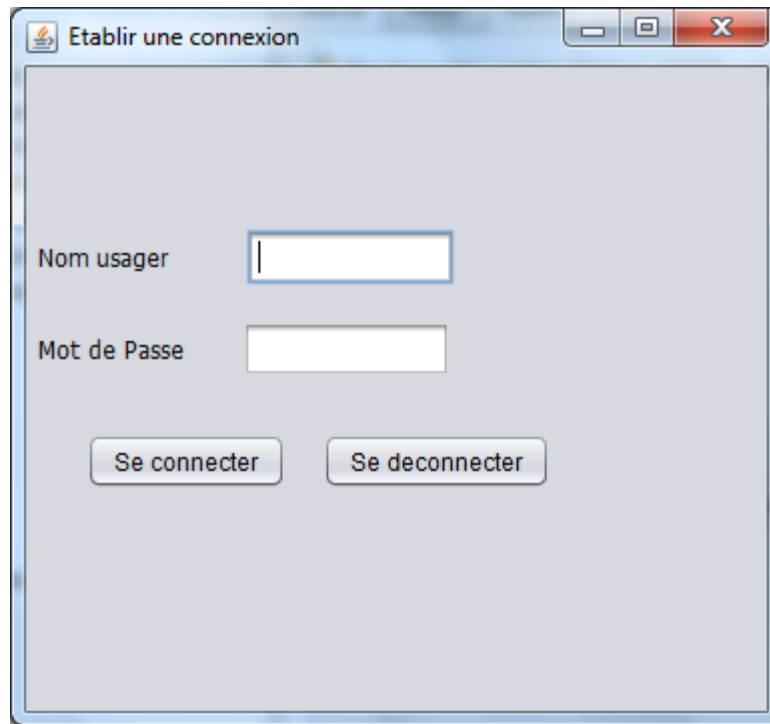
À Droite vous avez votre Jform et à gauche vous avez vos contrôles NetBeans.



En mode Source, vous devez importer les packages suivants :

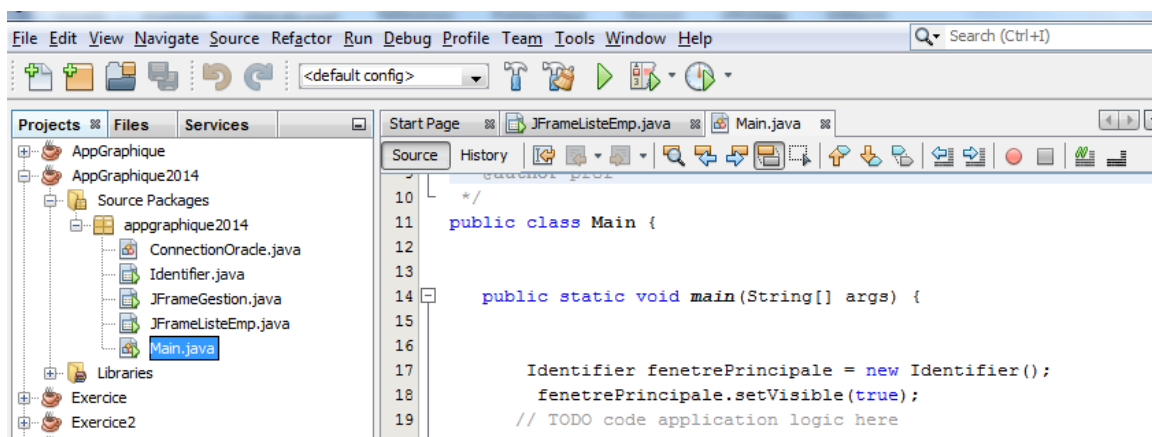
```
import javax.swing.*; // pour les interface graphique
import java.awt.*; // pour les interface graphique (version antérieur)
import java.sql.*;
```

À travers cette classe on saisit le nom d'utilisateur et le mot de passe pour obtenir une connexion à la base de données.



Rôle de la classe Main :

C'est cette Classe qui va appeler la classe Identifier. Voir code source dans la fenêtre suivante.

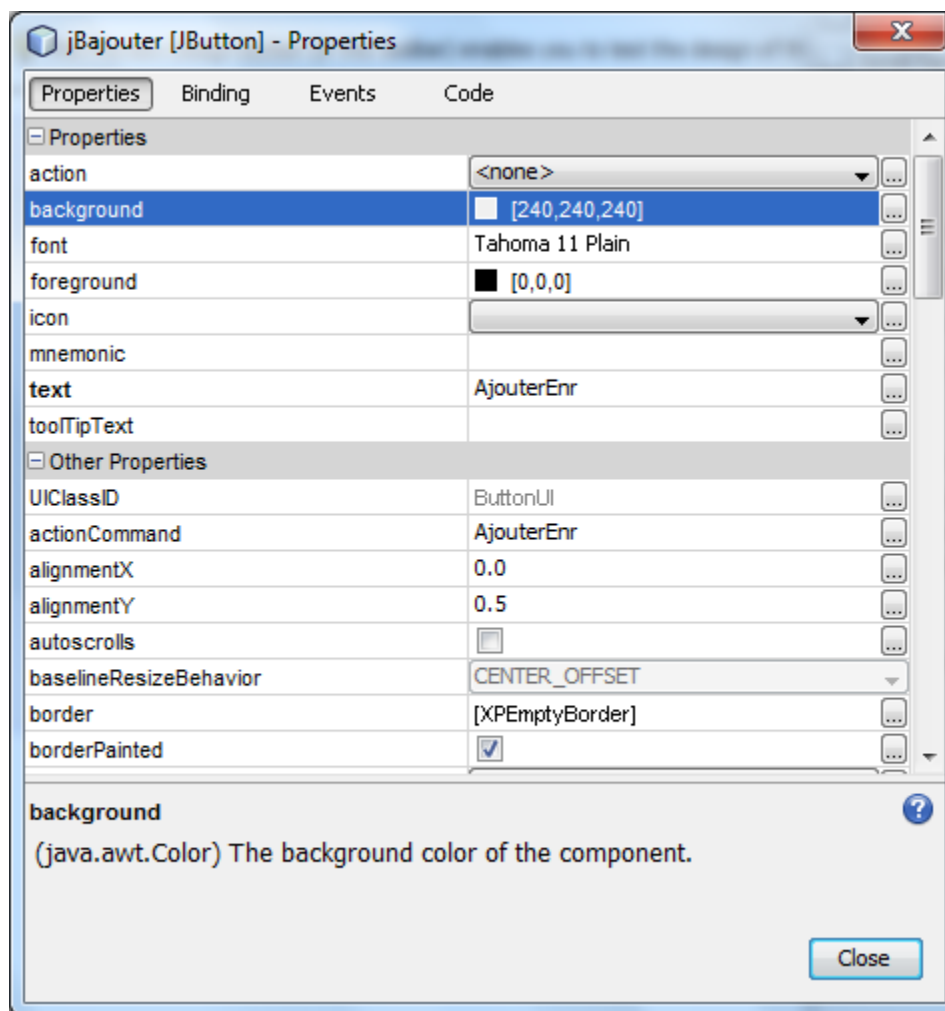


Rôle Classe JFrameGestion.

Cette classe appelée dès que la connexion est établie (par la classe identifier)

```
private void jBconnecterActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String n = jTextUser.getText();  
    String m = jTextMPasse.getText();  
    connBD.setConnection(n,m);  
    connBD.connecter();  
    // appel de la classe JFrameGestion (en utilisant un objet Connexion)  
    JFrameGestion fenetre = new JFrameGestion(connBD);  
    fenetre.setVisible(true);  
}
```

C'est au niveau de cette classe que nous avons décidé d'afficher et d'ajouter des enregistrements.



On accède aux propriétés des Contrôles graphiques par le bouton droit/propriétés (figure précédente).

Pour accéder au nom du contrôle, utiliser l'onglet **CODE**

IMPORTANT :

Dans les classes JFrame qui utilisent une connexion à la base de données qui a été définie dans une autre classe (ConnectionOrale), il est important de définir un objet (une variable) de type ConnectionOracle comme suit :

```
ConnectionOracle connBD;
```

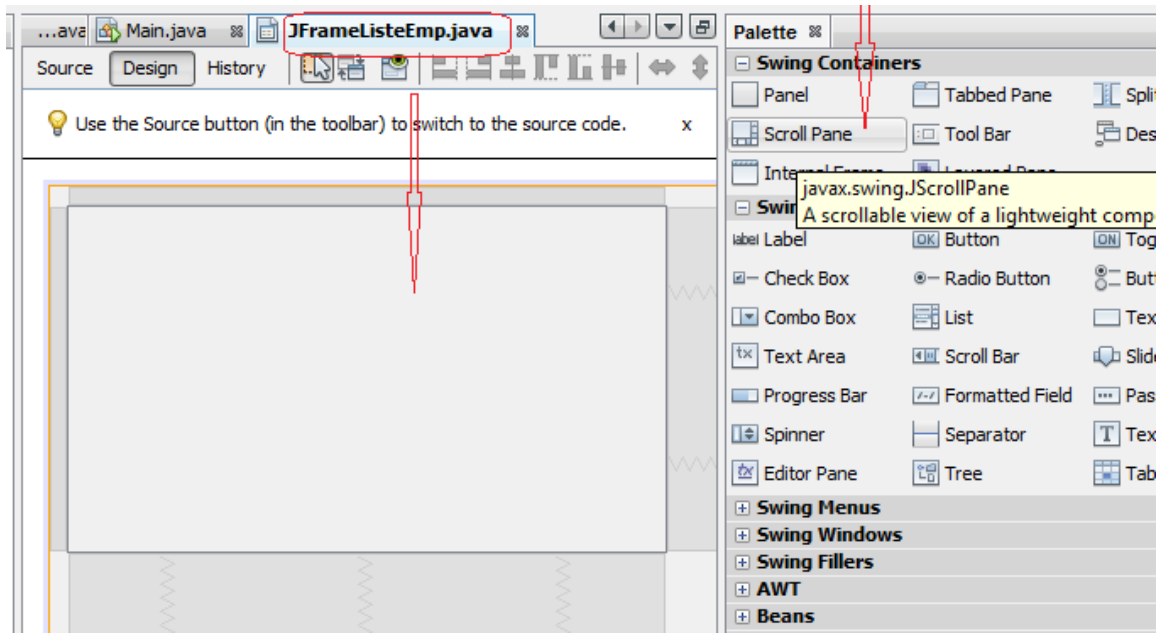
Lors de la définition d'un Statement ou d'un PreparedStatement ou un CallableStatement, on utilise l'instruction suivante

```
PreparedStatement stminsert=connBD.getConnection().prepareStatement(sqlajout);
```

Au lieu de

```
PreparedStatement stminsert=connBD.prepareStatement(sqlajout);
```

Rôle de la classe JFrameListeEmp



On utilise un Jtable (Tableau de lignes et de colonnes) pour contenir

Toutes les lignes et toutes les colonnes. un constructeur de Jtable est de la forme:

```
JTable(Object[][] rowData, Object[] columnNames)
```

On utilise le JScrollPane pour visualiser le JTable.

La méthode qui permet de faire lister tous les employés est **public void lister()**

Il ne faut pas formater le JScrollPane en mode Design. C'est le vecteur qui est défini par le code dans le Jtable qui va formater vos données. Le JScrollPane est vu comme une fenêtre de visualisation seulement.