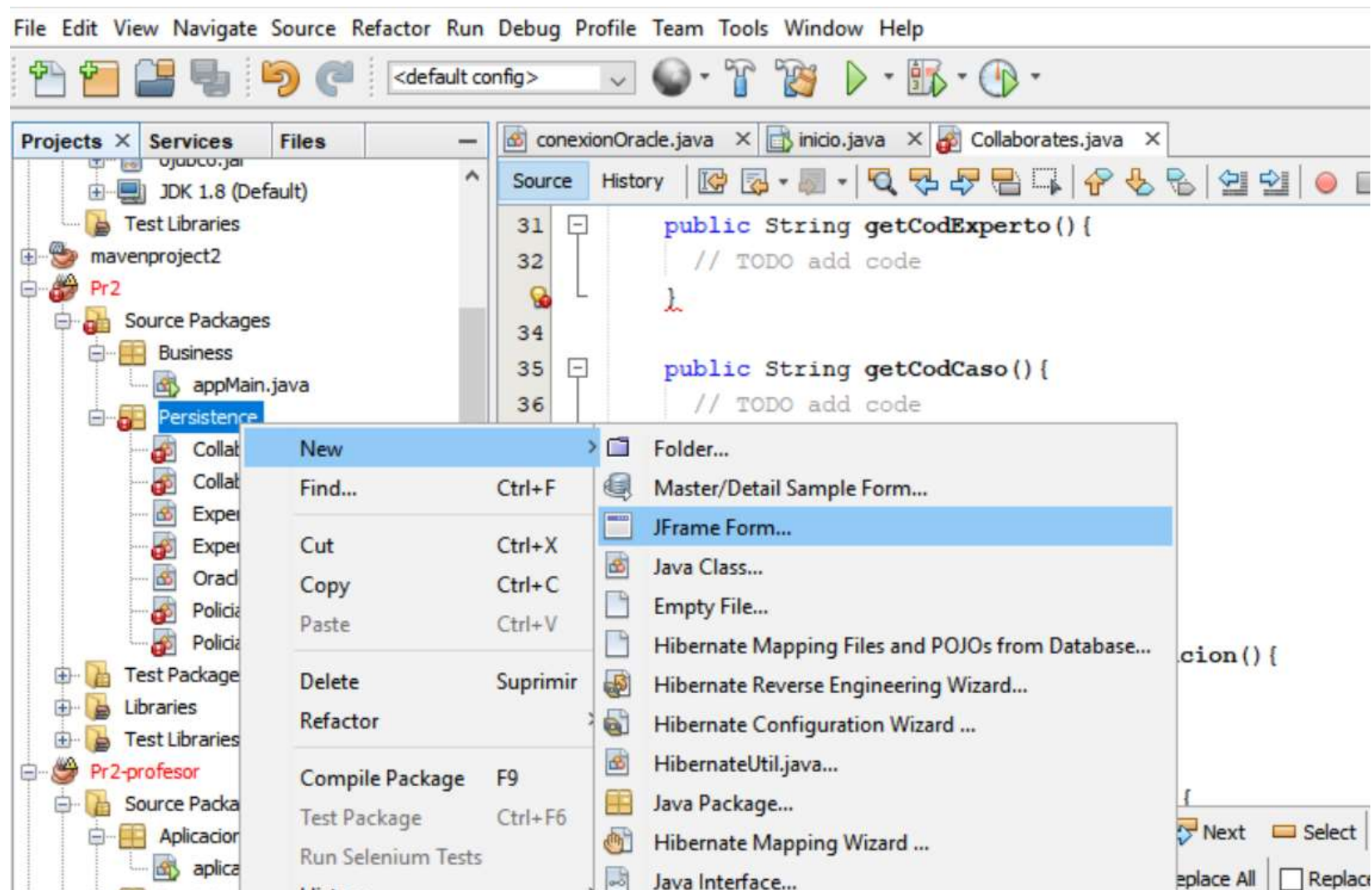# Swing Library

- The objective of these slides is to show, in a brief and concise way, how to create a Java application with a graphical user interface using the Swing library, which is already incorporated in NetBeans

- Of course, there are many libraries to design interfaces (for example, JavaFX) but, in this case, we will focus on this library because it is sufficiently stable and very widespread

- This is not a Swing manual. They are only very basic notions to interact, from Java and through a graphical interface, with a database

- In addition, we will do it in "design" mode, which means that a large part of the code will be generated automatically

Projects × **Services** **Files** —

conexionOracle.java ×   inicio.java ×   Collaborates.java ×

ujubco.jar
JDK 1.8 (Default)
Test Libraries
mavenproject2
Pr2
  Source Packages
    Business
      appMain.java
    Persistence
      Collab
      Collab
      Exper
      Exper
      Orad
      Polici
      Polici
  Test Package
  Libraries
  Test Libraries
Pr2-profesor
  Source Packa
    Aplicacior
      aplica

```
Source  History

31  ⊟        public String getCodExperto(){
32              // TODO add code
              }
34
35  ⊟        public String getCodCaso(){
36              // TODO add code
```

New		>	Folder...
Find...	Ctrl+F		Master/Detail Sample Form...
			JFrame Form...
Cut	Ctrl+X		Java Class...
Copy	Ctrl+C		Empty File...
Paste	Ctrl+V		Hibernate Mapping Files and POJOs from Database...
Delete	Suprimir		Hibernate Reverse Engineering Wizard...
Refactor			Hibernate Configuration Wizard ...
			HibernateUtil.java...
Compile Package	F9		Java Package...
Test Package	Ctrl+F6		Hibernate Mapping Wizard ...
Run Selenium Tests			Java Interface...

...cion(){

Next   Select

eplace All   Replace

Source | Design | History

To add a component multiple times, select it via click in palette and then Shift-click on design canvas.

Palette ☓

▭ **Swing Containers**

| ▢ Panel | ▢ Tabbed Pane | ▯ Split Pane | ▣ Scroll Pane |
| ▢ Tool Bar | ▤ Desktop Pane | ▢ Internal Frame | ▦ Layered Pane |

▭ **Swing Controls**

label Label	OK Button	ON Toggle Button	☑─ Check Box
◉─ Radio Button	◉ Button Group	▣ Combo Box	▤ List
▭ Text Field	tx Text Area	◄▮▶ Scroll Bar	◖▸ Slider
▭ Progress Bar	/-/ Formatted Field	... Password Field	▯▢ Spinner
─ Separator	T Text Pane	℔ Editor Pane	▤ Tree
▦ Table

▭ **Swing Menus**

| File Menu Bar | ▤ Menu | ▬ Menu Item |
| ☑ Menu Item / CheckBox | ◉─ Menu Item / RadioButton | ▤ Popup Menu |
| ─ Separator |

▭ **Swing Windows**

| ▣ Dialog | ▤ Frame | ▨ Color Chooser | ▤ File Chooser |

[JFrame] - Properties ☓

Properties | Binding | Events | Code

JFrame

JManuBar

JMenuItem

JDesktopPane

Master/Detail Sample form
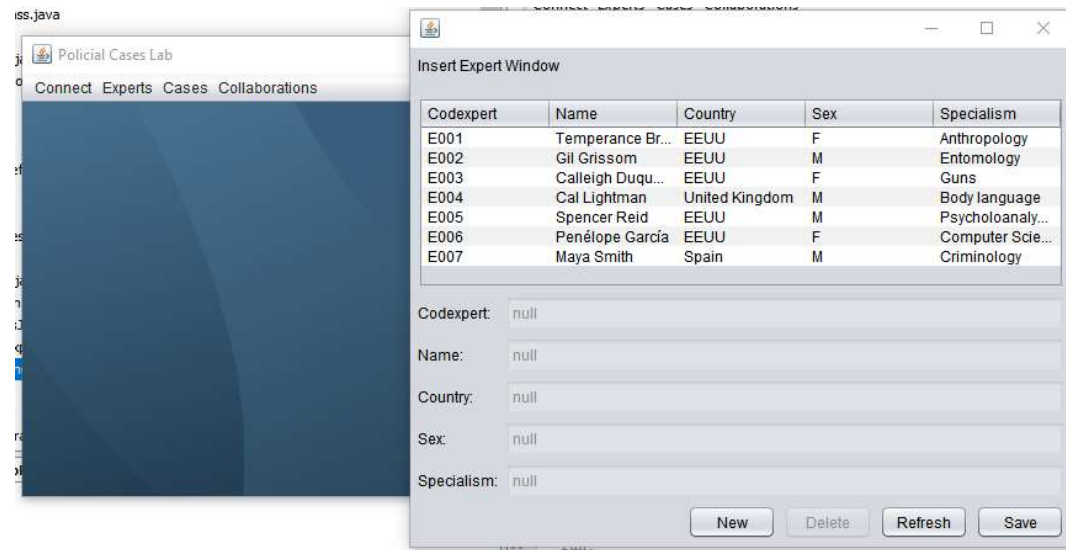
JOptionPane
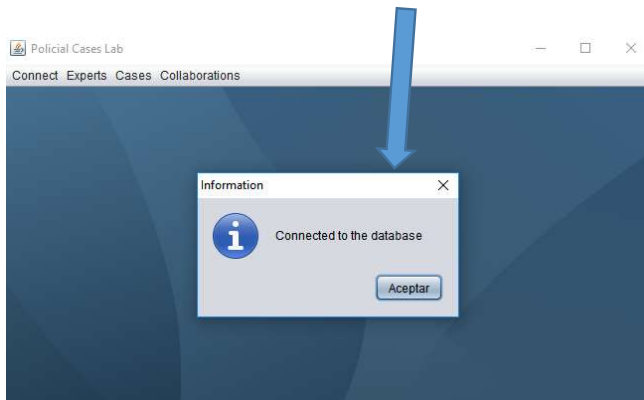
JInternalFrame
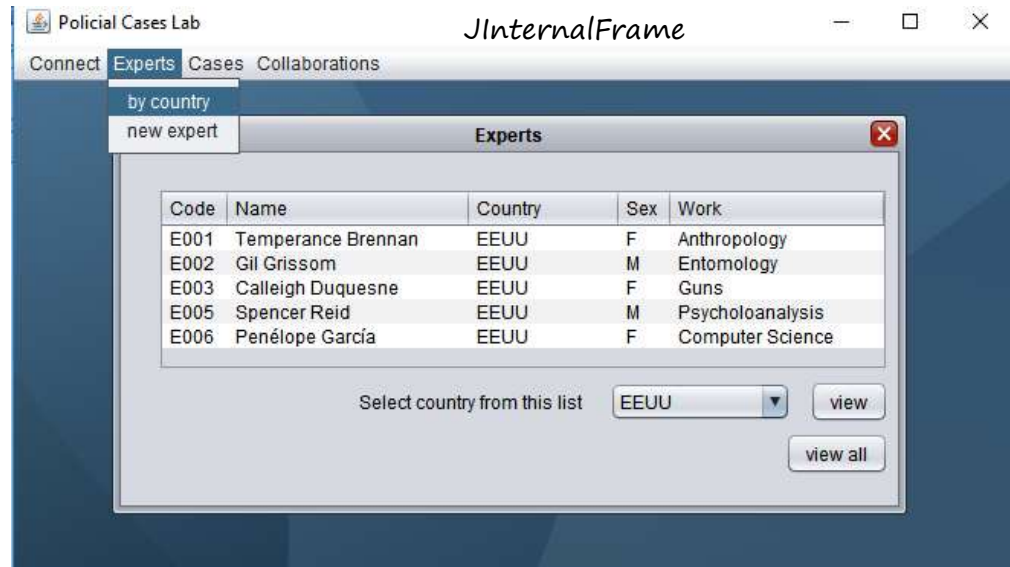
Creating menu

Connect
- Connect
- Disconnect

Experts
- By Country
- New Expert

Cases
- New Case

Policial Cases Lab

Connect Experts Cases Collaborations

Connect
Disconnect

```
 96          }
 97
 98    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
 99        try{
100            co = new conexionOracle();
101            JOptionPane.showMessageDialog(null, "Connected to database" );
102            this.setVisible(false);
103            this.dispose();
104        } catch (Exception ex) {
105            JOptionPane.showMessageDialog(null, ex.getStackTrace() );
106        }
107
108
109
110    }
```

```
public class mainWindow extends javax.swing.JFrame {
    OracleConnection co =null;
/**
 * Creates new form mainWindow
 */
    public mainWindow() {
        this.setTitle("Policial Cases Lab");
        initComponents();

    }
```

Policial Cases Lab

Connect Experts Cases Collaborations
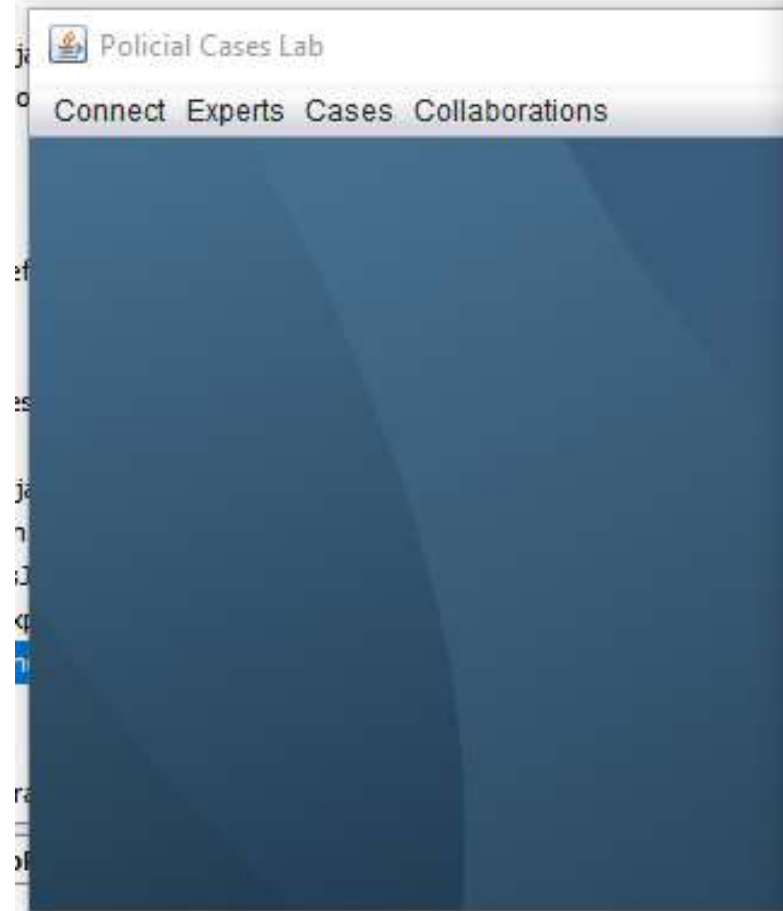
Insert Expert Window

Codexpert	Name	Country	Sex	Specialism
E001	Temperance Br...	EEUU	F	Anthropology
E002	Gil Grissom	EEUU	M	Entomology
E003	Calleigh Duqu...	EEUU	F	Guns
E004	Cal Lightman	United Kingdom	M	Body language
E005	Spencer Reid	EEUU	M	Psycholoanaly...
E006	Penélope García	EEUU	F	Computer Scie...
E007	Maya Smith	Spain	M	Criminology
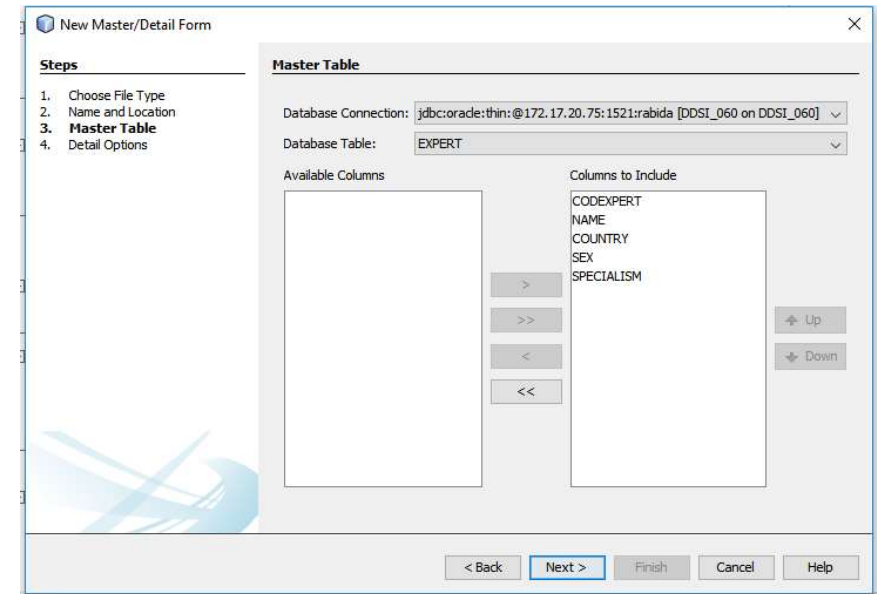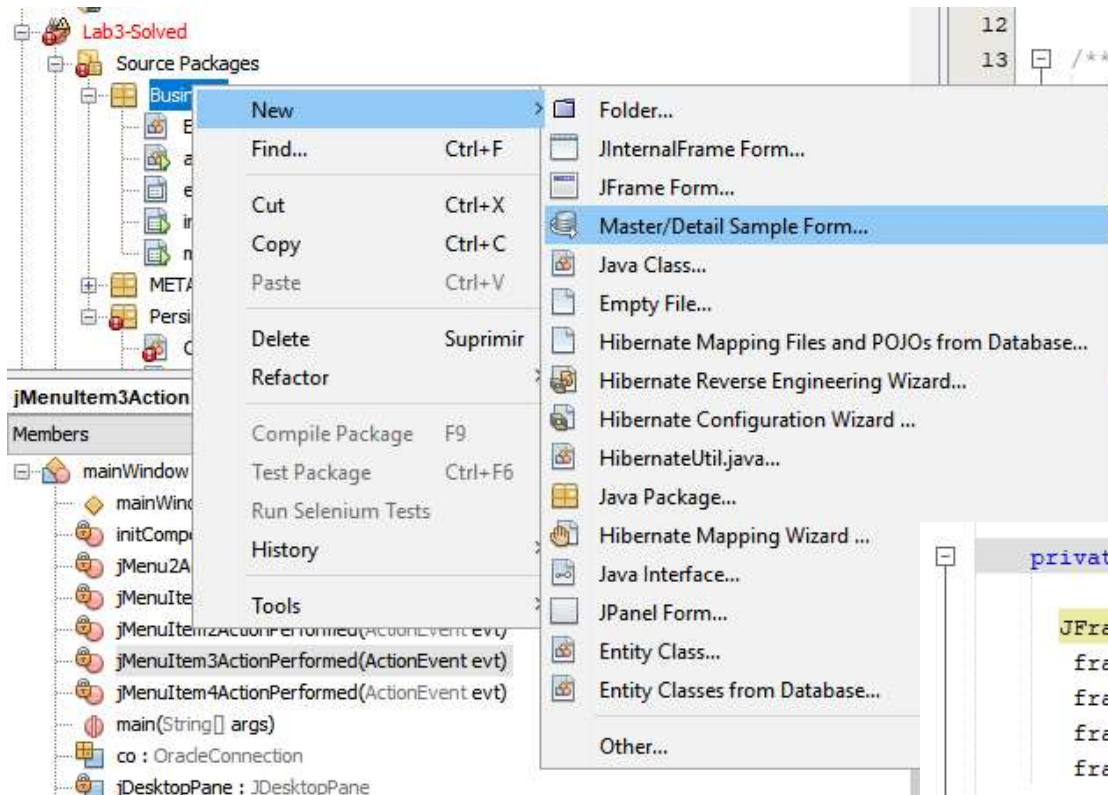
Codexpert: null

Name: null

Country: null

Sex: null

Specialism: null

New   Delete   Refresh   Save

Master / Detail Sample Form

Policial Cases Lab — □ X

Connect **Experts** Cases Collaborations

by country

new expert

Experts ⊠

Code	Name	Country	Sex	Work
E001	Temperance Brennan	EEUU	F	Anthropology
E002	Gil Grissom	EEUU	M	Entomology
E003	Calleigh Duquesne	EEUU	F	Guns
E005	Spencer Reid	EEUU	M	Psycholoanalysis
E006	Penélope García	EEUU	F	Computer Science

Select country from this list EEUU ▼   view

view all

JInternalFrame Form

jTable –DefaultTableModel

Code	Name	Country	Sex	Work
E001	Temperance Brennan	EEUU	F	Anthropology
E002	Gil Grissom	EEUU	M	Entomology
E003	Calleigh Duquesne	EEUU	F	Guns
E005	Spencer Reid	EEUU	M	Psycholoanalysis
E006	Penélope García	EEUU	F	Computer Science

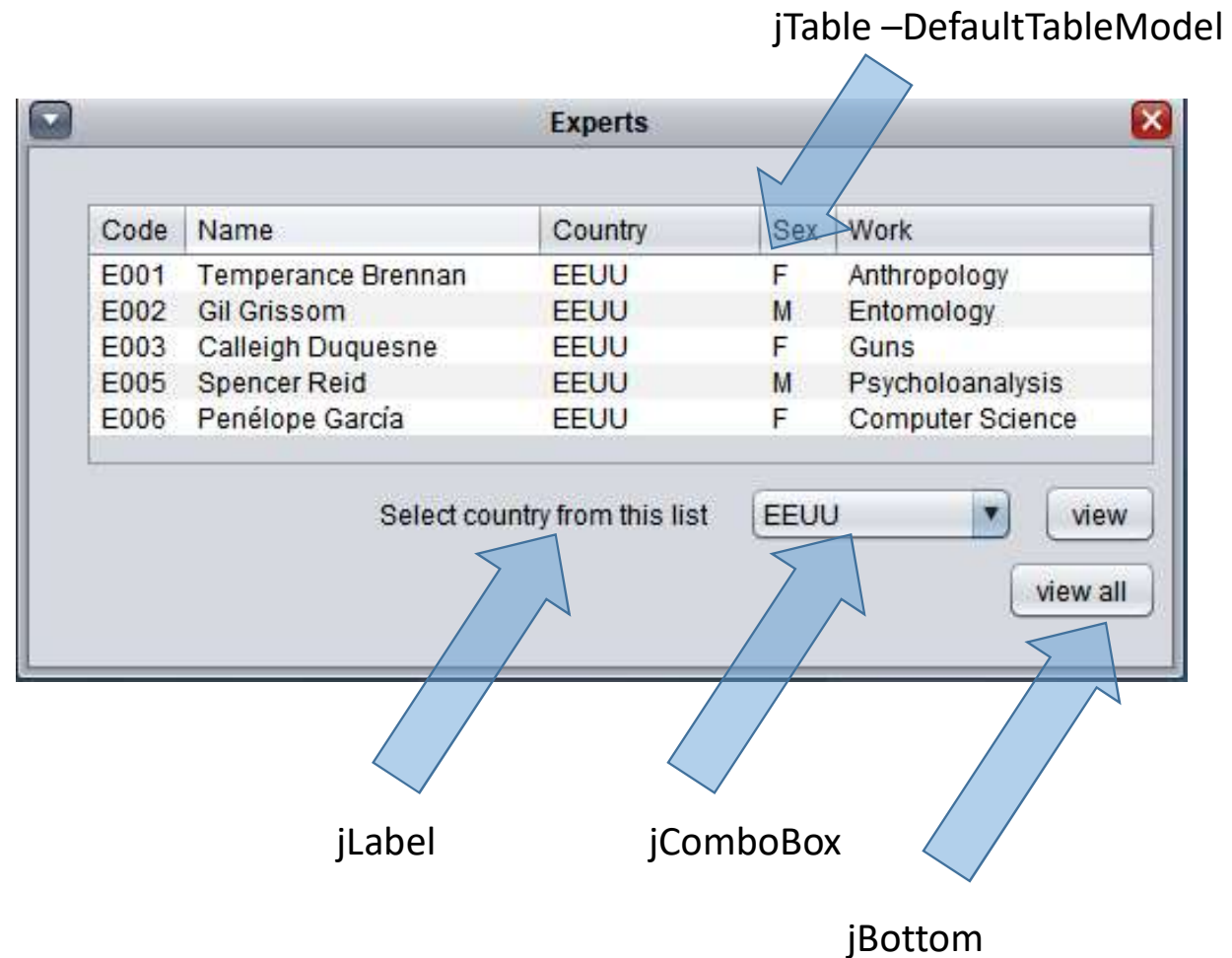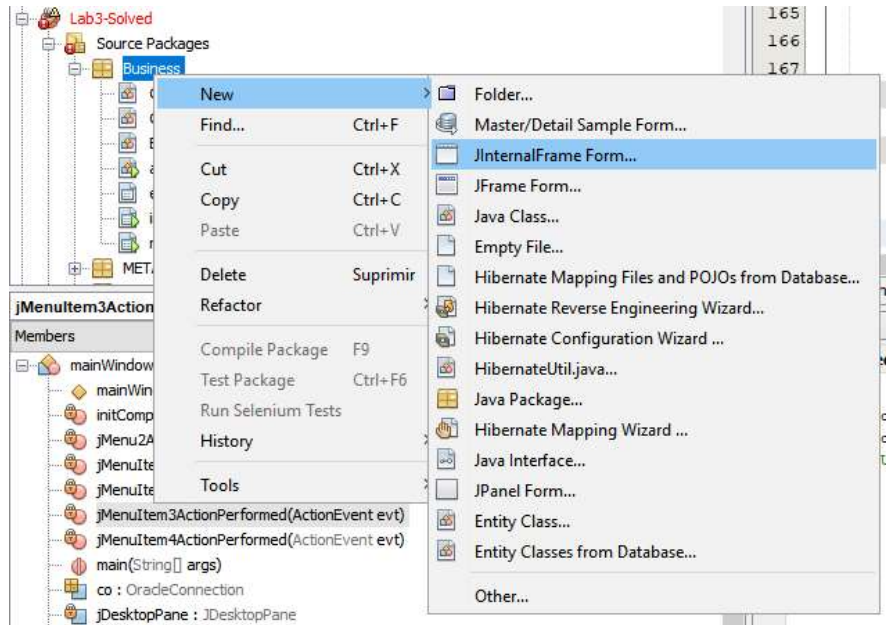Select country from this list   EEUU   view

view all

jLabel   jComboBox

jBottom

jTable and DefaultTableModel

- DefaultTableModel - As the name suggests it is the table model that is used by a JTable when no table model is specifically defined by the programmer.

- It is used to define how this table is going to be (for example column names)

```java
public class expertsJInternalFrame extends javax.swing.JInternalFrame {
    ExpertManager mExp = null;
    DefaultTableModel lookTExperts = new DefaultTableModel() {
        @Override
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    /**
     * Creates new form expertsJInternalFrame
     */
    public expertsJInternalFrame(OracleConnection c) {
        mExp = new ExpertManager(c);
        initComponents();
        this.setTitle("Experts");
        DrawingExpertsTable(); // Columns names
        try {
            viewExperts();
        }
        catch (SQLException e){
            JOptionPane.showMessageDialog(null, e.getMessage() );
        }
    }
}
```

```java
public class expertsJInternalFrame extends javax.swing.JInternalFrame {
    ExpertManager mExp = null;
    DefaultTableModel lookTExperts = new DefaultTableModel() {
            @Override
            public boolean isCellEditable(int row, int column) {
                    return false;
            }
    };
    /**
     * Creates new form expertsJInternalFrame
     */
    public expertsJInternalFrame(OracleConnection c) {
        mExp = new ExpertManager(c);
        initComponents();
        this.setTitle("Experts");
        DrawingExpertsTable(); // Columns names
        try {
            viewExperts();
        }
        catch (SQLException e){
                JOptionPane.showMessageDialog(null, e.getMessage() );
        }
    }
}
```

```java
private void DrawingExpertsTable() {
    tExperts.setModel(lookTExperts);
    String[] columns = {"Code","Name","Country","Sex","Work"};
    lookTExperts.setColumnIdentifiers(columns);

    // no re-dimension with the mouse is allowed
    tExperts.getTableHeader().setResizingAllowed(false);

    // fixing column width
    tExperts.getColumnModel().getColumn(0).setPreferredWidth(25);
    tExperts.getColumnModel().getColumn(1).setPreferredWidth(140);
    tExperts.getColumnModel().getColumn(2).setPreferredWidth(80);
    tExperts.getColumnModel().getColumn(3).setPreferredWidth(8);
    tExperts.getColumnModel().getColumn(4).setPreferredWidth(122);
}
```

```java
private void viewExperts() throws SQLException {
    ArrayList<Expert> lExp = mExp.expertList(); // list all the experts
    fillExpertsTable (lExp);
}
```

```java
private void fillExpertsTable (ArrayList<Expert> experts) {
    Object[] columna = new Object[5];
    int numRegistros = experts.size();
    for (int i = 0; i < numRegistros; i++)  {
        columna[0] = experts.get(i).getCodEXPERT();
        columna[1] = experts.get(i).getName();
        columna[2] = experts.get(i).getCountry();
        columna[3] = experts.get(i).getSex();
        columna[4] = experts.get(i).getSpecialism();
        lookTExperts.addRow(columna);
    }
}
```

```java
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    removeRowsTExperts();
    try {
        viewExpertsByCountry(jComboBox1.getSelectedItem().toString());
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, ex.getMessage());
    }

}
```

```java
private void viewExpertsByCountry(String country) throws SQLException {
    ArrayList<Expert> lExp = mExp.expertListByCountry(country); // list all the experts by country
    fillExpertsTable (lExp);
}
```

```java
private void removeRowsTExperts(){
    int i = lookTExperts.getRowCount();
    while (lookTExperts.getRowCount() > 0)
        lookTExperts.removeRow(0);
}
```