Lecture 02
Creating Graphical User Interface
Part 2

Events in Java

How to Use Menu

References:

(1) Textbook, chapter 11

(2) Java-Tutorials/tutorial-2015/uiswing/index.html

Contents

- Events in Java
- Working with Menu
- Demonstrations

1- Events in Java

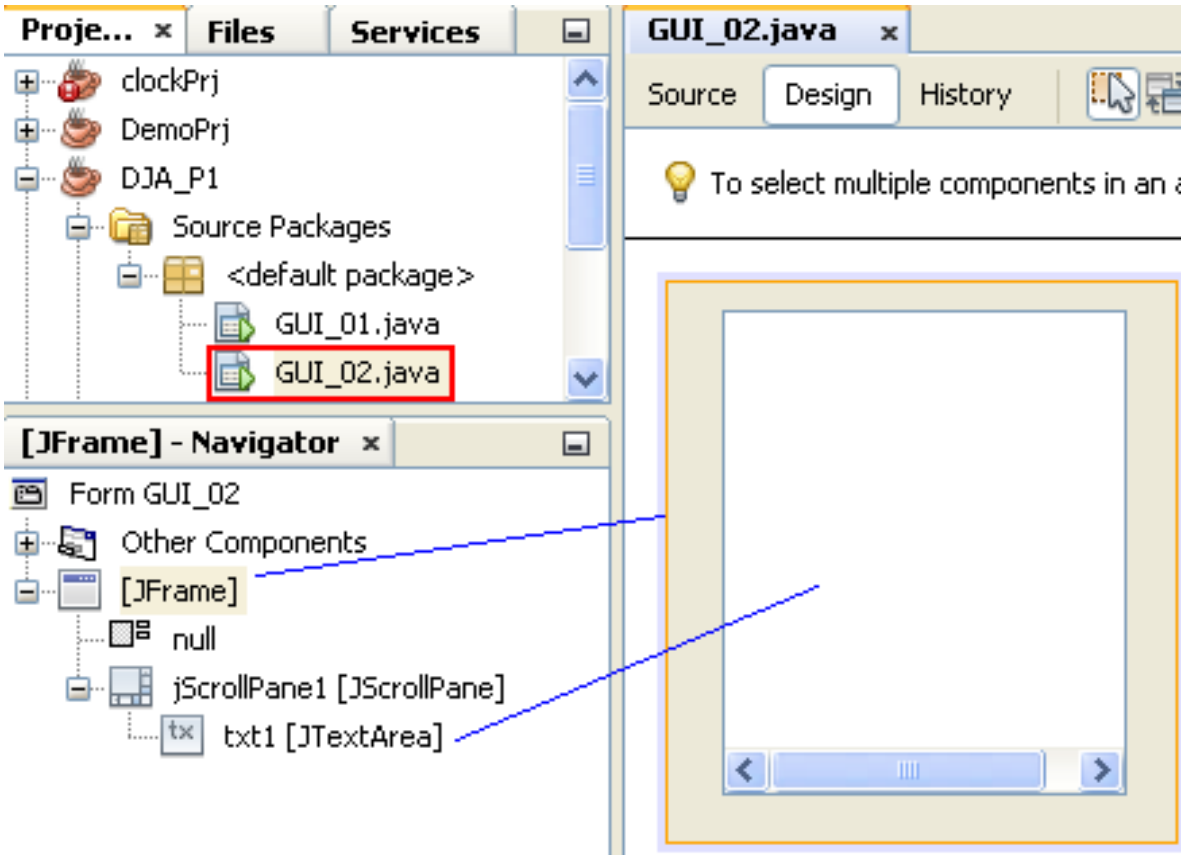
- java.util.**EventObject** (implements java.io.Serializable)
  - java.awt.**AWTEvent**
    - java.awt.event.**ActionEvent**
    - java.awt.event.**AdjustmentEvent**
    - java.awt.event.**ComponentEvent**
      - java.awt.event.**ContainerEvent**
      - java.awt.event.**FocusEvent**
      - java.awt.event.**InputEvent**
        - java.awt.event.**KeyEvent**
        - java.awt.event.**MouseEvent**
          - java.awt.event.**MouseWheelEvent**
      - java.awt.event.**PaintEvent**
      - java.awt.event.**WindowEvent**
    - java.awt.event.**HierarchyEvent**
    - java.awt.event.**InputMethodEvent**
    - java.awt.event.**InvocationEvent** (implements java.awt.ActiveEvent)
    - java.awt.event.**ItemEvent**
    - java.awt.event.**TextEvent**

- **2 types of events**:
  - Low-level events: FocusEvent, KeyEvent
  - High-level events: ActionEvent, ItemEvent
- Refer to the **java.awt.event** package for more details.

Demo 2: Window Events

This demonstration will help you exploring events on the window (Frame)

Design



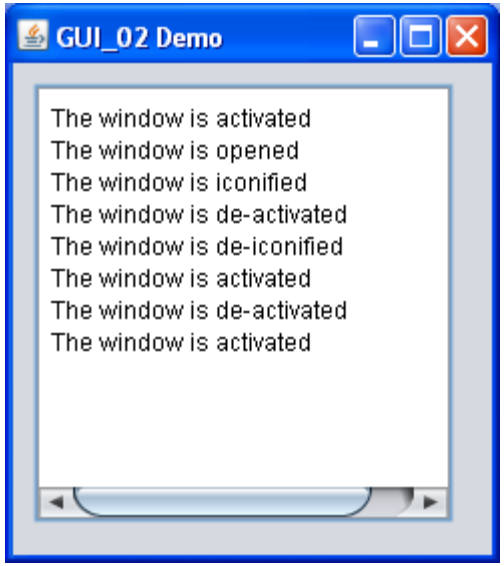
Frame Properties:
Layout: null
Title: GUI_02 Demo

Demo 2: Window Events

```
import javax.swing.JOptionPane;
public class GUI_02 extends javax.swing.JFrame {
    /** Creates new form GUI_02 ...3 lines */
    public GUI_02() {
        initComponents();
        this.setSize(300, 200);
    }

    private void formWindowActivated(java.awt.event.WindowEvent evt) {
        txt1.append("The window is activated\n");
    }
    private void formWindowClosed(java.awt.event.WindowEvent evt) {
        txt1.append("The window is closed\n");
        JOptionPane.showMessageDialog(this, "The window is closed!");
    }
    private void formWindowClosing(java.awt.event.WindowEvent evt) {
        txt1.append("The window is closing\n");
        JOptionPane.showMessageDialog(this, "The window is closing!");
    }
    private void formWindowDeactivated(java.awt.event.WindowEvent evt) {
        txt1.append("The window is de-activated\n");
    }
    private void formWindowDeiconified(java.awt.event.WindowEvent evt) {
        txt1.append("The window is de-iconified\n");
    }
```

Demo 2: Window Events

```
private void formWindowIconified(java.awt.event.WindowEvent evt) {
    txt1.append("The window is iconified\n");
}
private void formWindowOpened(java.awt.event.WindowEvent evt) {
    txt1.append("The window is opened\n");
}
```



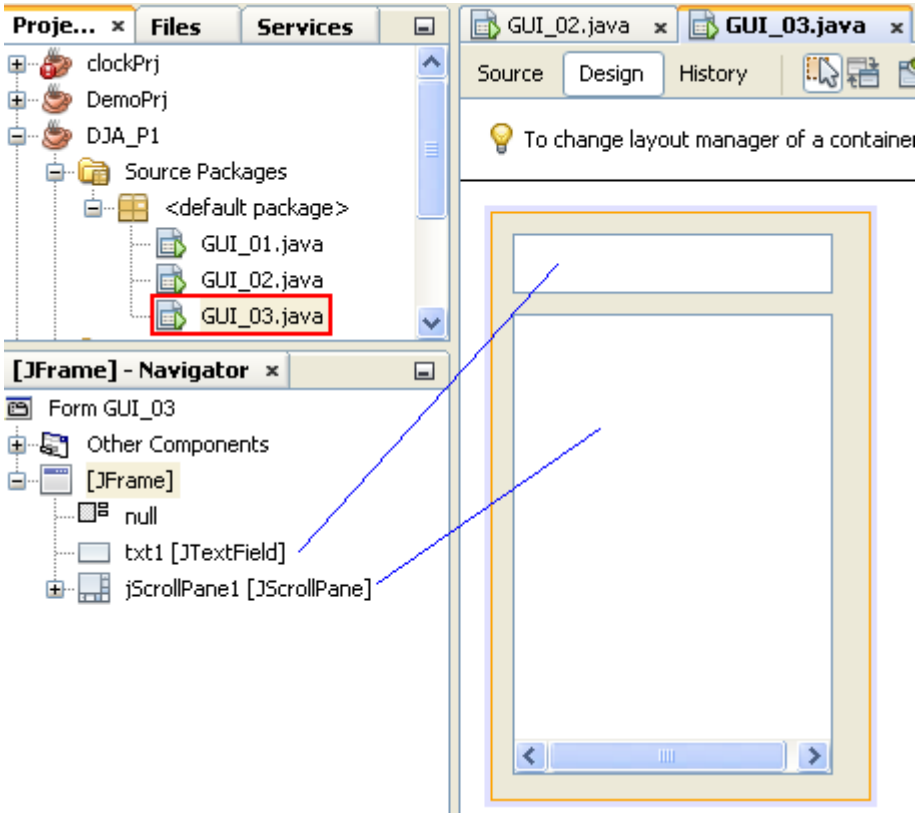
Run the program:

Use the mouse, you will activate/iconify/de-iconify/close the program. You will see results presented in the textarea when the window (frame) is impacted.

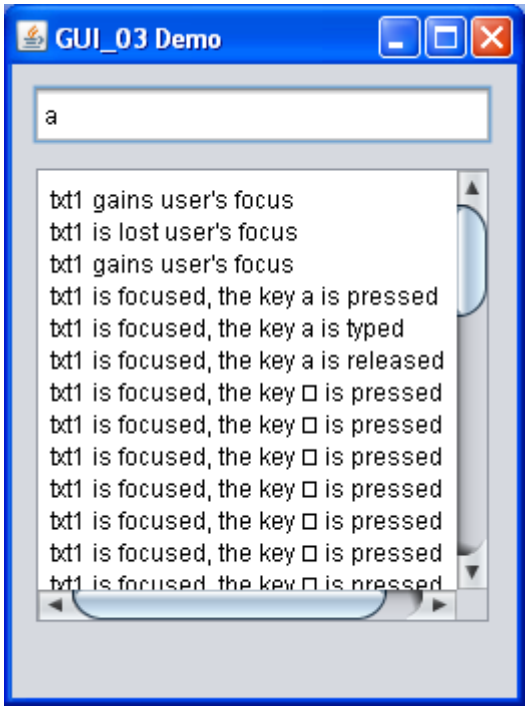
Questions:

Give your opinion about window events.

Demo 3: Low Level Events



Frame Properties:
Layout: null
Title: GUI_03 Demo



After you complete code, run the program. Click the mouse on textbox and/or textarea, press some key on the keyboard. The result will like as the GUI in this slide

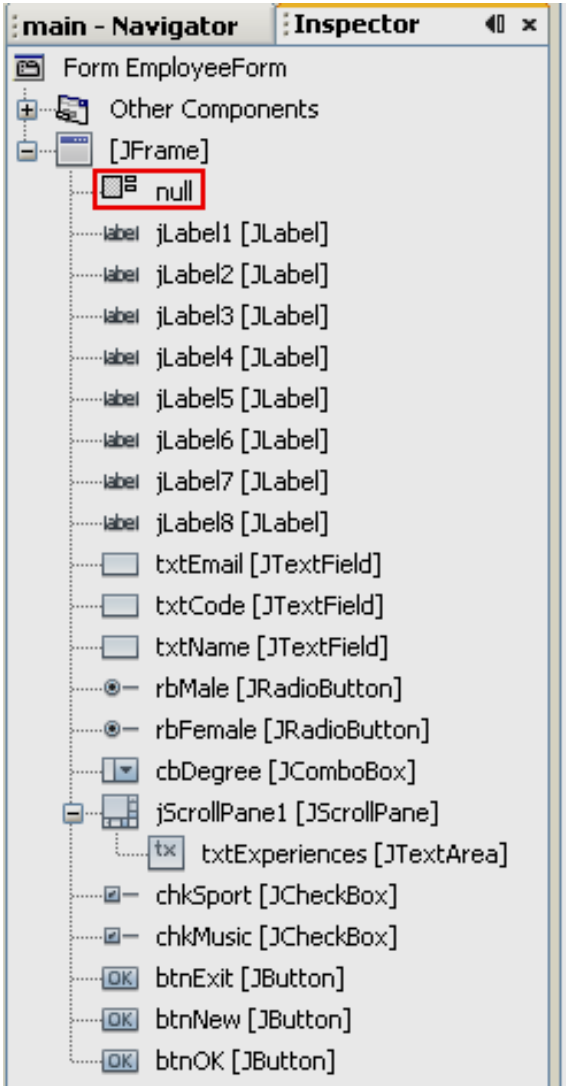
Questions:
Give your opinion about low-level events.

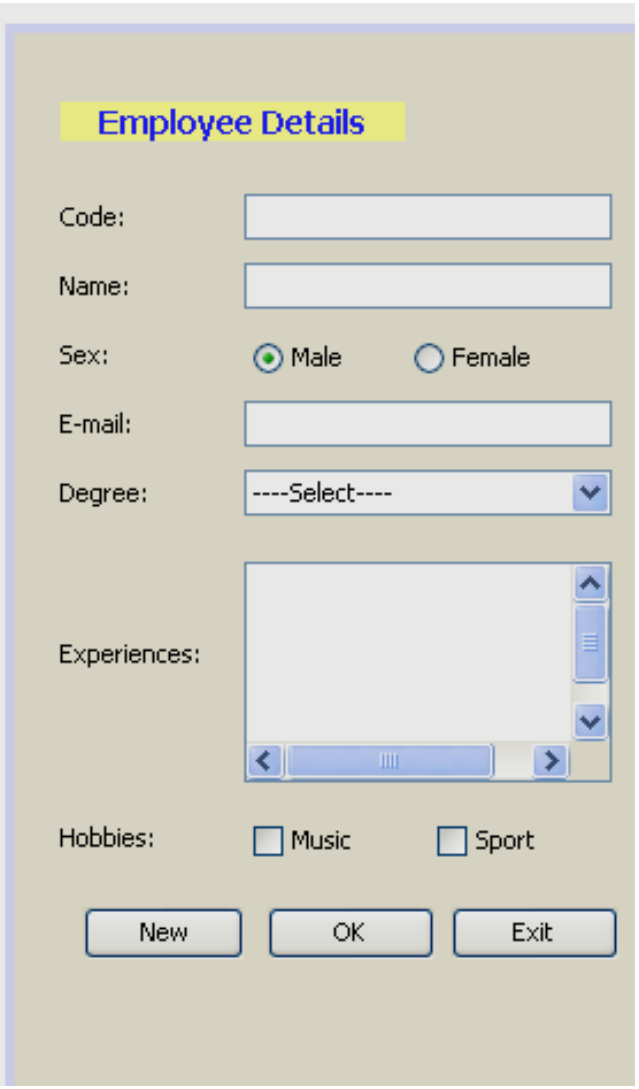
Demo 3: Low Level Events

```
public GUI_03() {
    initComponents();
    this.setSize(260, 350);
}

private void txt1FocusGained(java.awt.event.FocusEvent evt) {
    jTextArea1.append("txt1 gains user's focus\n");
}
private void txt1FocusLost(java.awt.event.FocusEvent evt) {
    jTextArea1.append("txt1 is lost user's focus\n");
}
private void txt1KeyPressed(java.awt.event.KeyEvent evt) {
    char c= evt.getKeyChar();
    jTextArea1.append("txt1 is focused, the key " + c + " is pressed\n");
}
private void txt1KeyReleased(java.awt.event.KeyEvent evt) {
    char c= evt.getKeyChar();
    jTextArea1.append("txt1 is focused, the key " + c + " is released\n");
}
private void txt1KeyTyped(java.awt.event.KeyEvent evt) {
    char c= evt.getKeyChar();
    jTextArea1.append("txt1 is focused, the key " + c+ " is typed\n");
}
```

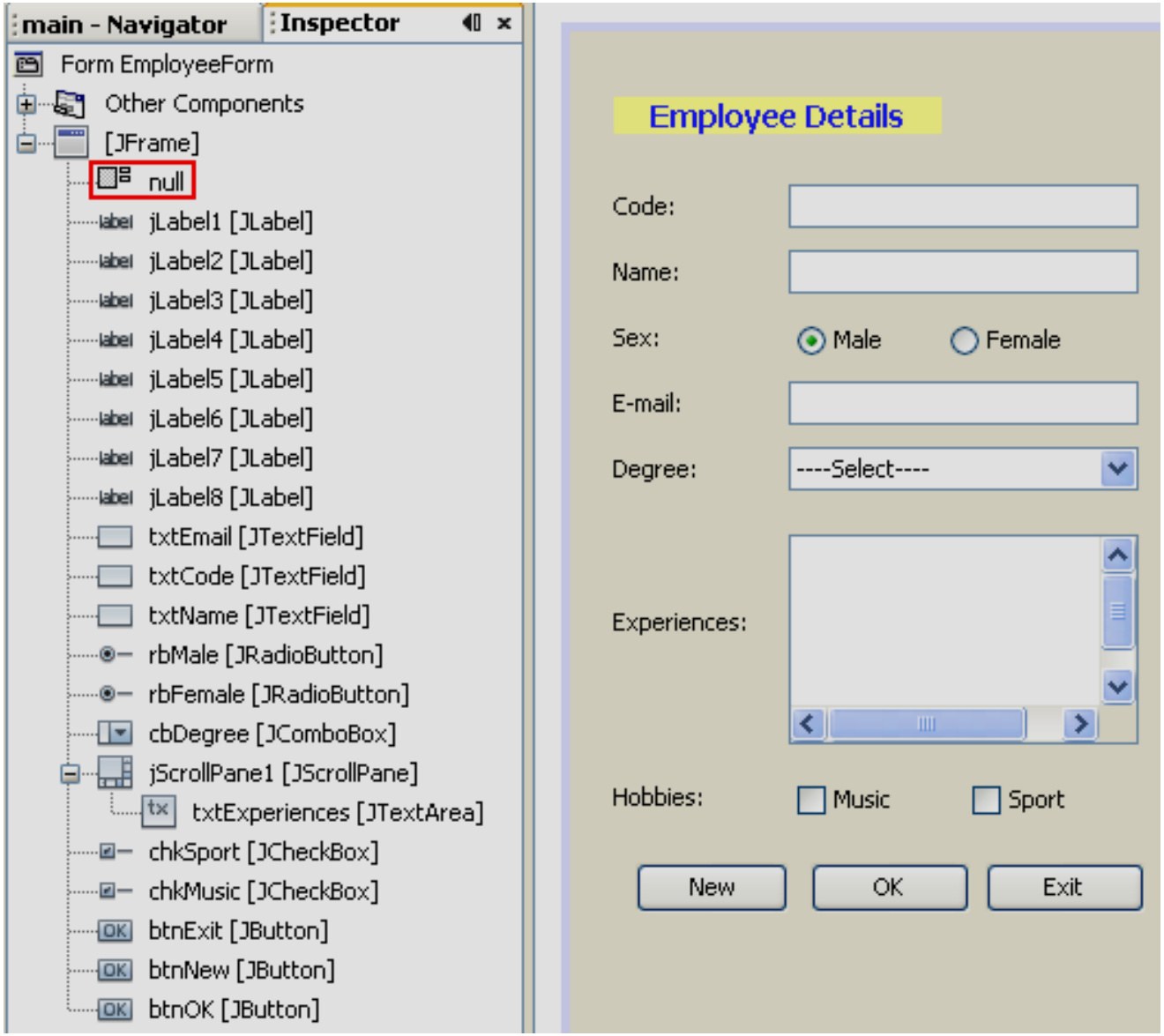
Demo 4: Ordinary Components

- Application for managing employees





Demo 4: Ordinary Components…



Demo 4: Ordinary Components…

```
import javax.swing.JOptionPane; // Class for System Dialog
public class EmployeeForm extends javax.swing.JFrame {
    /** Constructor-Creates new form EmployeeForm */
    public EmployeeForm() {
        initComponents();
        this.setSize(300,450);
    }
```

Exit

```
    private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }
```

New

```
private void btnNewActionPerformed(java.awt.event.ActionEvent evt) {
    // Clear add input component
    this.txtCode.setText("");
    this.txtName.setText("");
    this.rbMale.setSelected(true);
    this.rbFemale.setSelected(false);
    this.txtEmail.setText("");
    this.cbDegree.setSelectedIndex(0);
    this.txtExperiences.setText("");
    this.chkMusic.setSelected(false);
    this.chkSport.setSelected(false);
}
```

Demo 4: Ordinary Components…


```
// checking whether data enterd are valid or not
private boolean validData()
{   String S;
    // Checking code format: E000
    S= this.txtCode.getText().trim().toUpperCase();
    this.txtCode.setText(S);
    if (!S.matches("^E\\d{3}$"))
    {  JOptionPane.showMessageDialog(this," Code: E + 3 digits.");
       txtCode.requestFocus();
       return false;
    }
    // Name is required
    S= this.txtName.getText().trim().toUpperCase();
    this.txtName.setText(S);
    if (S.length()==0)
    {  JOptionPane.showMessageDialog(this," Name is required.");
       txtName.requestFocus();
       return false;
    }
    // Checking E-mail
    S= this.txtEmail.getText().trim();
    this.txtEmail.setText(S);
    if (!S.matches("^\\w+@\\w+[.]\\w+$"))
    {  JOptionPane.showMessageDialog(this," Email: anyname@any.any");
       txtEmail.requestFocus();
       return false;
    }
```

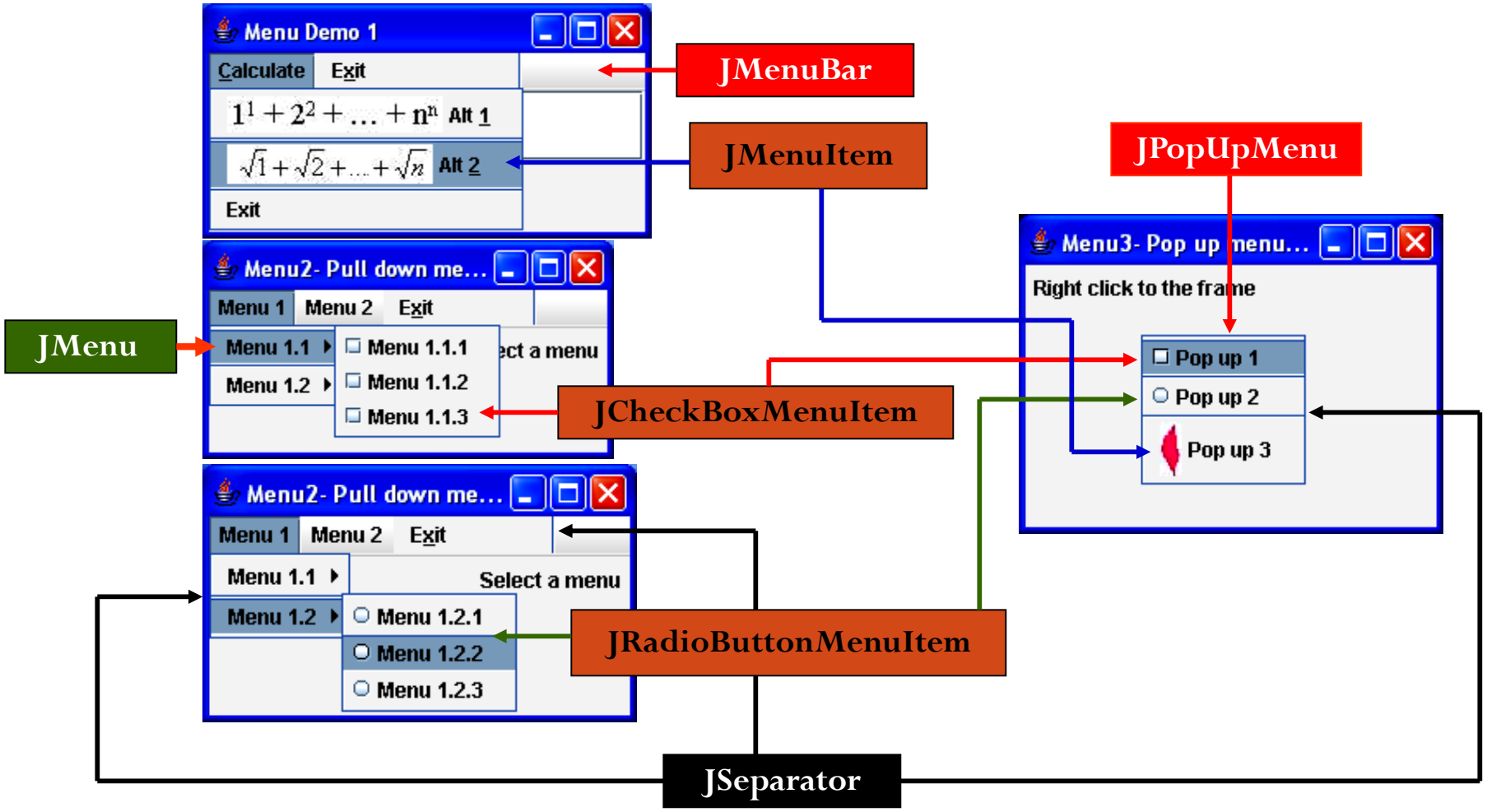
Demo 4: Ordinary Components…

```
        // Degree must be selected
        if (this.cbDegree.getSelectedIndex()<=0)
        {  JOptionPane.showMessageDialog(this," Degree must be selected.");
           this.cbDegree.requestFocus();
           return false;
        }
        // experience can not be required
        // Hobbies can not be required
        return true;
    }
    private void btnOKActionPerformed(java.awt.event.ActionEvent evt) {
        // Process data entered.
        if (validData())
        {  // Accessing data from input components
           String S="Welcome ";
           S+= this.txtCode.getText() + ", " +
               this.txtName.getText() + ", " +
               (this.rbMale.isSelected()? "Male " : "Female, ") +
               this.txtEmail.getText() + ", " +
               this.cbDegree.getSelectedItem() + ", " +
               this.txtExperiences.getText() + ", " +
               (this.chkMusic.isSelected()? " Music, " : "") +
               (this.chkSport.isSelected()? "Sport" : "") ;
           JOptionPane.showMessageDialog(this, S);
        }
    }
```

2- Menu

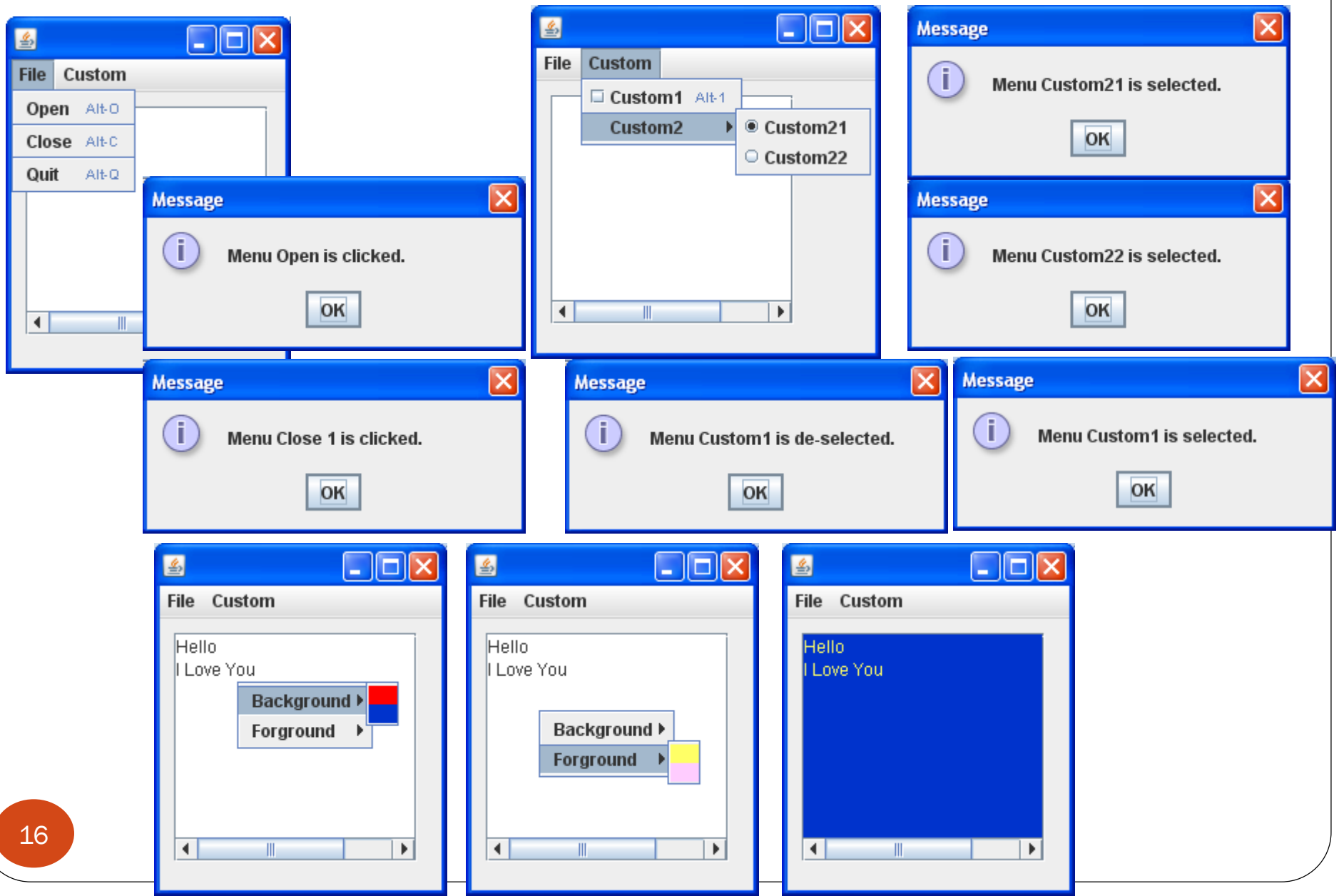
- Menu is a mean which allows user choose an function of a program at a time
- Main characteristic of menu: It needs not to be supply a wide screen area
- A menu should be created if the program supporting a lot of operations but we do not want to pay more screen area.

Menu...

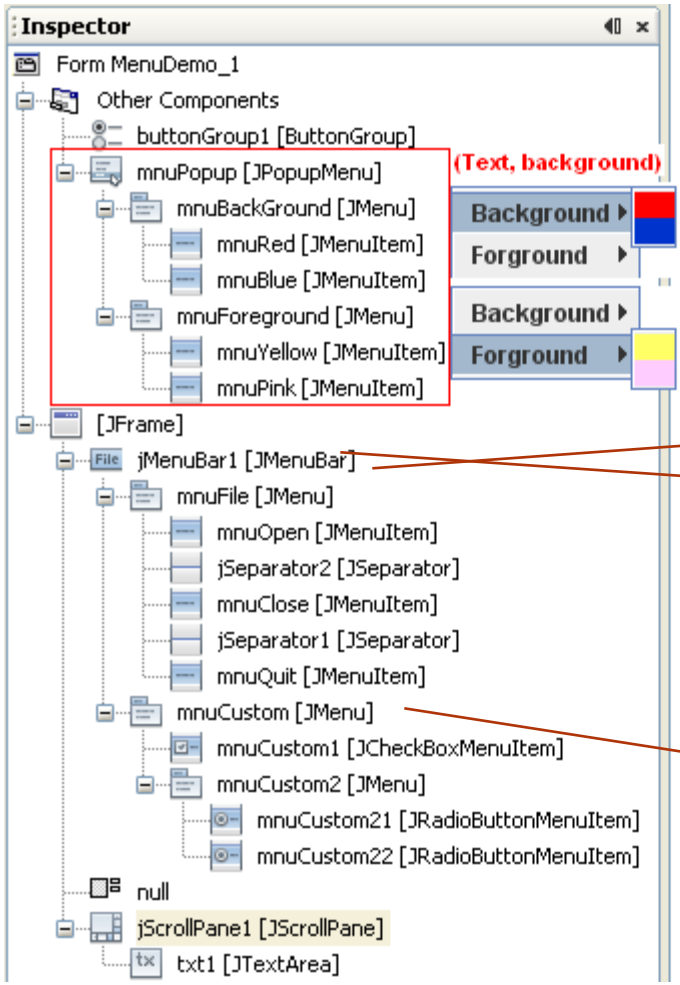


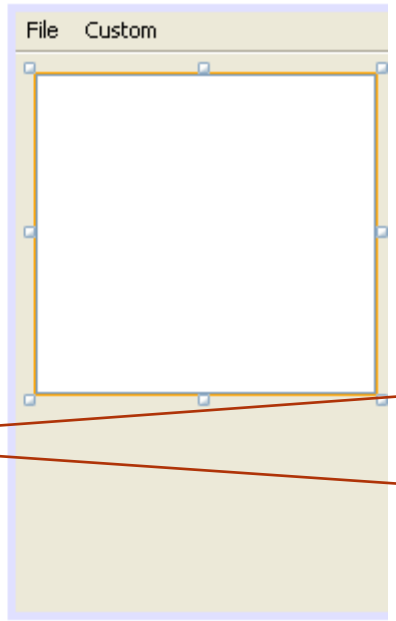
Menu Organization

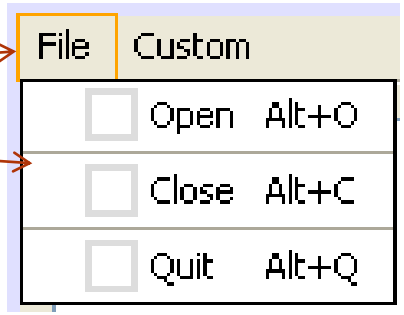
Demo 5: Menu

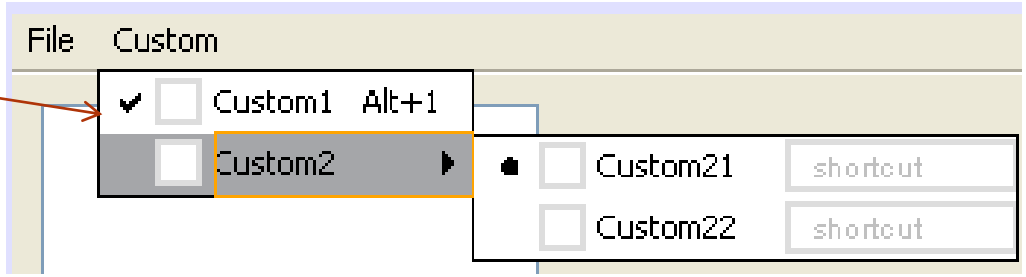

File
Custom
Open Alt-O
Close Alt-C
Quit Alt-Q
Message
Menu Open is clicked.
OK
File
Custom
Custom1 Alt-1
Custom2
Custom21
Custom22
Message
Menu Custom21 is selected.
OK
Message
Menu Custom22 is selected.
OK
Message
Menu Close 1 is clicked.
OK
Message
Menu Custom1 is de-selected.
OK
Message
Menu Custom1 is selected.
OK
File
Custom
Hello
I Love You
Background
Forground
File
Custom
Hello
I Love You
Background
Forground
File
Custom
Hello
I Love You

Demo 5: Menu…

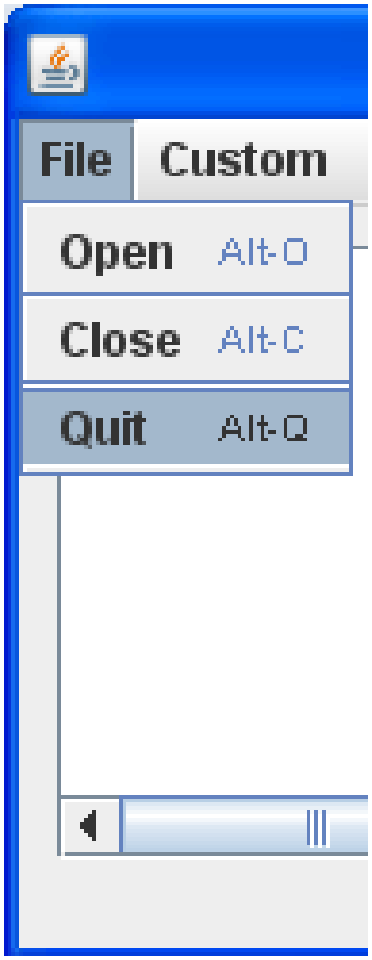

Inspector
Form MenuDemo_1
Other Components
buttonGroup1 [ButtonGroup]
mnuPopup [JPopupMenu]
(Text, background)
mnuBackGround [JMenu]
mnuRed [JMenuItem]
mnuBlue [JMenuItem]
mnuForeground [JMenu]
mnuYellow [JMenuItem]
mnuPink [JMenuItem]
Background
Forground
Background
Forground
[JFrame]
jMenuBar1 [JMenuBar]
mnuFile [JMenu]
mnuOpen [JMenuItem]
jSeparator2 [JSeparator]
mnuClose [JMenuItem]
jSeparator1 [JSeparator]
mnuQuit [JMenuItem]
mnuCustom [JMenu]
mnuCustom1 [JCheckBoxMenuItem]
mnuCustom2 [JMenu]
mnuCustom21 [JRadioButtonMenuItem]
mnuCustom22 [JRadioButtonMenuItem]
null
jScrollPane1 [JScrollPane]
txt1 [JTextArea]


File
Custom


File
Custom
Open Alt+O
Close Alt+C
Quit Alt+Q

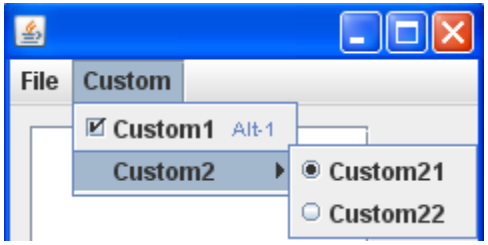

File
Custom
Custom1 Alt+1
Custom2
Custom21
shortcut
Custom22
shortcut

Demo 5: Menu




```
import javax.swing.JOptionPane; // for common dialog
public class MenuDemo_1 extends javax.swing.JFrame {

    /** Creates new form MenuDemo_1 */
    public MenuDemo_1() {
        initComponents();
        this.setSize(200,250);
    }

private void mnuOpenActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JOptionPane.showMessageDialog(this, "Menu Open is clicked.");
}

private void mnuCloseActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JOptionPane.showMessageDialog(this, "Menu Close 1 is clicked.");
}

private void mnuQuitActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}
```

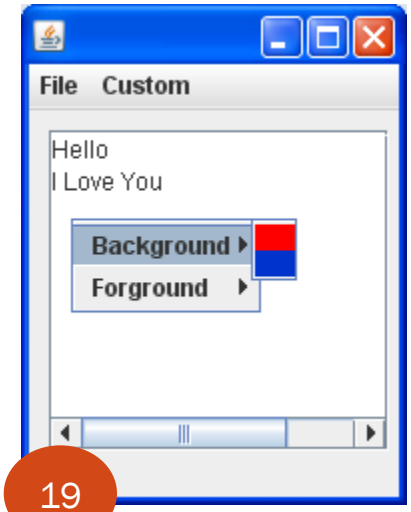
Demo 5: Menu…




```
private void mnuCustom1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (this.mnuCustom1.isSelected())
        JOptionPane.showMessageDialog(this, "Menu Custom1 is selected.");
    else
        JOptionPane.showMessageDialog(this, "Menu Custom1 is de-selected.");
}
```

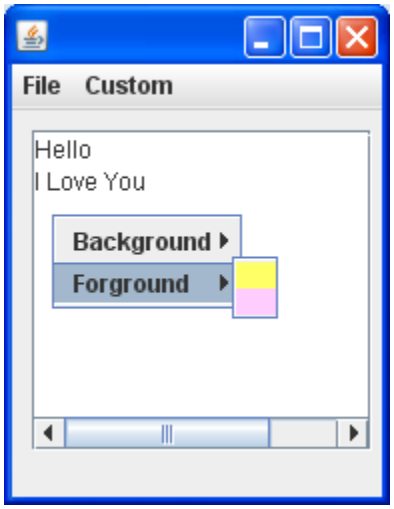
```
private void mnuCustom21ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(this.mnuCustom21.isSelected())
        JOptionPane.showMessageDialog(this, "Menu Custom21 is selected.");
}
```

```
private void mnuCustom22ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(this.mnuCustom22.isSelected())
        JOptionPane.showMessageDialog(this, "Menu Custom22 is selected.");
}
```




```
private void mnuRedActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
   this.txt1.setBackground(mnuRed.getBackground());
}
private void mnuBlueActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.txt1.setBackground(mnuBlue.getBackground());
}
```

Demo 5: menu…




```
private void mnuYellowActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.txt1.setForeground(mnuYellow.getBackground());
}
private void mnuPinkActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.txt1.setForeground(mnuPink.getBackground());
}

private void txt1MouseReleased(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    if (evt.isPopupTrigger())
        this.mnuPopup.show(this.txt1, evt.getX(),evt.getY());
}
```

Summary

- Events in Java
- Working with Menu
- Demonstrations

Thank You