

Object Oriented Programming with Java

COMS M0103
2004

Lecture 15 -- Non-Text Swing Components

Tim Kovacs
kovacs@cs.bris.ac.uk

Frederik Vercauteren
frederik@cs.bris.ac.uk

Overview

This lecture is a quick survey of some non-text Swing components such as buttons, sliders, menus and toolbars

Fragments of source code are given here

Complete listings are on the web

You will need to compare them on your own to understand them fully

2

1 Check Boxes

- For Yes/No input
- Come with labels (set with constructor)
- Clicking generates an `ActionEvent`



Partial API:

```
JCheckBox(String label)
JCheckBox(String label, boolean state)
boolean isSelected()
void setSelected(boolean state)
```

3

1 Check Boxes

`CheckBoxTest.java` has 2 boxes sharing same action listener:

```
ActionListener listener = ...;
bold.addActionListener(listener);
italic.addActionListener(listener);
```



- Same method handles both boxes:
- ```
public void actionPerformed(ActionEvent e) {
 int mode = 0;
 if (bold.isSelected()) mode += Font.BOLD;
 if (italic.isSelected()) mode += Font.ITALIC;
 label.setFont(new Font("Serif", mode,
 FONTSIZE));
}
```

4

## 2 Radio Buttons

- For selecting one of a group of options
- Pushing 1 button turns all others in group off
- Use 1 `ButtonGroup` object for each group
- Put `JRadioButtons` inside `ButtonGroup`

```
ButtonGroup group = new ButtonGroup();
JRadioButton smallButton = new JRadioButton("Small", false);
group.add(smallButton);
JRadioButton mediumButton = new JRadioButton("Medium", false);
group.add(mediumButton);
```

where false = initially off



5

## 2 Radio Buttons

Use `ActionListener` (like all buttons)

Each button makes an event handler (inside `addRadioButton`):

```
ActionListener listener = new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
 // 'size' is argument to addRadioButton
 label.setFont(new Font("Serif", Font.PLAIN, size));
 }
};
```

We could have used 1 handler for all buttons:

```
if (smallButton.isSelected()) size = 8;
else if (mediumButton.isSelected()) size = 12; ...
```

But that means checking many buttons when only 1 is on.

See `RadioButtonTest.java`.

6

### 3 Borders

- Useful for visually grouping components
- Normal use: put border around a panel and add components to panel

1. Call BorderFactory to create a border object.
2. Add border to component using setBorder

```
Border titled =
BorderFactory.createTitledBorder(etched, "my title");
panel.setBorder(titled);
```

BorderTest.java

- 2 panels, each with a border
- Top panel contains buttons
- Bottom panel demonstrates border styles



7

### 4 Combo Boxes

- For selecting 1 of a set of options (like Radio Buttons)
- Use instead of Radio Buttons when there's more than a few choices
- Takes up less space on screen than many Radio Buttons

Enable editing of text:

```
myComboBox.setEditable(true);
```

Editing only changes text currently shown - does not update items

Adding / removing items:

```
myComboBox.addItem("Option A"); // add to end of list
myComboBox.insertItemAt("Option A", 0);
myComboBox.removeItem("Option A");
myComboBox.removeItemAt(0);
myComboBox.removeAllItems();
```

You can add any subclass of Object -- will display using toString()

8

### 4 Combo Boxes

- Selecting an item generates an ActionEvent
- Use getSelectedItem() to get item
- Need to cast this item back to its type

```
public void actionPerformed(ActionEvent event)
{
 label.setFont(new Font(
 (String)faceCombo.getSelectedItem(),
 Font.PLAIN,
 DEFAULT_SIZE));
}
```

See ComboBoxTest.java



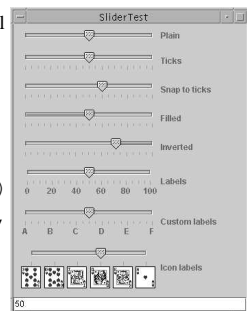
9

### 5 Sliders

- Select number using a sliding control
- See SliderTest.java

Constructors:

```
JSlider()
JSlider(int direction)
JSlider(int min, int max)
JSlider(int min, int max,
 int initialValue)
```



Where direction is

SwingConstants.HORIZONTAL or SwingConstants.VERTICAL

10

### 5 Sliders

Ticks can be added:

```
slider.setMajorTickSpacing(20);
slider.setMinorTickSpacing(5);
slider.setPaintTicks(true);
```

Make slider snap to nearest tick:

```
slider.setSnapToTicks(true);
```

Major ticks can be labelled with numbers:

```
slider.setPaintLabels(true);
```

Using Metal look and feel you can show how far slider is filled:

```
slider.putClientProperty("JSlider.isFilled",
 Boolean.TRUE);
```

11

### 5 Sliders

Reverse direction of slider:

```
slider.setInverted(true);
```

You can use strings or icons to label ticks

- Make a hashtable
- Fill with Integer/Component pairs

```
Hashtable labelTable = new Hashtable();
labelTable.put(new Integer(0), new JLabel("A"));
labelTable.put(new Integer(20), new JLabel("B"));
...
slider.setLabelTable(labelTable);
```

12

## 5 Sliders and Events

- When slider moves a `ChangeEvent` occurs
- Use `addChangeListener` to register object which implements `ChangeListener` interface
- Only method: `stateChanged()`

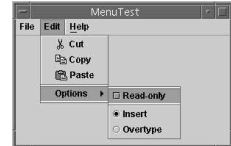
```
public void stateChanged(ChangeEvent event) {
 JSlider slider = (JSlider)event.getSource();
 int value = slider.getValue();
 ...
}
```

13

## 6 Menus

- A *menu bar* holds menu objects
- Menu objects hold *menu items*, separators and *submenus*

See `MenuTest.java` for examples



1. Add menu bars to any container

- but normally the top of a frame:

```
JMenuBar menuBar = new JMenuBar();
frame.setJMenuBar(menuBar);
```

2. Add menu objects to menu bar:

```
JMenuItem editMenu = new JMenuItem("Edit");
menuBar.add(editMenu);
```

14

## 6 Menus

3. Add items to menu object:

```
JMenuItem pasteItem = new JMenuItem("Paste");
editMenu.add(pasteItem);
editMenu.addSeparator();
...
```

Submenus: add a menu object to a menu object:

```
JMenuItem optionMenu = new JMenuItem("Options");
editMenu.add(optionMenu);
```

15

## 6 Menus and Event Handling

- Menu item selection triggers action events
  - Register an action listener for each menu item
- ```
ActionListener listener = ...;
myMenuItem.addActionListener(listener);
```

16

6 Menus – A Shortcut

Long way to create a menu item and add a listener to it:

```
JMenuItem pasteItem = new JMenuItem("Paste");
editMenu.add(pasteItem);
pasteItem.addActionListener(listener);
```

Shortcut:

- `JMenuItem.add(String)` returns a `JMenuItem`
 - You can create an item from a string and add a listener:
- ```
JMenuItem pasteItem = editMenu.add("Paste");
pasteItem.addActionListener(listener);
```

17

## 6 Partial Menu API

From `javax.swing.JMenu`:

```
JMenu(String label)
JMenuItem add(JMenuItem item)
JMenuItem add(String label)
JMenuItem insert(JMenuItem menu, int
 index)
void insertSeparator(int index)
void remove(int index)
void remove(JMenuItem item)
```

18

## 6 Icons in Menu Items

- Menu items are much like buttons
- You can have text, an icon, or both

```
JMenuItem(String)
JMenuItem(Icon)
JMenuItem(String, Icon)
```

E.g. `JMenuItem cutItem = new JMenuItem("Cut", new ImageIcon("cut.gif"));`

Text defaults to right of icon. To change:

```
cutItem.setHorizontalTextPosition(SwingConstants.LEFT);
```

19

## 6 Check Box and Radio Box Menu Items

Check Box menu items are just like any other menu item:

```
JCheckBoxMenuItem readonlyItem = new
 JCheckBoxMenuItem("Read-only");
optionsMenu.add(readonlyItem);
```

Radio button menu items belong to a button group:

```
ButtonGroup group = new ButtonGroup();
JRadioButtonMenuItem insertItem = new
 JRadioButtonMenuItem("Insert");
insertItem.setSelected(true); // button initially on
JRadioButtonMenuItem overtypeItem = new
 JRadioButtonMenuItem("Overtype");
group.add(insertItem);
group.add(overtypeItem);
optionsMenu.add(insertItem);
optionsMenu.add(overtypeItem);
```

20

## 6 Pop-up Menus

- A menu that is not attached to a menu bar

Pop-up menus have no title:

```
JPopupMenu popup = new JPopupMenu();
```

Add items as normal

Unlike menu bars, pop-ups must be told to show themselves:

```
popup.show(panel, x, y);
```

To pop-up where the mouse clicks:

1. Register a mouse listener

2. Add an event handler such as:

```
public void mousePressed(MouseEvent event) {
 if (event.isPopupTrigger())
 popup.show(event.getComponent(),
 event.getX(), event.getY());
}
```

21

## 6 Keyboard Mnemonics and Accelerators

- Mnemonics let you select items from an open menu using keys
- Accelerators let you select without opening a menu

Specify mnemonics by adding a char to constructor:

```
JMenuItem cutItem = new JMenuItem("Cut", 'T');
```

- Now the cut item can be selected by pressing T key.

- T in Cut gets underlined

Use `setAccelerator` to add accelerators to menu items:

```
openItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O,
 InputEvent.CTRL_MASK));
```

- Now pressing CTRL-O triggers action event as if `openItem` had been selected

- Note that menu does not actually appear on screen - but event occurs

22

## 6 Enabling and Disabling Menu Items

Sometimes a menu item is not appropriate

- E.g. user can't save document if none is open
- Better to disable 'save' item than remove it
- Disabled items are drawn gray (not black)

E.g.

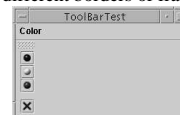
```
saveItem.setEnabled(false);
```

23

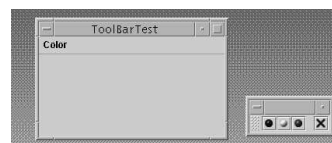
## 7 Tool Bars

A bar with buttons for most commonly used commands

Can be dragged to different borders of frame



User can detach it from frame



24

## 7 Tool Bars

Add components to tool bar

```
JToolBar bar = new JToolBar();
bar.add(blueButton);
bar.addSeparator();
```

Then add bar to a container:

```
contentPane.add(bar, BorderLayout.NORTH);
```

You can specify a title for floating toolbars:

```
bar = new JToolBar(titleString);
```

Bar defaults horizontal. Set vertical in constructor:

```
bar = new JToolBar(SwingConstants.VERTICAL);
```

See <http://developer.java.sun.com/developer/techDocs/hi/repository>  
for nice buttons

25

## 7 Tool Tips

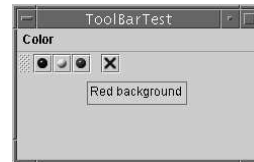
Tool bars with many buttons can be confusing to novices

Solution: add tool tips:

- When mouse rests on button, text appears saying what it does

Add tool tip to any JComponent using `setToolTipText`:

```
exitButton.setToolTipText("Exit");
```



See `ToolBarTest.java` for tool bar and tool tip examples

26

## Test Yourself

Borders

Check Boxes

Combo Boxes

Menus

- Menu Bar
- Menu Item
- Menu
- Menu Separator
- Submenu
- Pop-up menu
- Mnemonics
- Accelerators

Radio Buttons

Sliders

Tool Bars

Tool Tips

27