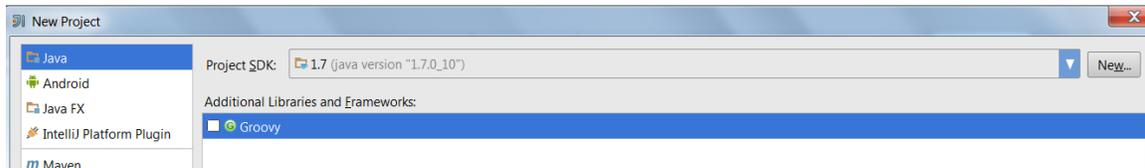


Comment démarrer un projet IntelliJ Idea utilisant une interface graphique?

IntelliJ Idea est un éditeur Java qui permet de réaliser des interfaces graphiques en utilisant les bibliothèques de swing. Voici une explication rapide de comment démarrer un projet IntelliJ Idea graphique et JDBC

Nouveau projet.(avec une version du JDK (1.7 et plus).(C:\Program File\Java\jdk1.8.0.25)

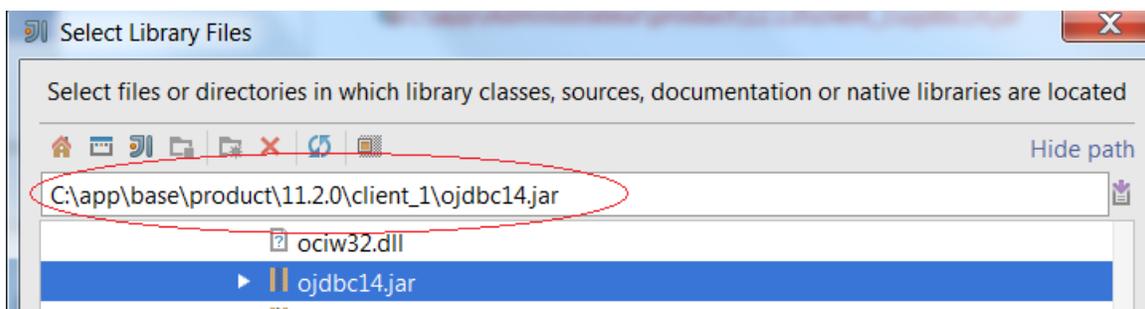
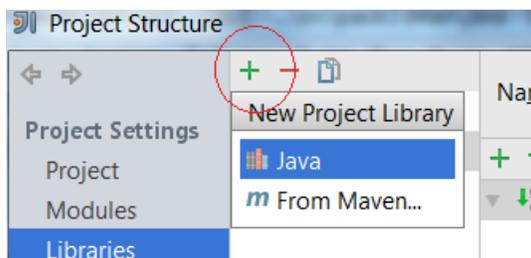


Comme c'est un projet JDBC, il faudra ajouter les classes **JAR ojdbc14.jar** qui se trouve dans le dossier C:\app\base\product\11.2.0\client_1. Procéder comme suit :

Si vous n'avez pas la dernière version, vous pouvez la télécharger à l'adresse :

<http://www.oracle.com/technetwork/apps-tech/jdbc-10201-088211.html>

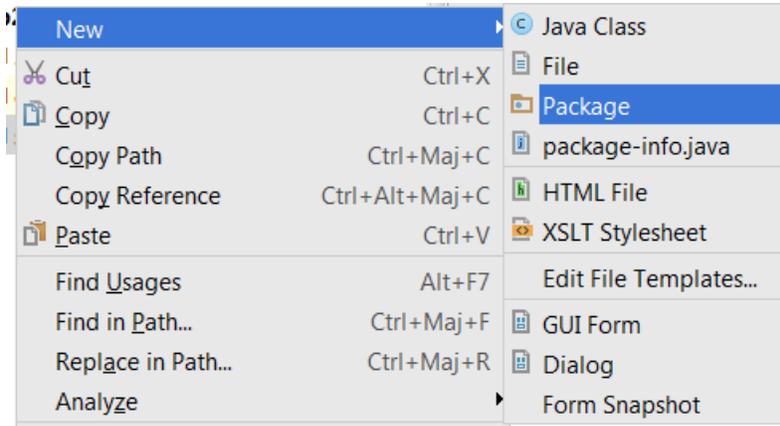
Par le menu Fichier/Project Structure/Libraire, puis ajouter.



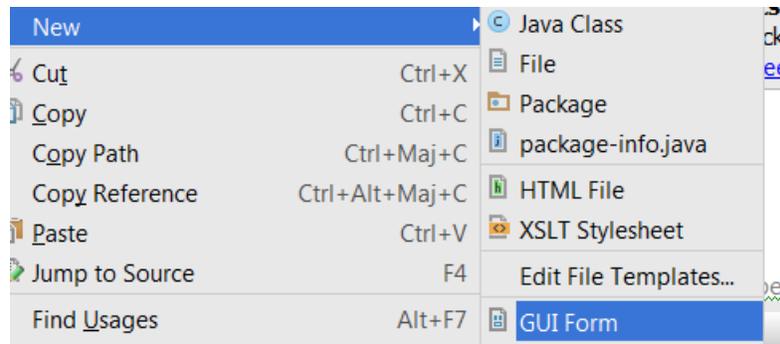
Maintenant que l'environnement de travail est bien configuré, alors on peut commencer

Il est également important de créer un **package** qui va contenir toutes les classes en rapport avec le projet.

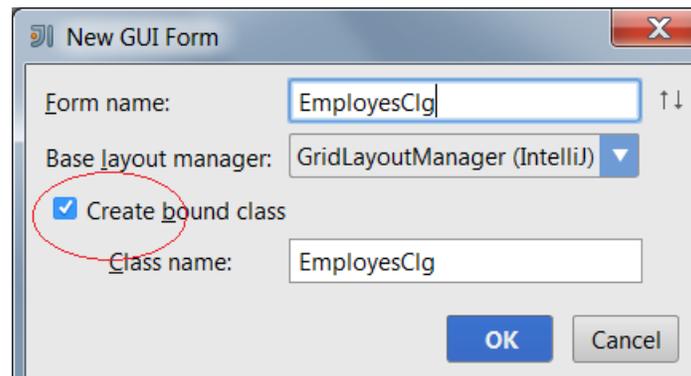
1. Bouton droit sur src, puis nouveau package. Donnez un nom significatif à votre package.



2. Puis bouton droit sur le nom **du package**, puis nouveau GUI Form



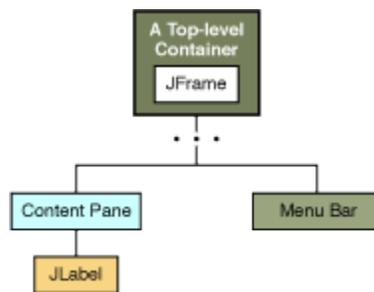
Ceci va vous permettre de créer deux classes



- La classe Java, contenant le code source de votre programme.
- Une classe de même nom avec l'extension **.form** Cette classe va contenir le code correspondant à la création et définition des contrôles swing (bouton, zone de text ..) associés à votre application.

Après la création de la GUI form, vous obtiendrez l'espace de travail suivant :

- ✓ Votre espace de travail est le Jpanel dont le nom est panel1.
- ✓ C'est ce conteneur qui va contenir l'ensemble des autres contrôles : JButton, JTextField, JLabel etc ...). Par la suite, votre JFrame, doit contenir votre Jpanel.



- ✓ Les contrôles swing ou la boîte à outil est complètement à droite.
- ✓ Complètement à gauche vous remarquerez que vous avez deux classes : EmployesClg et employesClg.form.

Il est inutile de double-cliquer sur le bouton «premier» pour aller chercher le code

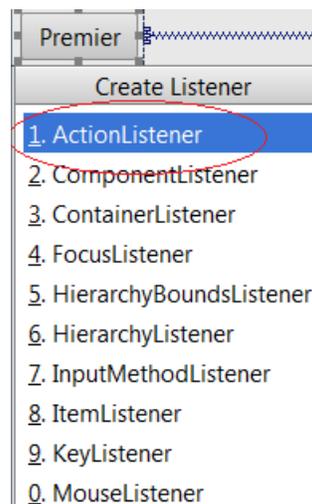
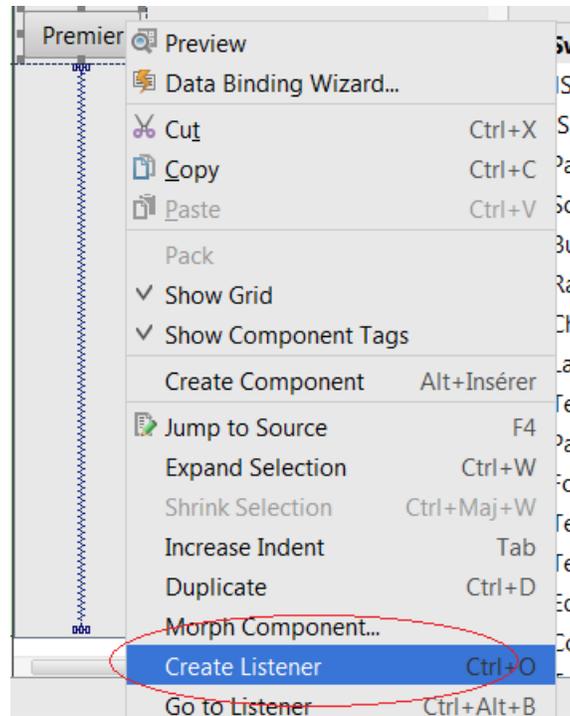
The screenshot shows the IntelliJ IDEA GUI Designer interface. On the left, the Project Structure view shows the package hierarchy: `GestionEmployesClg` containing `EmployesClg` and `EmployesClg.form`. The Component Tree in the center shows a `Form (GestionEmployesClg.EmployesClg)` containing `panel1 : JPanel` and `premier : JButton`. A red circle highlights `panel1 : JPanel`. Below the component tree is a Properties table:

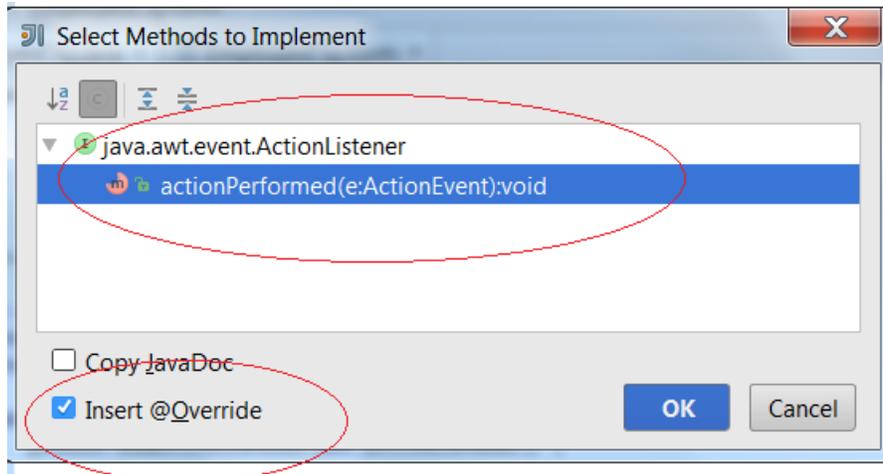
Property	Value
field name	panel1
Custom Create	<input type="checkbox"/>
Layout Manager	GridLayoutManager ...
border	None
margins	[0, 0, 0, 0]
Horizontal Gap	-1

On the right, the Palette shows various Swing components. A red circle highlights `JPanel`. A blue arrow points from the `premier` button in the design view to the `JPanel` in the palette. A box labeled 'Jpanel' with an arrow points to the `panel1` component in the component tree.

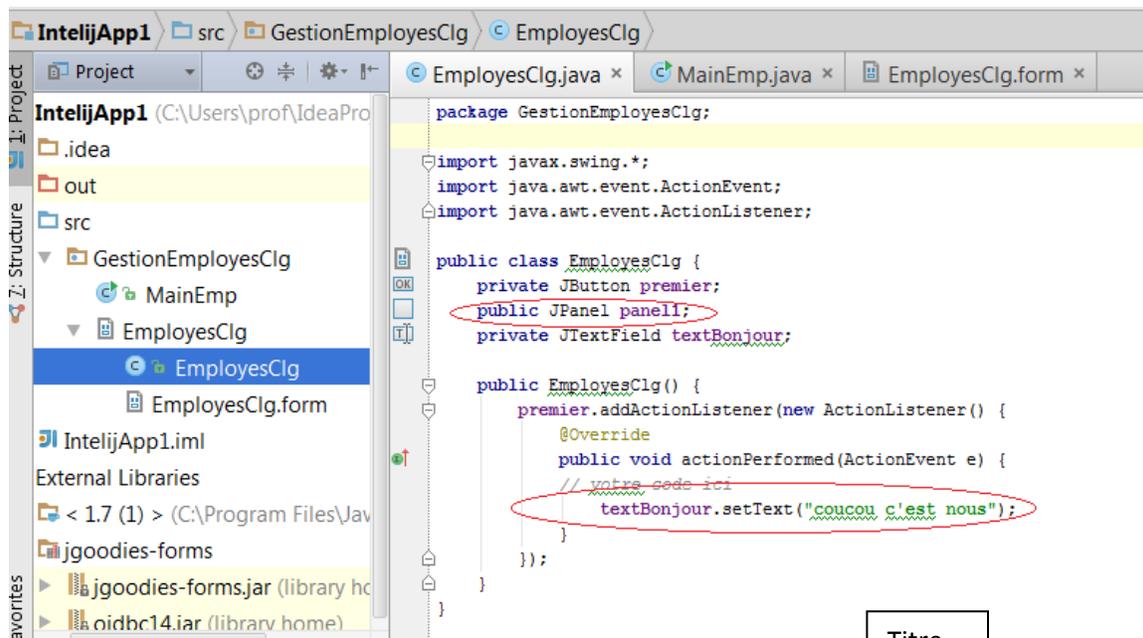
Quand on parle du développement graphique avec Java swing , on parle de Listener La méthode associer est ActionListener(). Pour accéder à cette méthode :

- Create Listener pour créer le code pour la première fois
- Go To Listener pour modifier le code déjà créé.



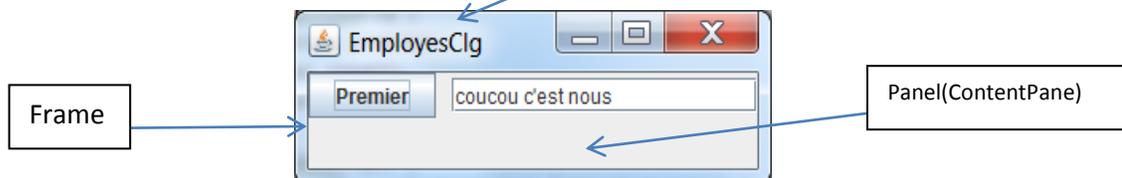


Voici ce que donne l'exemple dans la classe correspondant au code du Form

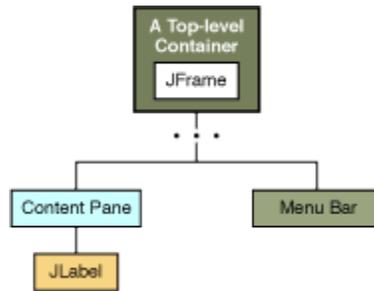


Titre

L'exécution donne ceci :

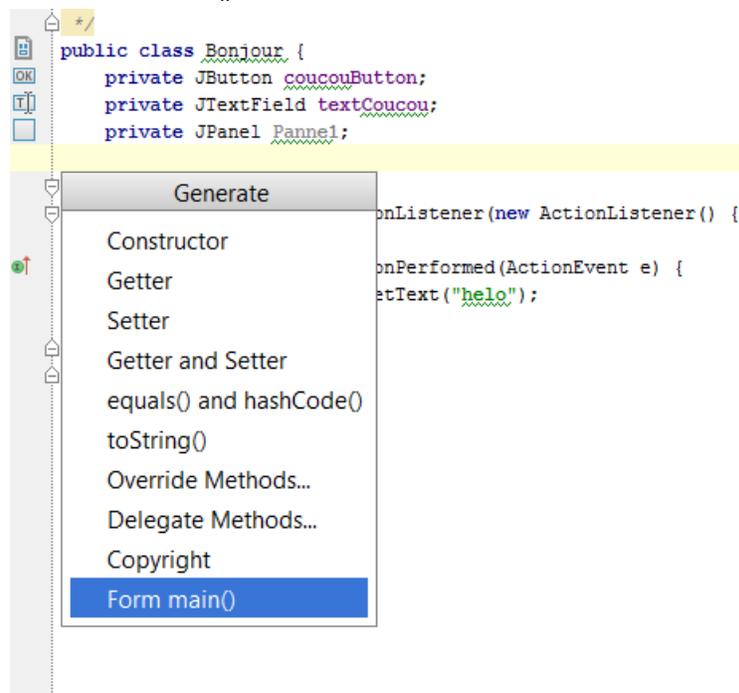


Panel(ContentPane) est contenu dans un JFrame.



Pour pouvoir exécuter le programme, il faudra lui inclure une classe main, deux options s'offrent à vous :

- 1- inclure une classe main dans le code source :
 - a. Ouvrir la class du code source (bound class pour édition – modification)
 - b. Presser ALT + Insert
 - c. choisir Formmain()



Le code suivant va alors s'insérer :

```
public static void main(String[] args) {
    // on crée une fenêtre dont le titre est "Bonjour"
    JFrame frame = new JFrame("Bonjour");
    // on ajoute le contenu du Pannel
    frame.setContentPane(new Bonjour().Pannel);
    //la fenêtre se ferme quand on clique sur la croix rouge
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //on attribue la taille minimale au frame
    frame.pack();
    // on rend le frame visible
    frame.setVisible(true);
}
```

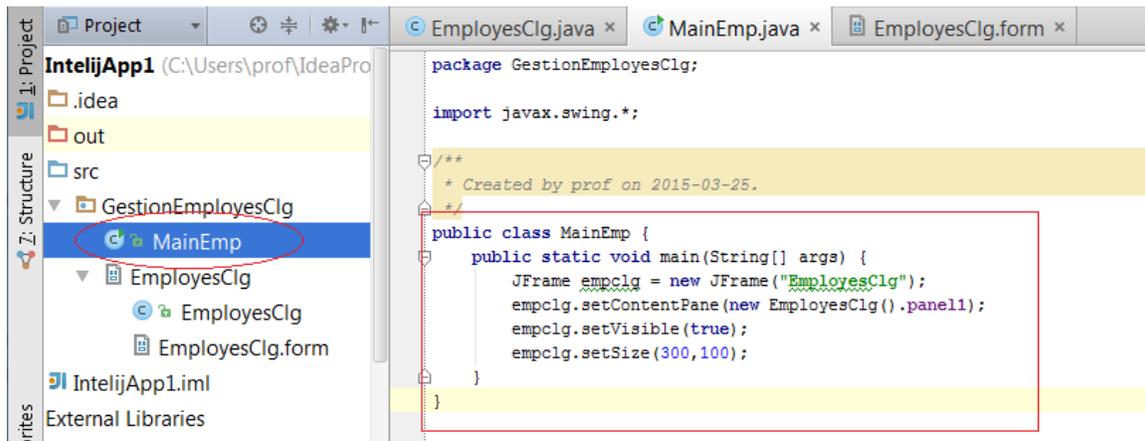
Pour le `frame.pack()`, il est préférable de le remplacer par

`frame.setSize (int widht, int height)`

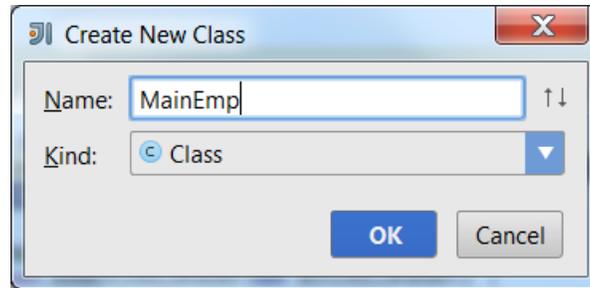
2- créer votre propre classe main, c'est ce qui est recommandé en suivant les étapes :
définir un Frame Java, dont le nom est `empclg.` et dont le titre est `EmployesClg`

1. affecter les propriétés du `JPanel panel1` au `JFrame`
2. affiche le `JFrame`
3. affecte une dimension au frame avec la methode **`setSize (int widht, int height)`**

(Voir exemple)

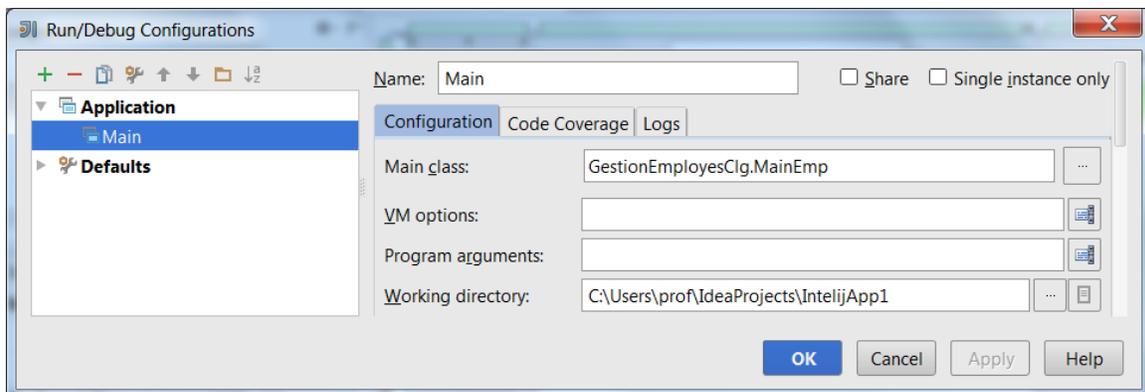


Vous pouvez créer votre classe `MainEmp` qui va contenir la classe main, comme suit :



Autre chose à configurer : Quelle est la classe qui va contenir la class main pour l'exécution du programme.

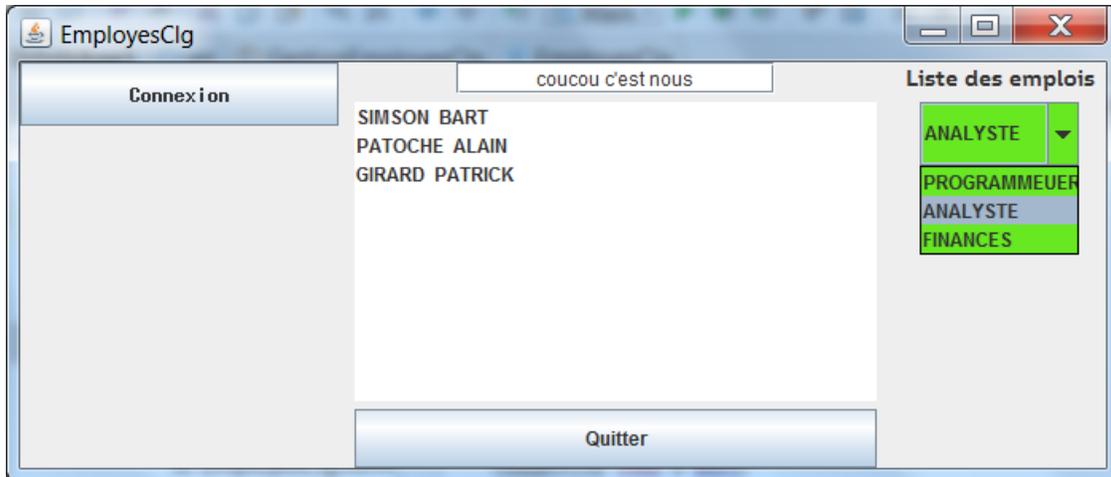
Par le menu Execution (Run), choisir Edit Configurations et choisir la classe qui va contenir votre classe **main**.



Exemple :

Pour obtenir l'interface graphique suivante, nous avons eu besoin des classes suivantes dans le package **GestionEmployesClg**

- 1- Une classe EmployesClg de type GUI Form, qui correspond aux classes (EmployesClg et EmployesClg,form)
- 2- Une classe MainEmp qui contient la classe main



Contenu de la classe MainEmp

```
package GestionEmployesClg;
import javax.swing.*;

public class MainEmp {
    public static void main(String[] args) {
        JFrame empclg = new JFrame("EmployesClg");
        empclg.setContentPane(new EmployesClg().panell1);
        empclg.setVisible(true);
        empclg.setSize(700, 300);
    }
}
```

Contenu de la classe EmployesClg.form : Contien tous les controles (Jpanel, JTextField, JButton etc ..)de notre application

La classe EmployesClg contient :

```
package GestionEmployesClg;
import oracle.jdbc.pool.*;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class EmployesClg {
    // Declaration des controles de la forme
    private JButton premier;
    public JPanel panel1;
    private JTextField textBonjour ;
    private JButton quitter;
    private JList list1;
    private JComboBox comboBox1;

    // Declaration des requetes SQL, resultSet et autres
    Connection conn = null;
    String user="user1";
    String mpasse ="user1";
    String url = "jdbc:oracle:thin:@205.237.244.251:1521:orcl";
    String sql1 ="SELECT nomemp, prenomemp FROM employesbidon where
    emploi = ?";
    String sqlemp ="select distinct emploi from employesbidon";
    ResultSet rst;

    // Fonction qui liste les emplois et les mets dans le ComboBox
    public void listeemploi()
    {
        try{
            Statement stm1 = conn.createStatement();
            ResultSet rst2 = stm1.executeQuery(sqlemp);
            while (rst2.next())
            {
                comboBox1.addItem(rst2.getString(1));
            }
            rst2.close();
        }
        catch (SQLException sql)
        { System.out.println(sql); }
    }

    public EmployesClg() {
        // Fonction du bouton de connexion
    }
}
```

```

premier.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // votre code ici
        // Afficher un message dans la zone de texte
        textBonjour.setText("coucou c'est nous");
        try {
            OracleDataSource ods = new OracleDataSource();
            ods.setURL(url);
            ods.setUser(user);
            ods.setPassword(mpasse);
            conn= ods.getConnection();
            System.out.println("vous etes connectés ");
            // appel de la fonction listeemploi
            listeemploi();
        }
        catch(SQLException exconn)
        {
            System.out.println(exconn);
        }
    }
});

// Fonction qui liste les employés en fonction d'une entrée du
//ComboBox.
comboBox1.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        try {
            PreparedStatement stm = conn.prepareStatement(sql1);
            stm.setString(1, comboBox1.getSelectedItem().toString());
            rst = stm.executeQuery();
            DefaultListModel listModel = new DefaultListModel();
            while (rst.next())
            {
                listModel.addElement(rst.getString(1) + " " +
                rst.getString(2));
            }
            list1.setModel(listModel);
            rst.close();
        }

        catch(SQLException sqllex)
        {
            System.out.println(sqllex);
        }
    }
});

// code du bouton quitter

```

```
    quitter.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {

            try
            {
                conn.close();
                System.out.println("connexion fermée");
            }

            catch(SQLException se)
            {
                conn = null;
            }
            System.exit(0);
        }
    });

// fin de public EmployesClg()
}
// fin du public class EmployesClg
}
```

420-KEH-LG, Travail No3 — pondération 10% (JDBC)

L'entreprise InfoClg possède une vaste bibliothèque qu'elle met à la disposition de ses employés. Les ouvrages de cette bibliothèque sont classés par genre (Informatique, sciences, divertissement, droit commercial, histoire, littérature ...). Notez que l'on peut ajouter un genre.

Nous souhaitons développer une application graphique avec **intellij Idea** permettant d'informatiser une partie de la bibliothèque pour faciliter les tâches suivantes :

- 1- ajouter, modifier et supprimer un adhérent. (Un employé doit s'inscrire à la bibliothèque pour pouvoir emprunter)
- 2- consulter la liste des livres par genre. La consultation doit se faire un livre à la fois et on doit se déplacer d'un enregistrement à un autre.
- 3- saisir un prêt.(ajouter un prêt).
- 4- consulter la liste des livres en cours de prêts (titre du livre, le genre, la date du prêt, la date prévue pour le retour, le nom et le prénom de l'adhérent..)
- 5- la recherche d'un livre par auteur ou par titre (début d'un titre).
- 6- la liste des livres les plus empruntés.

Indications :

- 1- Un livre a plusieurs exemplaires et un exemplaire appartient à un seul livre.
- 2- un livre a un numéro, un titre, un auteur, un genre, une date de parution et la maison d'édition.
- 3- un adhérent est un employé de la compagnie. Il a un numéro, un nom, un prénom, adresse, numéro de téléphone etc. Le numéro doit être géré par un Trigger.
- 4- un prêt a une date début et une date prévue pour le retour.

La remise et la correction du travail a lieu le jeudi 23 avril. Aucun délai supplémentaire ne sera accordé. Le travail peut être fait en équipe de deux, mais pas plus.

L'utilisation du PreparedStatement ou du CallableStatement est fortement recommandé.