



## LẬP TRÌNH JAVA 3

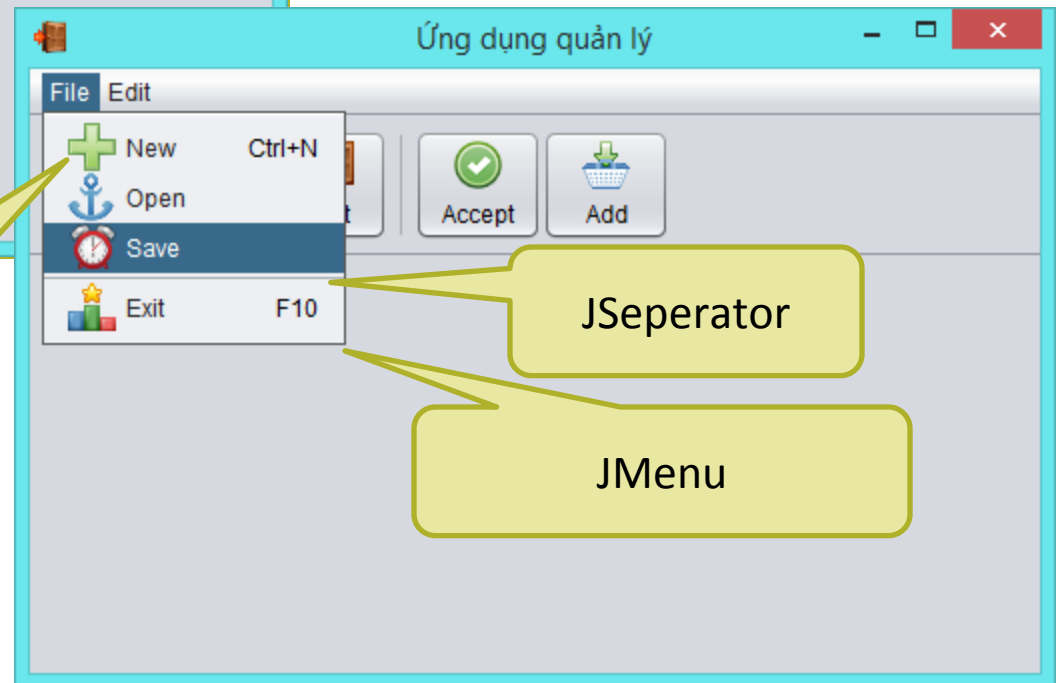
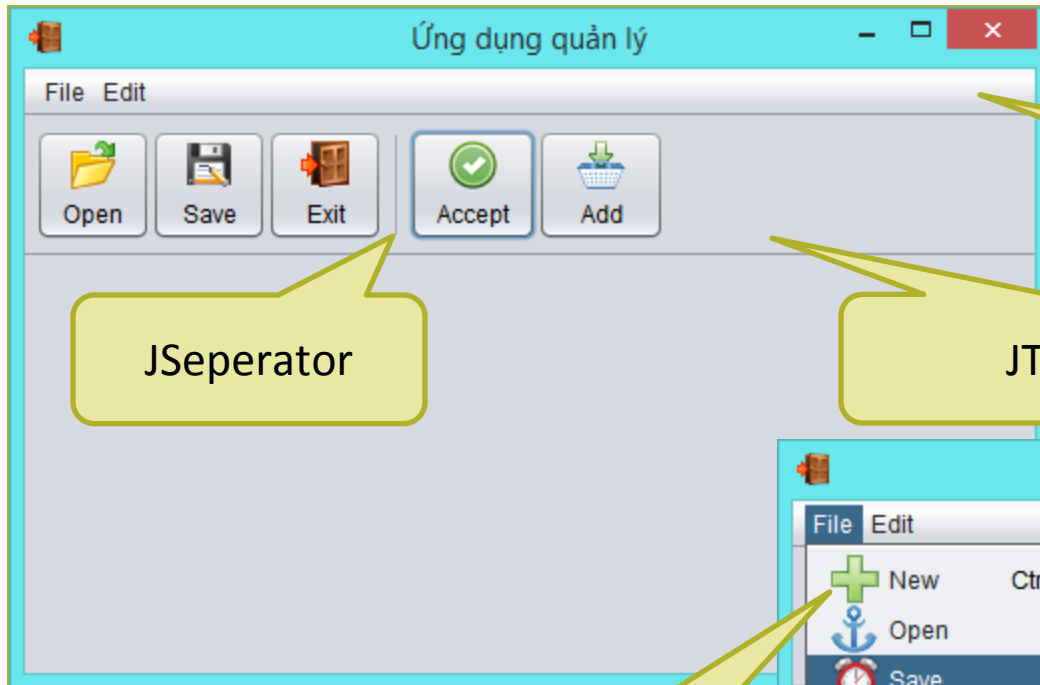
**BÀI 4: JMENU, JLIST, JCOMBOBOX,  
JTABLE, JTREE**

**PHẦN 1**

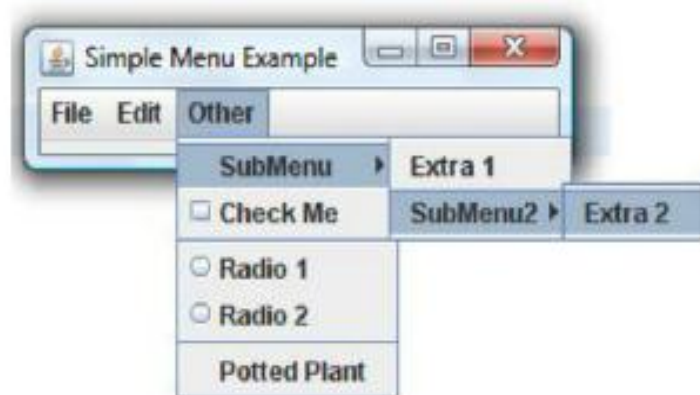
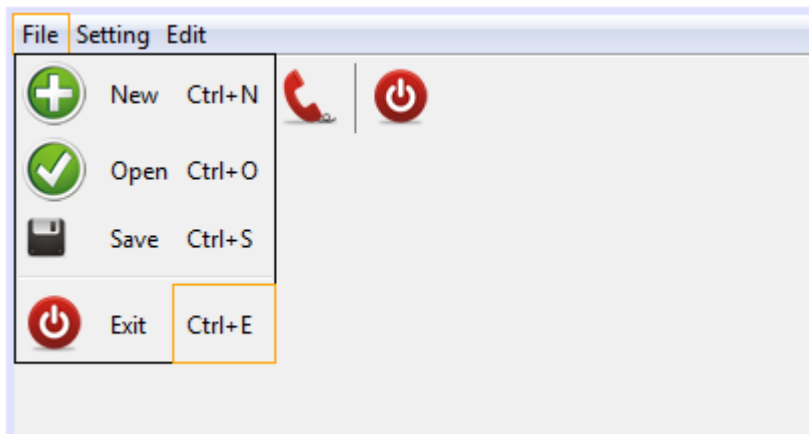
🎯 Kết thúc bài học này bạn có khả năng

- ❖ JMenu
- ❖ JMenuBar
- ❖ JMenuItem
- ❖ JPopupMenu
- ❖ JToolBar
- ❖ JList
- ❖ JComboBox
- ❖ JTable
- ❖ JTree





- ❑ JMenu có thể được dùng giống như một layout, để quản lý các Component
- ❑ Chỉ được phép có 1 lựa chọn tại 1 thời điểm
- ❑ Icon có thể dùng để thay thế cho các menu items
- ❑ Hầu hết các component chuẩn đều có thể là Menu Item (radio button...)
- ❑ Có thể gán phím tắt cho các Menu Item



## ❑ Khởi tạo

- ❖ JMenu()
- ❖ Khởi tạo một menu mới không có tiêu đề
- ❖ JMenu(Action a)
- ❖ Khởi tạo một menu mới với các thuộc tính lấy từ Action a.
- ❖ JMenu(String s)
- ❖ Khởi tạo một menu mới với tiêu đề là s
- ❖ JMenu(String s, boolean b)
- ❖ Khởi tạo một menu mới với tiêu đề là s và qui định là một menu tách rời hay không.

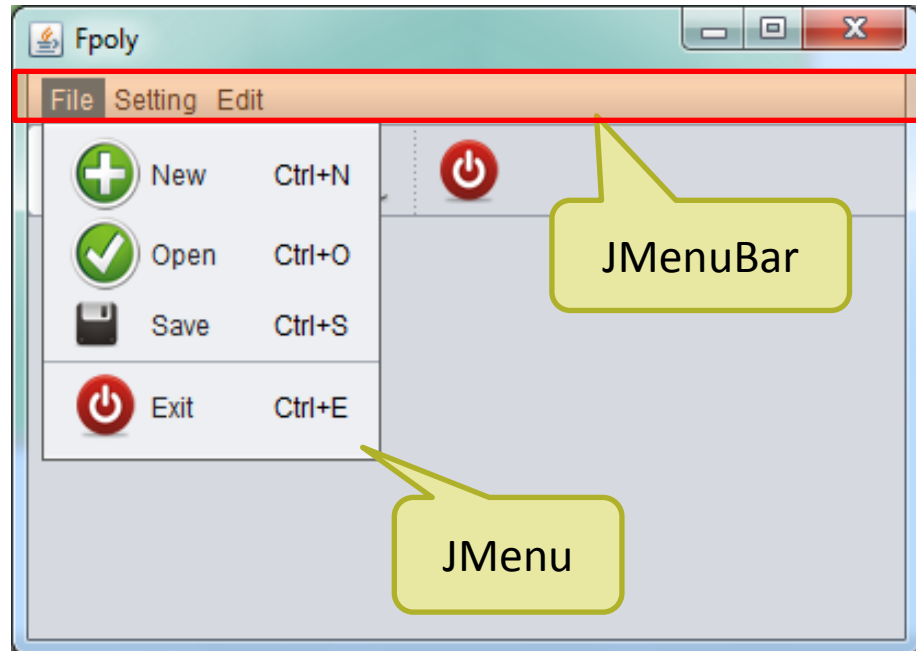
## ❑ Events: ChangeEvent

- ❖ void addChangeListener(ChangeListener listener)
- ❖ void removeChangeListener(ChangeListener listener)

## □ Các phương thức

JMenuItem	add(Action a)
Component	add(Component c)
Component	add(Component c, int index)
JMenuItem	add(JMenuItem menuItem)
JMenuItem	add(String s)
void	addMenuListener(MenuListener l)
void	addSeparator()
void	applyComponentOrientation(ComponentOrientation o)
PropertyChangeListener	createActionChangeListener(JMenuItem b)
JMenuItem	createActionComponent(Action a)
JMenu.WinListener	createWinListener(JPopupMenu p)
void	doClick(int pressTime)
void	fireMenuCanceled()
void	fireMenuDeselected()
...	...

- ❑ Dùng để tạo ra 1 MenuBar theo chiều ngang của component với 0, 1 hoặc nhiều phần tử gắn lên đó
- ❑ Dùng phương thức add để thêm vào các JMenuItem trên JMenuItem
- ❑ JMenuItem sẽ hiển thị các JMenuItem theo thứ tự từ trái sang phải.
- ❑ Khởi tạo
  - ❖ JMenuItem()
- ❑ Sự kiện
  - ❖ ActionListener



## ❑ Các phương thức

<a href="#">JMenu</a>	<a href="#">add(JMenu c)</a>
void	<a href="#">addNotify()</a>
<a href="#">AccessibleContext</a>	<a href="#">getAccessibleContext()</a>
<a href="#">Component</a>	<a href="#">getComponent()</a>
int	<a href="#">getComponentIndex(Component c)</a>
<a href="#">JMenu</a>	<a href="#">getHelpMenu()</a>
<a href="#">Insets</a>	<a href="#">getMargin()</a>
<a href="#">JMenu</a>	<a href="#">getMenu(int index)</a>
int	<a href="#">getMenuCount()</a>
<a href="#">SingleSelectionModel</a>	<a href="#">getSelectionModel()</a>
<a href="#">MenuElement[]</a>	<a href="#">getSubElements()</a>
<a href="#">MenuBarUI</a>	<a href="#">getUI()</a>
<a href="#">String</a>	<a href="#">getUIClassID()</a>
boolean	<a href="#">isBorderPainted()</a>
boolean	<a href="#">isSelected()</a>



- ❑ Có thể gắn 1 menu bar lên frame theo 1 trong 2 cách sau:
- ❑ Sử dụng setJMenuBar()
  - ❖ JFrame frame = new JFrame;
  - ❖ JMenuBar menuBar = new JMenuBar();  
frame.setJMenuBar(menuBar);
- ❑ Dùng layout để định vị
  - ❖ menuBar.setBorder(new BevelBorder(BevelBorder.RAISED));  
frame.getContentPane().add(menuBar, BorderLayout.SOUTH);

❑ JMenuItem là 1 loại nút đặc biệt ( xử lý mouseListener)

❑ Khởi tạo

❖ JMenuItem()

➤ Khởi tạo một JMenuItem không có tiêu đề và icon.

❖ JMenuItem(Action a)

➤ Khởi tạo một menu item với thuộc tính là a.

❖ JMenuItem(Icon icon)

➤ Khởi tạo một JMenuItem với icon.

❖ JMenuItem(String text)

➤ Khởi tạo một JMenuItem với tiêu đề là text.

❖ JMenuItem(String text, Icon icon)

➤ Khởi tạo một JMenuItem với tiêu đề và icon.

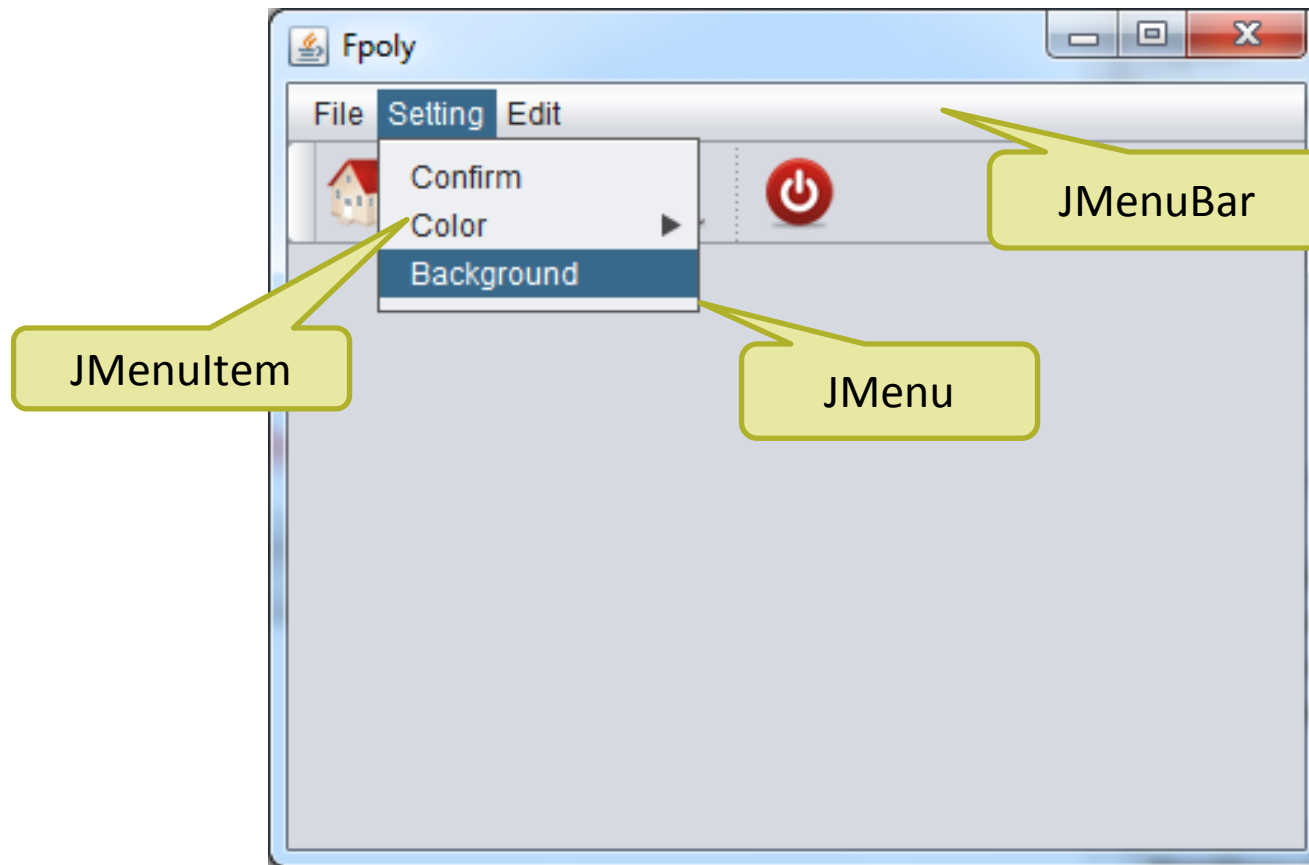
❖ JMenuItem(String text, int mnemonic)

➤ Khởi tạo một JMenuItem với tiêu đề và phím tắt

## □ Các phương thức

void	actionPropertyChanged(Action action, String propertyName)
void	addMenuDragMouseListener(MenuDragMouseListener l)
void	addMenuKeyListener(MenuKeyListener l)
void	configurePropertiesFromAction(Action a)
void	fireMenuDragMouseDragged(MenuDragMouseEvent event)
void	fireMenuDragMouseEntered(MenuDragMouseEvent event)
void	fireMenuDragMouseExited(MenuDragMouseEvent event)
void	fireMenuDragMouseReleased(MenuDragMouseEvent event)
...	...

- ❑ Các JMenuItem nằm trong JMenu, JMenuBar chứa các JMenuItem



SOF203 - NetBeans IDE 8.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

sof203.slide4

CBoxDemo.java

mnuFile [JMenu] - Navigator

Form MainMenu

Other Components

[JFrame]

jMenuBar1 [JMenuBar]

mnuFile [JMenu]

mnuFileNew [JMenuItem]

mnuFileOpen [JMenuItem]

mnuFileSave [JMenuItem]

jSeparator1 [Separator]

mnuFileExit [JMenuItem]

mnuSetting [JMenu]

jMenuItem4 [JMenuItem]

jMenu1 [JMenu]

jMenuItem5 [JMenuItem]

mnuEdit [JMenu]

jRadioButtonMenuItem1 [JRadioButtonMenuItem]

jRadioButtonMenuItem2 [JRadioButtonMenuItem]

jRadioButtonMenuItem3 [JRadioButtonMenuItem]

jCheckBoxMenuItem1 [JCheckBoxMenuItem]

jCheckBoxMenuItem2 [JCheckBoxMenuItem]

jCheckBoxMenuItem3 [JCheckBoxMenuItem]

jToolBar1 [JToolBar]

Source Design History

File Setting Edit

Text Area

Slider

Formatted Field

Spinner

Text Pane

Tree

Scroll Bar

Progress Bar

Password Field

Separator

Editor Pane

Table

Swing Menus

Menu

Menu Item

Menu Item / CheckBox

Menu Item / RadioButton

Popup Menu

Separator

Multiple Objects - Properties

Properties Binding

Events Code

Properties

background [240,240,240]

border <Different Valu...

foreground [0,0,0]

toolTipText

Other Properties

UIClassID <Different Valu...

alignmentX 0.5

alignmentY <Different Valu...

mnuFile [JMenu], mnuSetting [JMe...

mnuFile [JMenu], mnuSetting [JMenu], mnuEdit [JMenu], jMenuBar1 [JMenuBar]

Output - SOF203 (run)

run:

HTTP Server Monitor

SOF203 (run) running...

3:50 PM 7/27/2017

- ❑ Là một loại menu đặc biệt mà không cần phải gắn vào menu bar
- ❑ Nó có thể được hiển thị ở bất cứ vị trí nào trên thành phần chứa.
- ❑ Bạn có thể thêm, chèn một JMenuItem, 1 component hay 1 Action tùy ý vào popup menu này với phương thức add() và insert()
- ❑ JPopupMenu sẽ gán cho mỗi menu item 1 số thứ tự rồi gắn chúng vào popup menu theo layout mà nó có
- ❑ Bạn cũng có thể thêm 1 separator vào popup menu với phương thức addSeparator()

## ❑ Khởi tạo

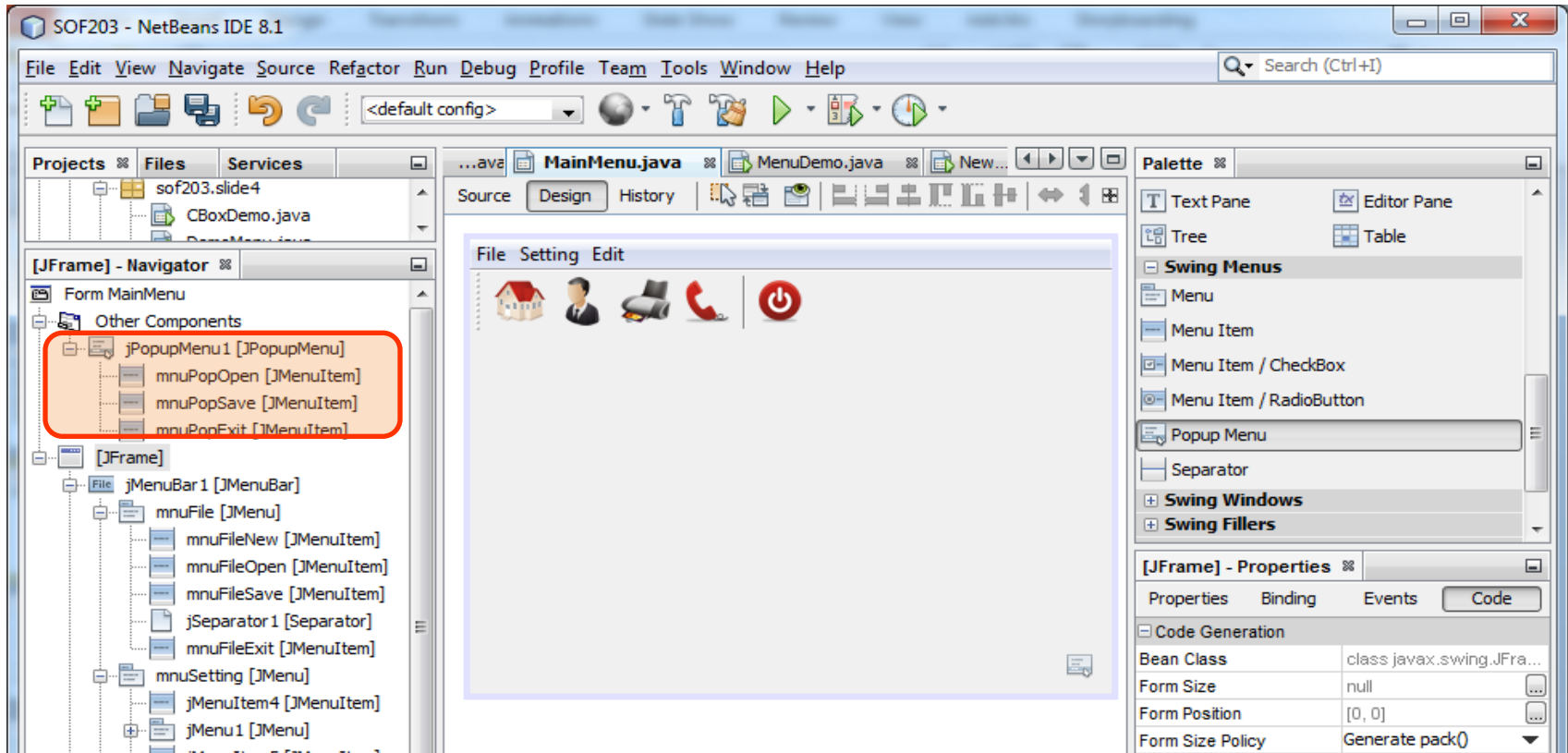
- ❖ `public JPopupMenu()`
- ❖ `public JPopupMenu(String title)`
- ❖ `public JMenuItem add(JMenuItem menuItem)`
- ❖ `public Component add(Component c)`
- ❖ `public JMenuItem add(Action a)`
- ❖ `public JMenuItem insert(Action a, int index)`
- ❖ `public Component insert(Component comp, int index)`
- ❖ `public void addSeparator()`

❑ Để hiển thị popup menu dùng phương thức  
show()

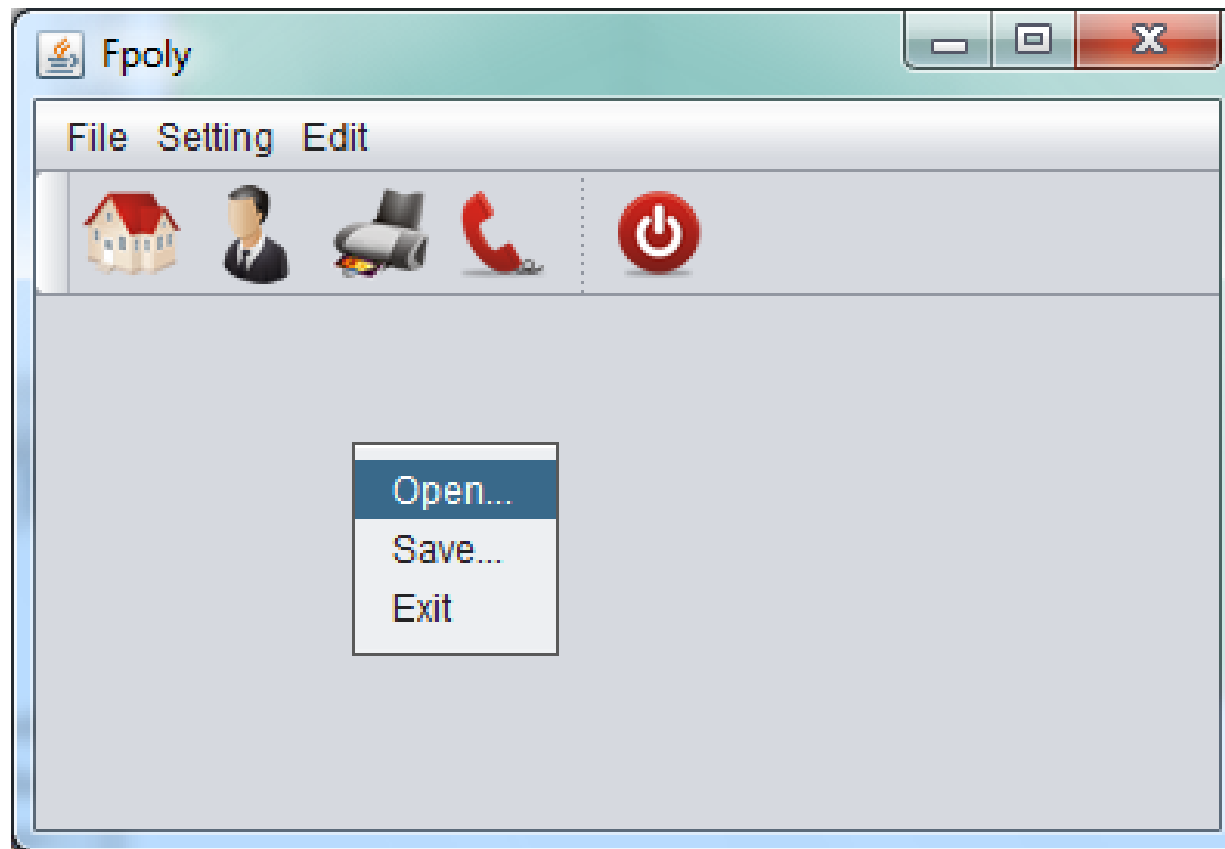
```
public void processMouseEvent(MouseEvent e) {  
    if (e.isPopupTrigger()) {  
        popup.show(this, e.getX(), e.getY());  
    } else {  
        super.processMouseEvent(e);  
    }  
}
```



## ❑ Ví dụ JPopupMenu



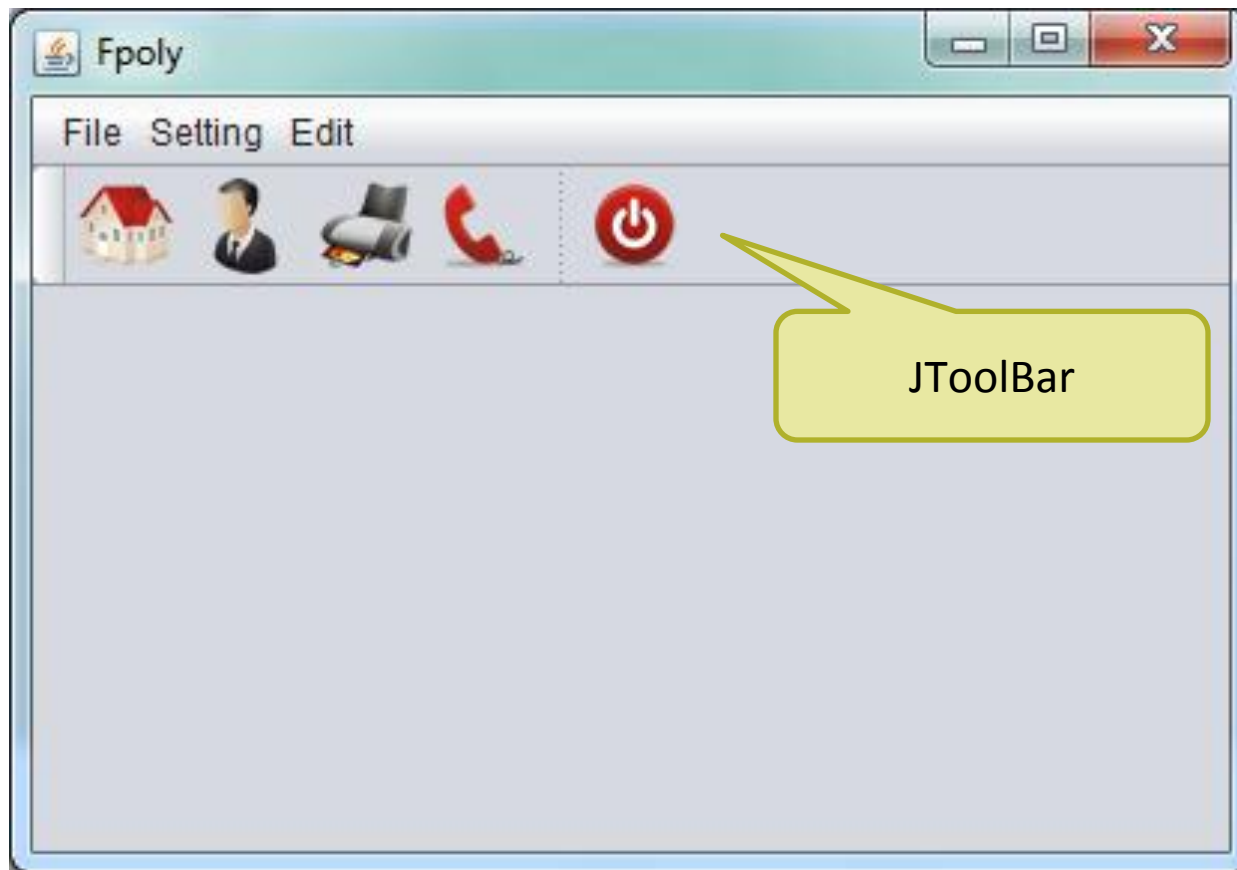
## ❑ Ví dụ JPopupMenu



- ❑ Là lớp chứa cho nhiều thành phần khác
- ❑ Khi 1 component được gắn vào tool bar nó sẽ được định vị từ trái sang phải theo chỉ mục của nó
- ❑ Khởi tạo
  - ❖ JToolBar()
    - Khởi tạo một tool bar; Chiều mặc định nằm ngang.
  - ❖ JToolBar(int orientation)
    - Khởi tạo một tool bar với chiều ngang/dọc.
  - ❖ JToolBar(String name)
    - Khởi tạo một tool bar với tên gọi name.
  - ❖ JToolBar(String name, int orientation)
- ❑ Sự kiện
  - ❖ Dùng Sự kiện `PropertyChangeEvent` khi có 1 sự thay đổi giá trị của các thuộc tính

## ☐ Các phương thức

<a href="#"><u>JButton</u></a>	<code>add(Action a)</code>
<code>protected void</code>	<code>addImpl(Component comp, Object constraints, int index)</code>
<code>void</code>	<a href="#"><u>addSeparator()</u></a>
<code>void</code>	<code>addSeparator(Dimension size)</code>
<a href="#"><u>protected PropertyChangeListener</u></a>	<code>createActionChangeListener(JButton b)</code>
<a href="#"><u>protected JButton</u></a>	<code>createActionComponent(Action a)</code>
<a href="#"><u>AccessibleContext</u></a>	<a href="#"><u>getAccessibleContext()</u></a>
<a href="#"><u>Component</u></a>	<a href="#"><u>getComponentAtIndex(int i)</u></a>
<code>int</code>	<code>getComponentIndex(Component c)</code>
<a href="#"><u>Insets</u></a>	<a href="#"><u>getMargin()</u></a>
<code>int</code>	<a href="#"><u>getOrientation()</u></a>
<a href="#"><u>ToolBarUI</u></a>	<a href="#"><u>getUI()</u></a>
<a href="#"><u>String</u></a>	<a href="#"><u>getUIClassID()</u></a>
<code>boolean</code>	<a href="#"><u>isBorderPainted()</u></a>
<code>boolean</code>	<a href="#"><u>isFloatable()</u></a>
<code>...</code>	<code>...</code>



DemoSOF203 - NetBeans IDE 8.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

<default config>

Projects Files Services

Slide4  
MainMenu.java  
MenuDemo.java

jMenuBar1 [JMenuBar] - Navigator

Form MenuDemo  
Other Components  
[JFrame]  
jMenuBar1 [JMenuBar]  
mnFile [JMenu]  
mnNew [JMenuItem]  
mnOpen [JMenuItem]  
jSeparator1 [JSeparator]  
mnSave [JMenuItem]  
mnColor [JMenu]  
mnRed [JRadioButtonMenuItem]  
mnGreen [JRadioButtonMenuItem]  
mnBlue [JRadioButtonMenuItem]  
jSeparator2 [JSeparator]  
mnOther [JMenuItem]  
mnSys [JMenu]  
mnAbout [JMenuItem]  
mnExit [JMenuItem]  
jPanel1 [JPanel]  
jScrollPane1 [JScrollPane]  
edPage [JEditorPane]  
jPanel2 [JPanel]  
txtComment [JTextField]  
jToolBar1 [JToolBar]  
OK btnAbout [JButton]  
OK btnNew [JButton]  
OK btnColor [JButton]

Source Design History

File Color System

About us New File Set text Color

Welcome to Fpoly

Palette

Tabbed Pane  
Panel  
Split Pane  
Scroll Pane  
Tool Bar  
Desktop Pane  
Internal Frame  
Layered Pane  
Swing Controls  
Menu Bar  
Label Label

jMenuBar1 [JMenuBar] - ...

Properties Binding  
Events Code

Properties

background	[240,240,240]
border	[MenuBarBorder]
foreground	[0,0,0]
toolTipText	

Other Properties

UIClassID	MenuBarUI
alignmentX	0.5
alignmentY	0.0
autoscrolls	
baselineResizeBehavior	OTHER
borderPainted	<input checked="" type="checkbox"/>

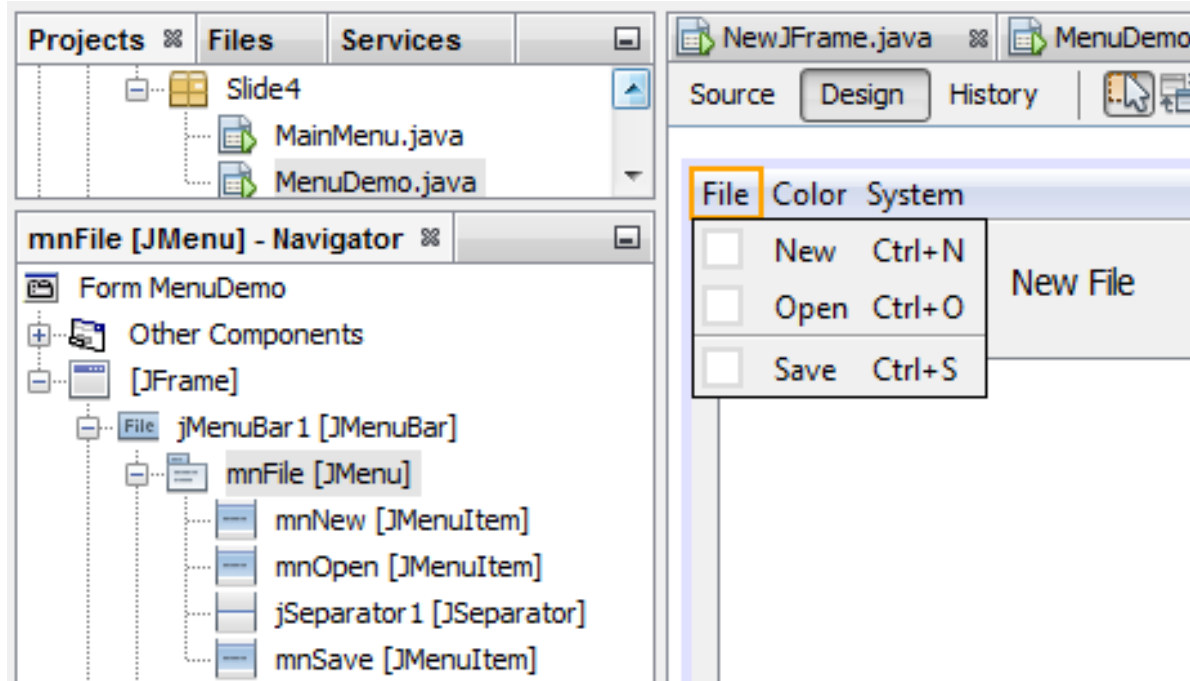
jMenuBar1 [JMenuBar]

Output - DemoSOF203 (run)

run:

HTTP Server Monitor

DemoSOF203 (run) running... 373:9

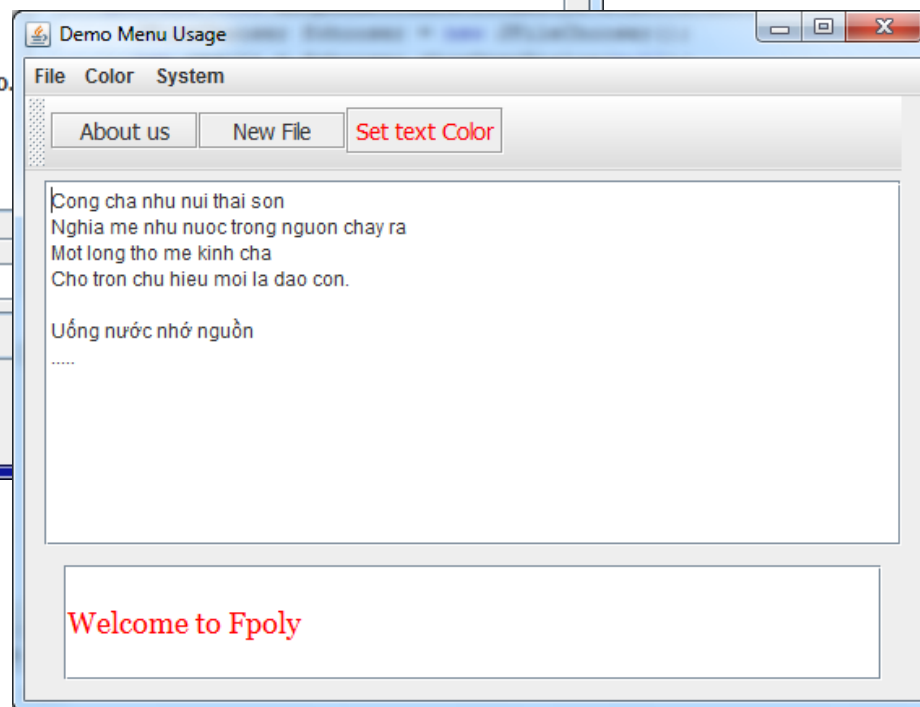
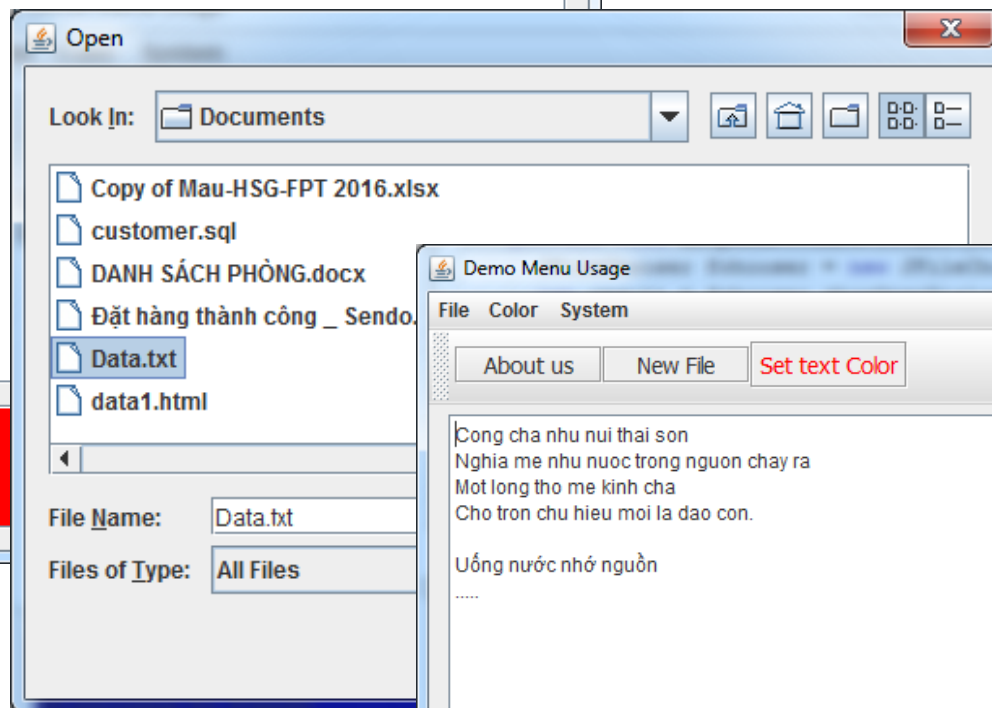
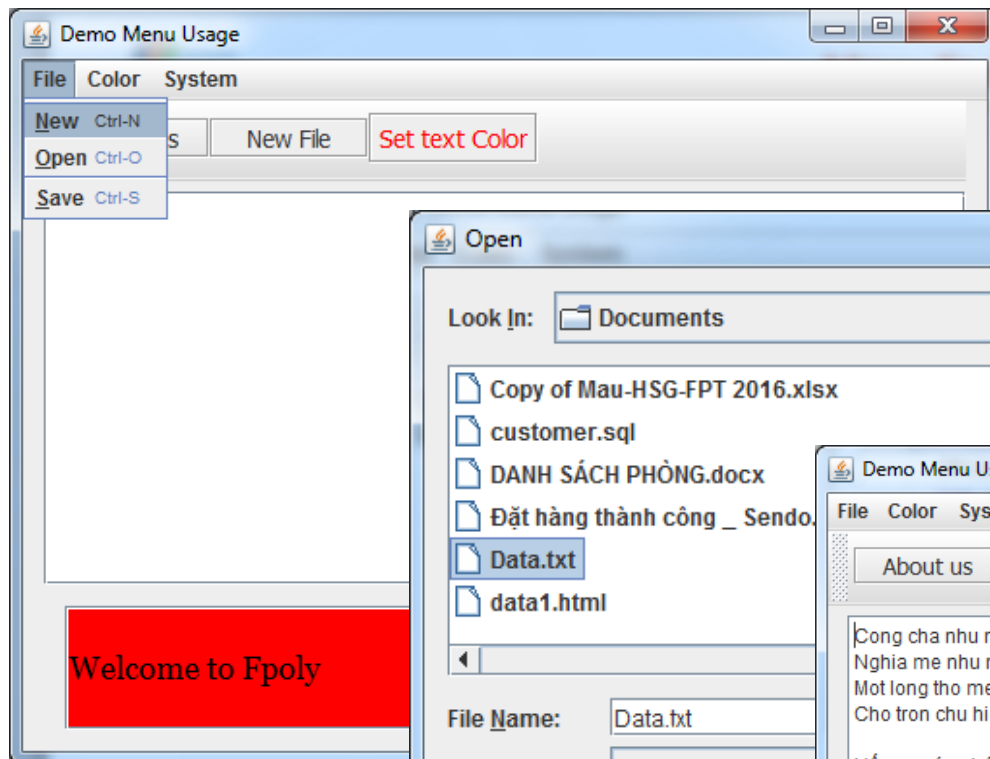


```
private void mnNewActionPerformed(java.awt.event.ActionEvent evt) {
    edPage.setText("");
}
```

```
private void mnOpenActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser fchooser = new JFileChooser();
    int result = fchooser.showOpenDialog(null);
    if (result == JFileChooser.APPROVE_OPTION) {
        try {
            edPage.setPage("file:///\" + fchooser.getSelectedFile().getPath());
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

```
private void mnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser fChooser = new JFileChooser();
    int result = fChooser.showSaveDialog(null);
    if(result==JFileChooser.APPROVE_OPTION){
        File fname = fChooser.getSelectedFile();
        try {
            PrintWriter pw = new PrintWriter(fname);
            pw.println(edPage.getText());
            pw.close();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```



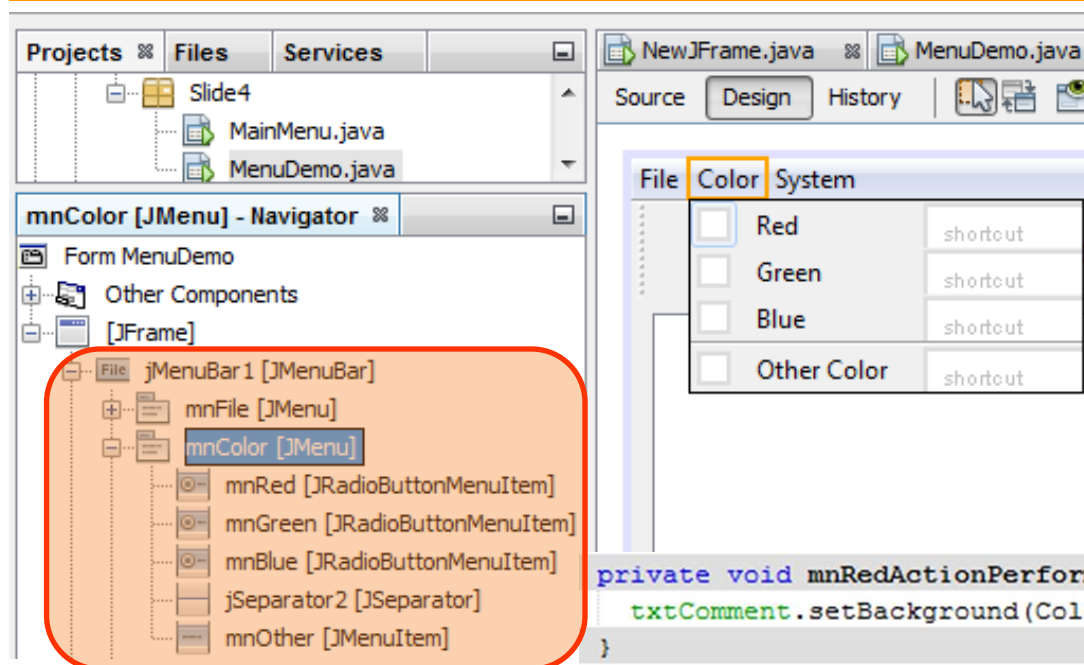




# DEMO

Chạy và giải thích





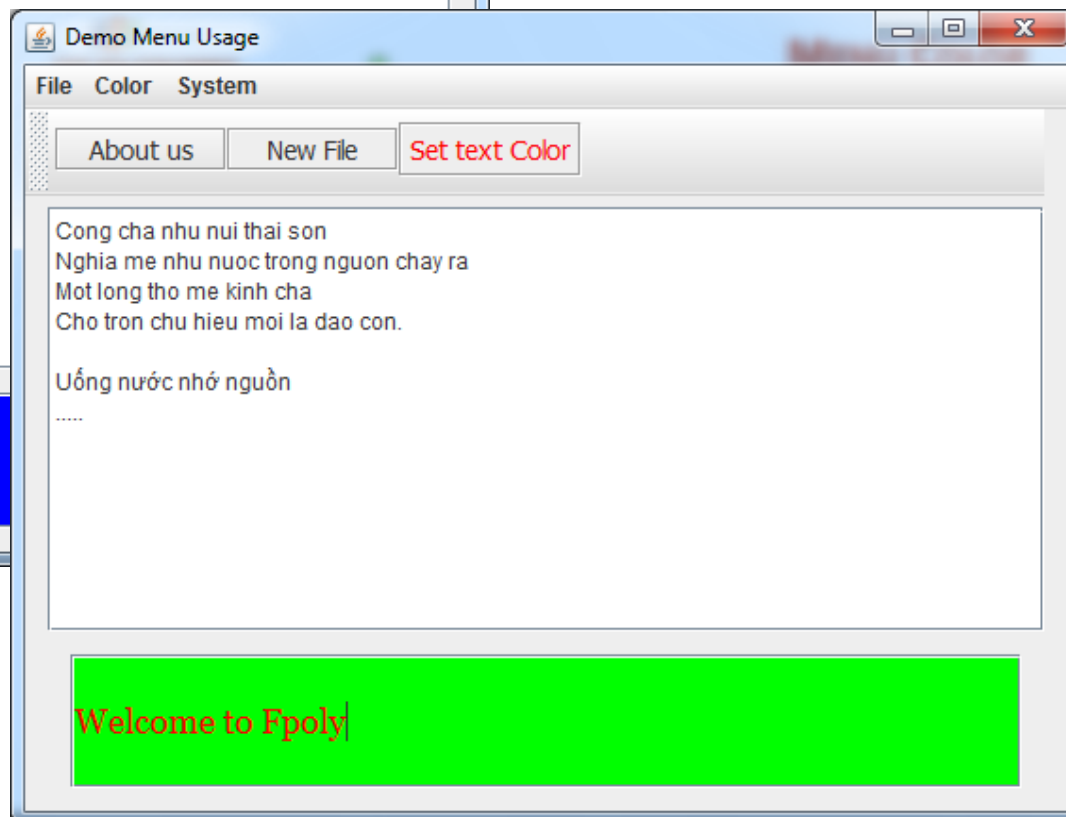
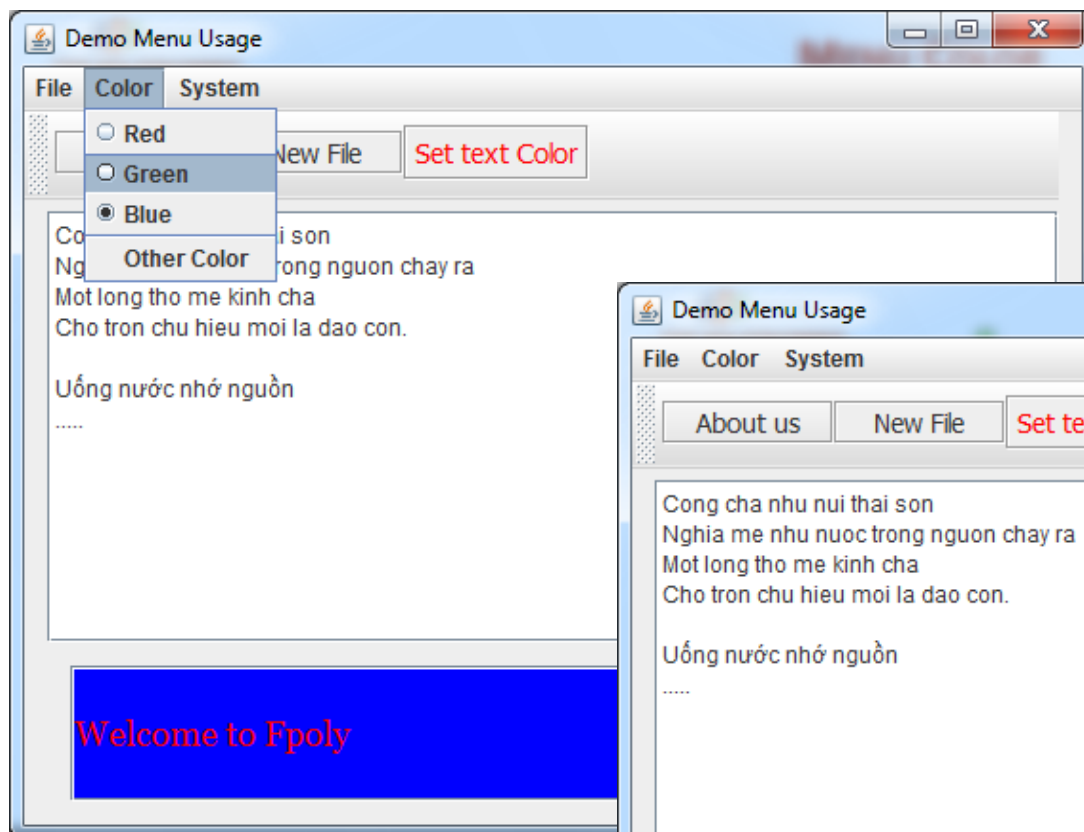
```
private void mnRedActionPerformed(java.awt.event.ActionEvent evt) {
    txtComment.setBackground(Color.red);
}
```

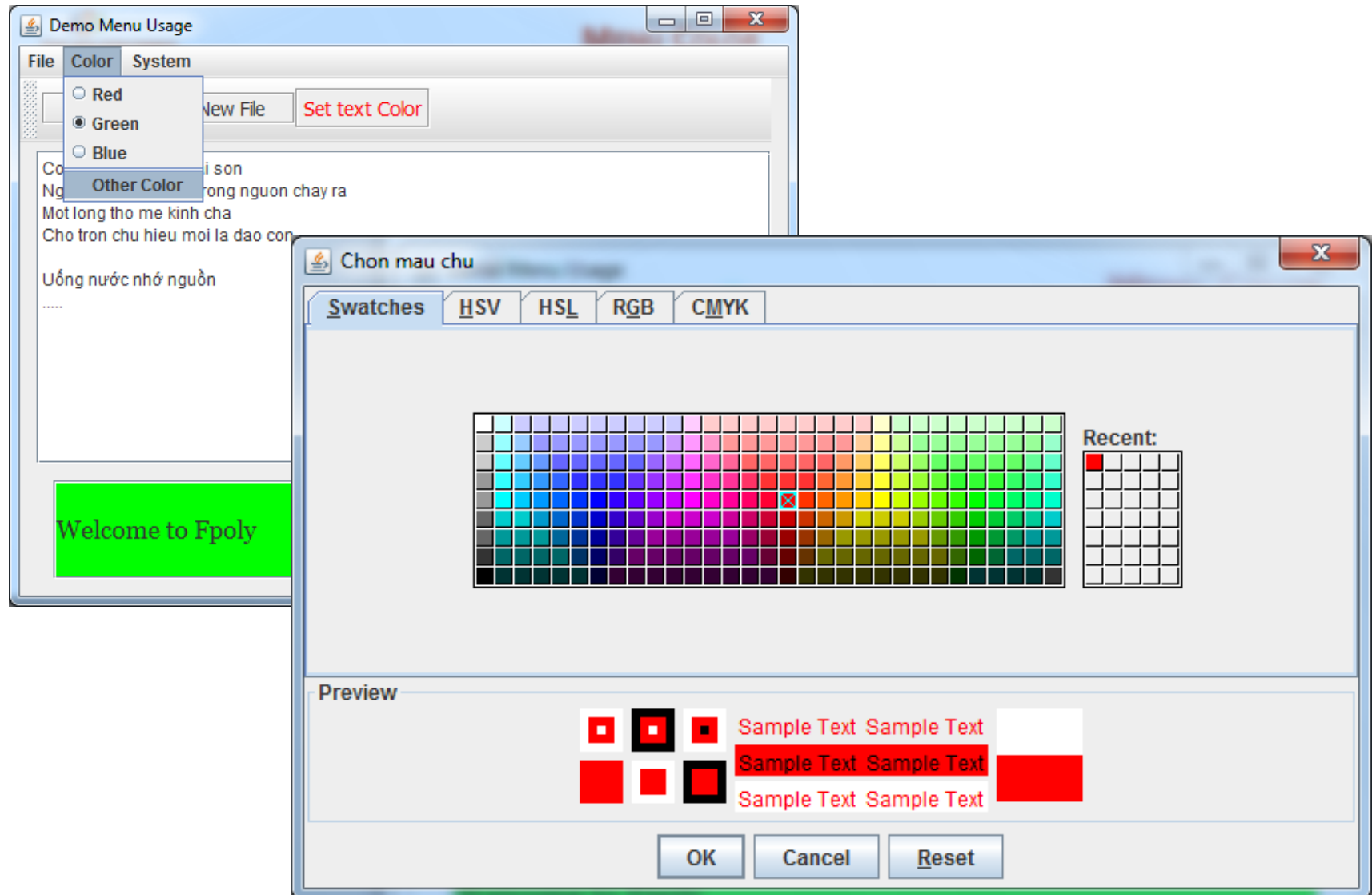
```
private void mnGreenActionPerformed(java.awt.event.ActionEvent evt) {
    txtComment.setBackground(Color.GREEN);
}
```

```
private void mnBlueActionPerformed(java.awt.event.ActionEvent evt) {
    txtComment.setBackground(Color.blue);
}
```

```
private void setTextColor() {
    Color c = JColorChooser.showDialog(null, "Chon mau chu", Color.white);
    txtComment.setForeground(c);
}
```

```
private void mnOtherActionPerformed(java.awt.event.ActionEvent evt) {
    setTextColor();
}
```



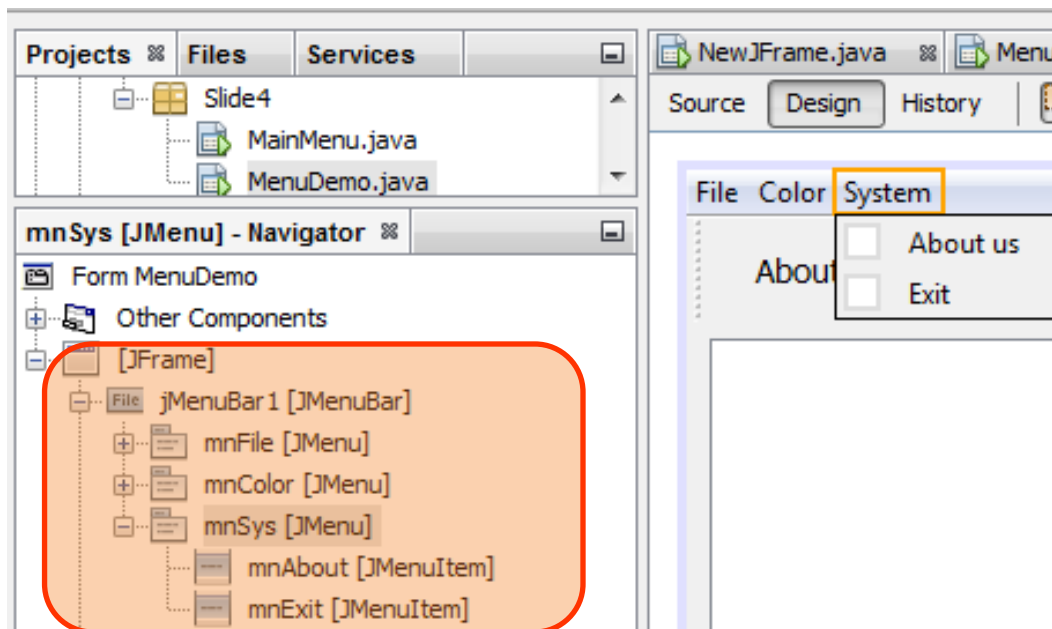




# DEMO

Chạy và giải thích

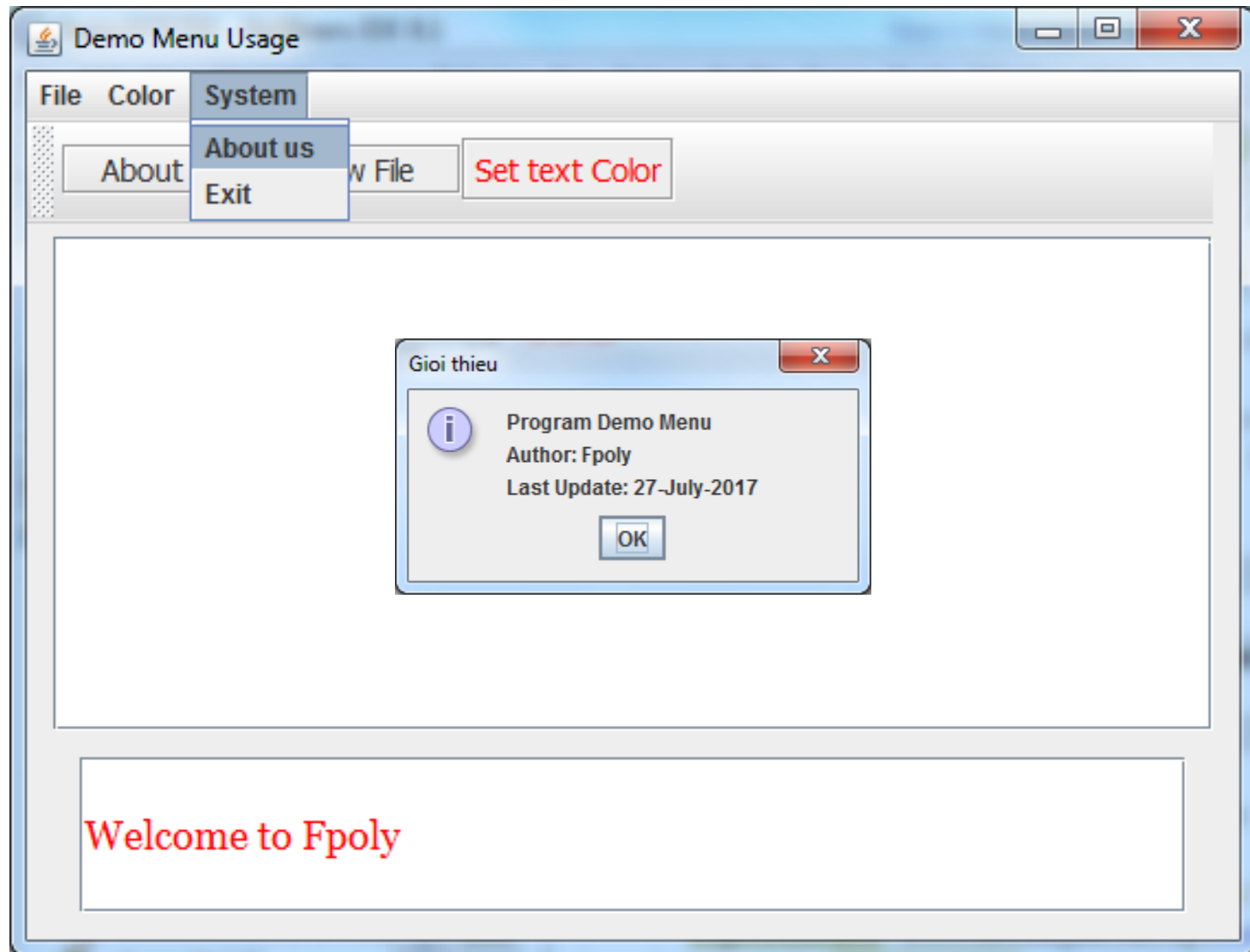




```
private void mnAboutActionPerformed(java.awt.event.ActionEvent evt) {
    showAboutUs();
}

private void showAboutUs() {
    String str = "Program Demo Menu";
    str += "\nAuthor: Fpoly";
    str += "\nLast Update: 27-July-2017";
    JOptionPane.showMessageDialog(null, str, "Gioi thieu",
        JOptionPane.INFORMATION_MESSAGE);
}

private void mnExitActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```





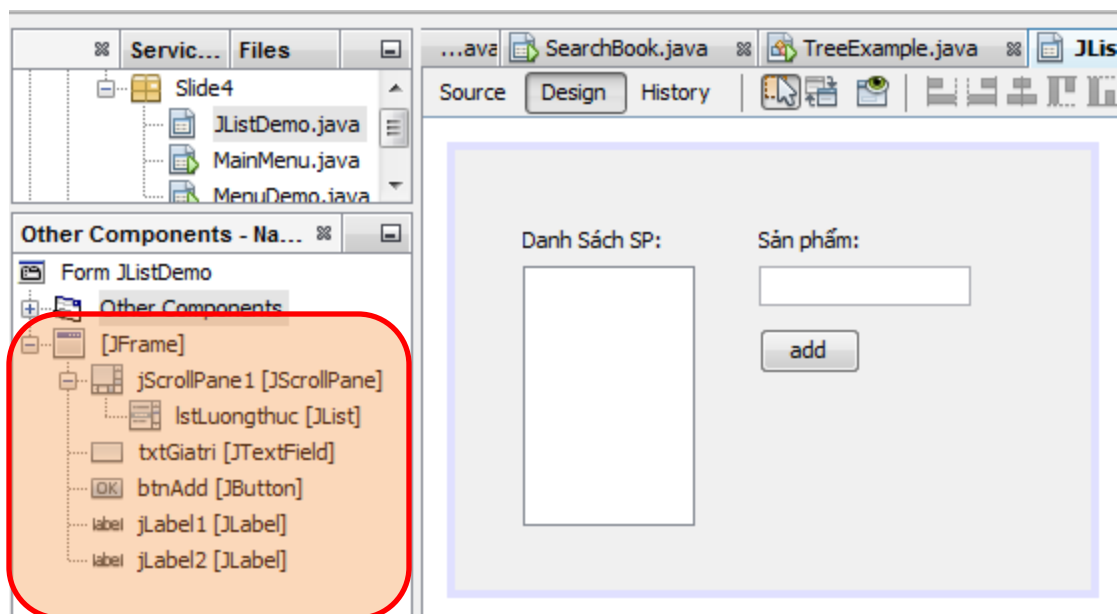


# DEMO

Chạy và giải thích

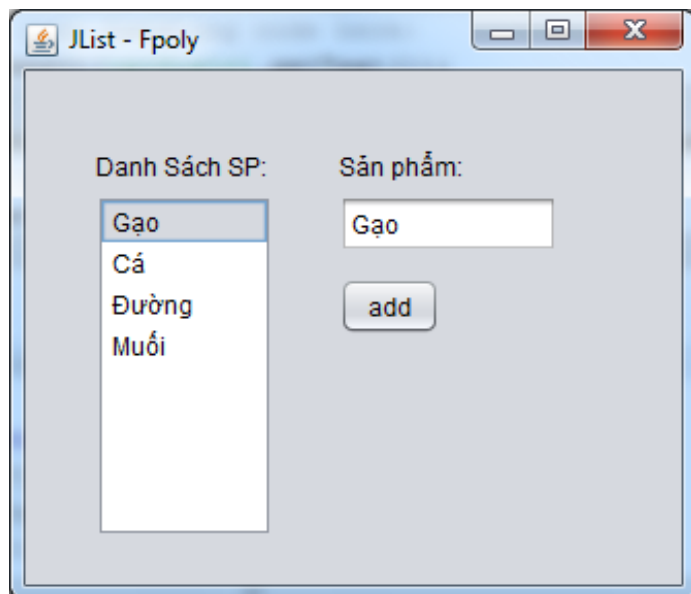


- ❑ Đối tượng dùng để hiển thị danh sách, người sử dụng có thể chọn một hoặc nhiều mục
- ❑ Khởi tạo
  - ❖ JList()
    - Khởi tạo một JList rỗng, chỉ đọc.
  - ❖ JList(E[] listData)
    - Khởi tạo một JList hiển thị các phần tử trong mảng E.



```
public class JListDemo extends javax.swing.JFrame {
    DefaultListModel model;
    public JListDemo() {
        initComponents();
        model = new DefaultListModel();
        String items[]={"Gạo","Cá","Đường","Muối"};
        for(String item : items){
            model.addElement(item);
        }
        lstLuongthuc.setModel(model);
    }
    //.....
}
```

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    model.addElement(txtGiatrri.getText());
}
private void lstLuongthucMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    txtGiatrri.setText(lstLuongthuc.getSelectedValue());
}
}
```



JList - Fpoly

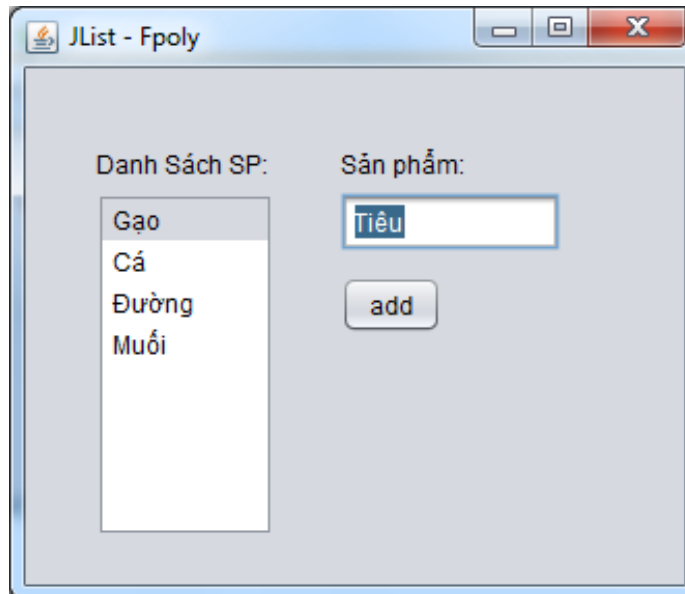
Danh Sách SP:

- Gạo
- Cá
- Đường
- Muối

Sản phẩm:

Gạo

add



JList - Fpoly

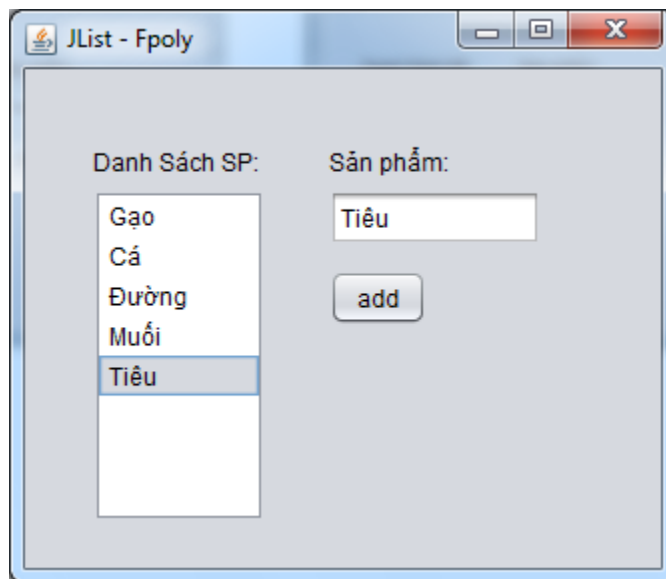
Danh Sách SP:

- Gạo
- Cá
- Đường
- Muối

Sản phẩm:

Tiêu

add



JList - Fpoly

Danh Sách SP:

- Gạo
- Cá
- Đường
- Muối
- Tiêu

Sản phẩm:

Tiêu

add

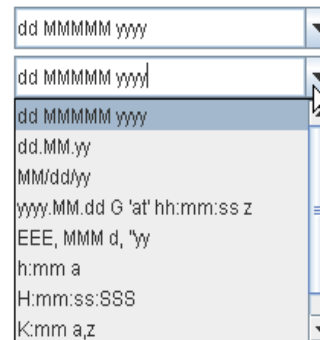


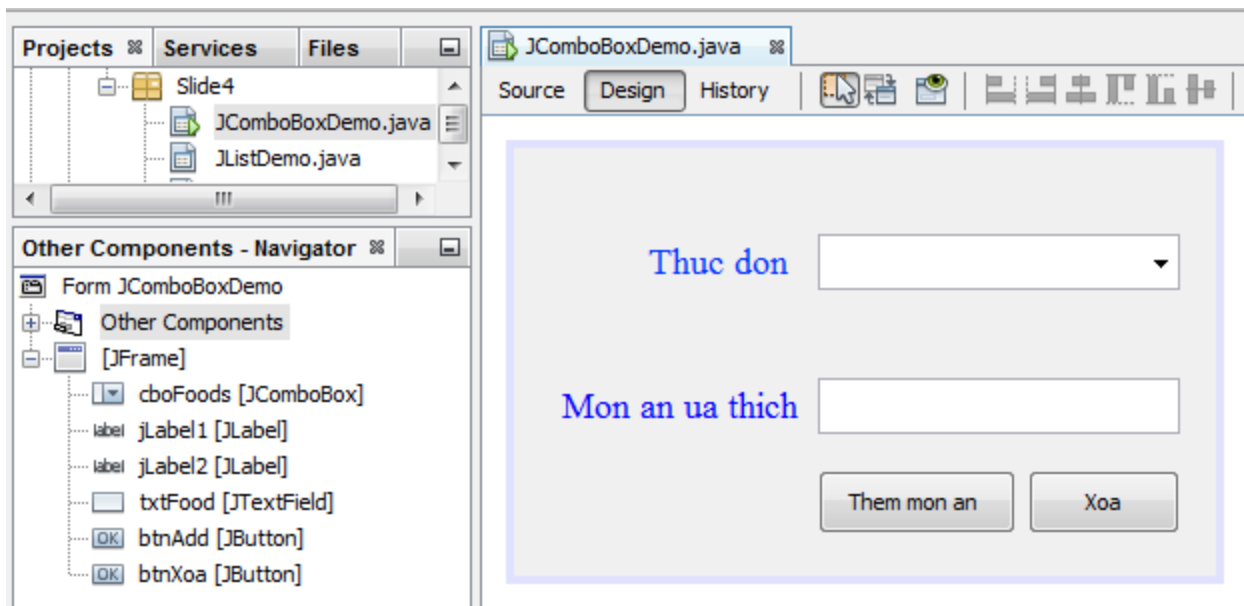
# DEMO

Chạy và giải thích



- ❑ Là thành phần cho phép người dùng chọn một item từ một danh sách item, có 2 loại JComboBox
- ❑ JComboBox mặc định là dạng không chỉnh sửa được, người dùng chỉ được phép chọn từ danh sách
- ❑ Loại thứ 2 là JComboBox chỉnh sửa được, người dùng gõ vào vùng text hoặc chọn từ danh sách
- ❑ Khởi tạo
  - ❖ JComboBox()
    - Tạo một ComboBox với dữ liệu mặc định
  - ❖ JComboBox(E[] items)
    - Tạo một ComboBox chứa các item trong mảng E



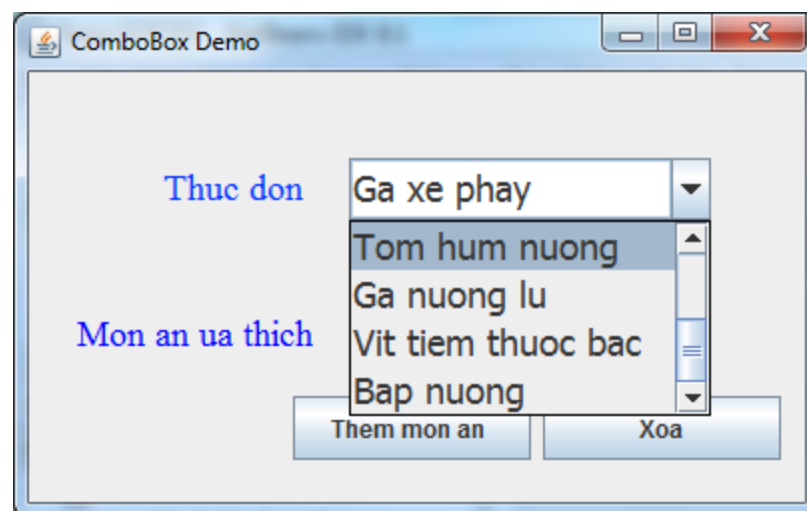
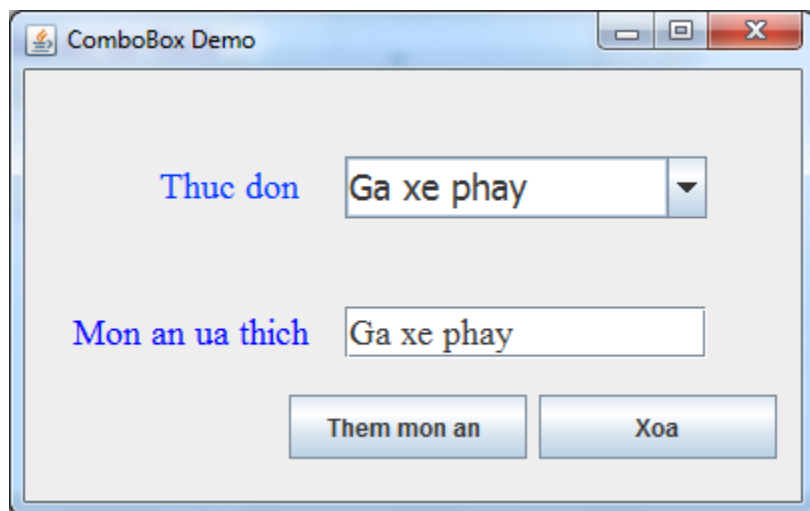


```
public class JComboBoxDemo extends javax.swing.JFrame {
    public JComboBoxDemo() {
        initComponents();
        String[] Menus = {"Ga xe phay", "Hot vit lon", "Cua rang me", "Che buoi",
            "Tom hum nuong", "Ga nuong lu", "Vit tiem thuoc bac", "Bap nuong"};
        for (String item : Menus) {
            cboFoods.addItem(item);
        }
    }
    //.....
}
```

```
private void cboFoodsItemStateChanged(java.awt.event.ItemEvent evt) {
    String food = cboFoods.getSelectedItem().toString();
    txtFood.setText(food);
}
```

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    cboFoods.addItem(txtFood.getText());
}
```

```
private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cboFoods.removeItem(txtFood.getText());
}
```







# DEMO

Chạy và giải thích





## LẬP TRÌNH JAVA 3

**BÀI 4: JMENU, JLIST, JCOMBOBOX,  
JTABLE, JTREE**

**PHẦN 2**

- ❑ Jtable dùng để hiển thị dữ liệu dưới dạng bảng biểu, JTable cũng cho phép chúng ta dễ dàng sửa đổi thông tin, đặt kích cỡ và đầu đề cho từng cột, và điều khiển cách dữ liệu hiển thị trong bảng biểu.
- ❑ Bản chất của JTable là nó lấy dữ liệu từ một data model và hiển thị dữ liệu từ đó lên.

The Header contains  
Column labels

First Name	Last Name	Sport	# of Years	Vegetarian
Kathy	Smith	Snowboarding	5	<input type="checkbox"/>
John	Doe	Rowing	3	<input checked="" type="checkbox"/>
Sue	Black	Knitting	2	<input type="checkbox"/>
Jane	White	Speed reading	20	<input checked="" type="checkbox"/>
Jan	Black	Reading	10	<input type="checkbox"/>

Each Cell displays  
a data item

Each Column displays  
one type of data

## ❑ Khởi tạo

### ❖ `JTable()`

- Khởi tạo một bảng kích thước 1x1

### ❖ `JTable(int numRows, int numColumns)`

- Khởi tạo một bảng với kích thước `numRows` x `numColumns`

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

- Khởi tạo một mảng 2 chiều (`rowData`) với tên các cột (`columnNames`)

## □ Ví dụ:

```

1  Object[][] data = {
2      {"Kathy", "Smith",
3       "Snowboarding", new Integer(5), new Boolean(false)},
4      {"John", "Doe",
5       "Rowing", new Integer(3), new Boolean(true)},
6      {"Sue", "Black",
7       "Knitting", new Integer(2), new Boolean(false)},
8      {"Jane", "White",
9       "Speed reading", new Integer(20), new Boolean(true)},
10     {"Joe", "Brown",
11      "Pool", new Integer(10), new Boolean(false)}
12 };
13 Object columnNames = {"First Name",
14                        "Last Name",
15                        "Sport",
16                        "# of Years",
17                        "Vegetarian"};
18 JTable table = new JTable(data, columnNames);
19 JScrollPane scrollPane = new JScrollPane(table);
20 frame.add(scrollPane, BorderLayout.CENTER);

```

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project named "Form jTableDemo" containing a "JFrame" with several components:
  - jPanel1 [JPanel] containing jScrollPane1 [JScrollPane] and tblProduct [JTable].
  - jPanel2 [JPanel] containing labels (jLabel1 to JLabel5), text fields (txtMaSP, txtTenSP, txtDonGia, txtNCC), and a combo box (cboDVT).
  - jPanel3 [JPanel] containing a FlowLayout with buttons: btnAdd [JButton], btnDel [JButton], and btnEdit [JButton].
- Main Window:** Displays the visual representation of the application.
  - Title Bar:** "Danh sach San Pham"
  - Table:** A table with 4 columns (Title 1, Title 2, Title 3, Title 4) and 5 rows (1 header, 4 data rows).
  - Form Fields:**
    - Ma san pham: Text field
    - Ten san pham: Text field
    - Don vi tinh: Combo box (currently showing "Kg")
    - Don gia: Text field
    - Nha cung cap: Text field
  - Buttons:** "Them San Pham", "Xoa San pham", and "Dieu chinh thong tin" at the bottom.
- Output Console:** Shows "Output - SOF203 (run)".

```
public class JTableDemo extends javax.swing.JFrame {
    DefaultTableModel modelProducts;
    public JTableDemo() {
        initComponents();
        String[] headers={"Ma SP","Ten SP","DVT","Don gia ban","Nha Cung Cap"};
        String[][]data={
            {"SP01","Dau goi dau Head & Shoulder","Chai","34000","Unilevers"},
            {"SP02","Xa bong Omo","Thung","124000","Unilevers"},
            {"SP03","Dau an Tuong An 5 lit","Chai","100000","Tuong An"},
            {"SP04","Mi an lien Hao Hao","Thung","75000","AceCook"},
            {"SP05","Duong tinh luyen","Kg","14000","Duong Bien Hoa"},
            {"SP06","Sua Ong Tho","Lon","13000","VinaMilk"},
            {"SP07","Ruou Nho Nguyen Chat","Chai","130000","Cty Huong Nhien"}
        };
        modelProducts = new DefaultTableModel(data, headers);
        tblProduct.setModel(modelProducts);
    }
    //.....
}
```

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    Vector dataRow = new Vector();
    dataRow.add(txtMaSP.getText().trim());
    dataRow.add(txtTenSP.getText().trim());
    dataRow.add(cboDVT.getSelectedItem());
    dataRow.add(txtDonGia.getText().trim());
    dataRow.add(txtNCC.getText().trim());
    modelProducts.addRow(dataRow);
}
```

```
private void btnDelActionPerformed(java.awt.event.ActionEvent evt) {
    int row = tblProduct.getSelectedRow();
    if(row<0){
        JOptionPane.showMessageDialog(null, "Chua chon dong nao trong bang!");
        return;
    }
    modelProducts.removeRow(row);
}

private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
    int row = tblProduct.getSelectedRow();
    if(row<0){
        JOptionPane.showMessageDialog(null, "Chua chon dong de thay doi du lieu.");
        return;
    }
    modelProducts.setValueAt(txtTenSP.getText().trim(), row, 1);
    modelProducts.setValueAt(cboDVT.getSelectedItem(), row, 2);
    modelProducts.setValueAt(txtDonGia.getText().trim(), row, 3);
    modelProducts.setValueAt(txtNCC.getText().trim(), row, 4);
}

private void tblProductMouseClicked(java.awt.event.MouseEvent evt) {
    int r = tblProduct.getSelectedRow();
    if(r<0){
        return;
    }
    txtMaSP.setText(tblProduct.getValueAt(r, 0).toString());
    txtTenSP.setText(tblProduct.getValueAt(r, 1).toString());
    cboDVT.setSelectedItem(tblProduct.getValueAt(r, 2));
    txtDonGia.setText(tblProduct.getValueAt(r, 3).toString());
    txtNCC.setText(tblProduct.getValueAt(r, 4).toString());
}
```



Table Demo

Danh sách Sản Phẩm

Ma SP	Ten SP	DVT	Đơn giá bán	Nhà Cung Cấp
SP01	Dầu gội đầu Head & S...	Chai	34000	Unilevers
SP02	Xà bông Omo	Thung	124000	Unilevers
SP03	Dầu ăn Tuong An 5 lit	Chai	100000	Tuong An
SP04	Mì ăn liền Hào Hào	Thung	75000	AceCook
SP05	Dương tinh luyện	Kg	14000	Duong Bien Hoa
SP06	Sữa Ông Thọ	Lon	13000	VinaMilk
SP07	Rượu Nho Nguyễn Chất	Chai	130000	Cty Hương Nhiên

Ma sản phẩm

SP07

Tên sản phẩm

Rượu Nho Nguyễn Chất

Đơn vị tính

Chai

Đơn giá

130000

Nhà cung cấp

Cty Hương Nhiên

Thêm Sản Phẩm

Xóa Sản phẩm

Điều chỉnh thông tin

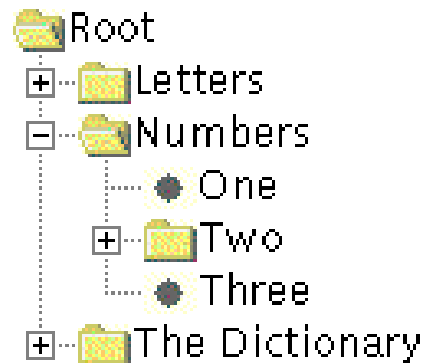


# DEMO

Chạy và giải thích



- ❑ Với lớp JTree, bạn có thể hiển thị dữ liệu phân cấp.
- ❑ Một đối tượng JTree không thực sự chứa dữ liệu của bạn; nó chỉ đơn giản là cung cấp một cái nhìn của các dữ liệu.



- ❑ Như hình vẽ, JTree hiển thị dữ liệu theo chiều dọc. Mỗi hàng được hiển thị bởi các cây có chứa chính xác một mục dữ liệu, được gọi là một nút. Mỗi cây có một nút gốc nút gốc duy nhất.

- ❑ Một nút có thể hoặc là có con hay không. Các nút có nút con gọi là các nút nhánh. Các nút mà không có nút con gọi là các nút lá.
- ❑ Một nút nhánh có thể có nhiều nút con. Để xem các nút con của một nhánh, ta bấm vào dấu “mở rộng”. Một chương trình có thể phát hiện những thay đổi trong trạng thái mở rộng các nút nhánh
- ❑ Một nút cụ thể trong một cây được xác định, hoặc bởi một TreePath, một đối tượng mà đóng gói một nút, hoặc bởi hàng đặc trưng của nó, trong đó mỗi hàng trong khu vực hiển thị sẽ hiển thị một nút

## □ Khởi tạo

### ❖ JTree(Object[] value)

- Khởi tạo một JTree với các thành phần của mảng Object là nút con, nút gốc chưa được xác định

### ❖ JTree(TreeNode root)

- Khởi tạo một JTree với nút gốc là root

- ❑ Ví dụ: Tạo một nút gốc, khởi tạo JTree từ nút gốc vừa tạo và đặt Tree vào một Scroll Pane

```
//Where instance variables are declared:
private JTree tree;
...
❑ public TreeDemo() {
    ...
    DefaultMutableTreeNode top =
        new DefaultMutableTreeNode("The Java Series");
    createNodes(top);
    tree = new JTree(top);
    ...
    JScrollPane treeView = new JScrollPane(tree);
    ...
}
```

```
//Add node con của root
DefaultMutableTreeNode category =
    new DefaultMutableTreeNode("Books for Java Programmers");
top.add(category);
```

```
DefaultMutableTreeNode book = null;
```

```
//Add node con cho node nhánh category
book = new DefaultMutableTreeNode(new BookInfo
    ("The Java Tutorial Continued: The Rest of the JDK",
    "tutorialcont.html"));
category.add(book);
book = new DefaultMutableTreeNode(new BookInfo
    ("The Swing Tutorial: A Guide to Constructing GUIs",
    "swingtutorial.html"));
category.add(book);
```

## ❑ Một số Phương thức

- ❖ node.add()
- ❖ node.remove()
- ❖ node.removeAllChildren()
- ❖ node.getChildCount()
- ❖ node.getChildAt()
- ❖ node.getParent()
- ❖ node.setParent()
- ❖ node.removeFromParent()
- ❖ node.getUserObject()
- ❖ node.setUserObject()
- ❖ model.getRoot()
- ❖ model.setRoot()

## ❑ Sự kiện

- ❖ TreeExpansion, TreeSelection, TreeWillExpand



- ❖ JMenu
- ❖ JMenuBar
- ❖ JMenuItem
- ❖ JPopupMenu
- ❖ JToolBar
- ❖ JList
- ❖ JComboBox
- ❖ JTable
- ❖ JTree





**Cảm ơn**