JAVA NOTES

GRAPHICAL USER INTERFACES

Terry Marris July 2001

7 RADIO BUTTONS

7.1 LEARNING OUTCOMES

By the end of this lesson the student should be able to

- understand and use radio buttons

7.2 INTRODUCTION

Only one button at a time can be selected in a radio button group.

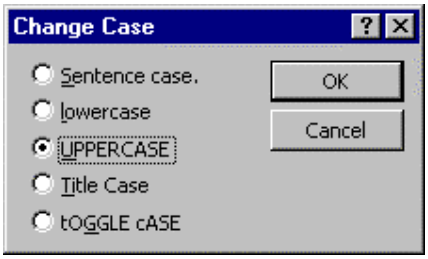


Figure 7.1 *Radio Buttons*

In this *Change Case* frame, from the Microsoft Word Format Change Case option, only one of *Sentence case*, *lowercase*, *UPPERCASE*, *Title Case* and *tOGGLE cASE* may be selected at any one time.

What we are aiming for is shown in Figure 7.2 below.

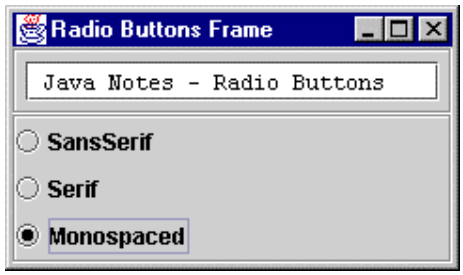


Figure 7.2 *Font Selection*

The user selects a font - an example of the font is shown in the text box. Here the example text, *Java Notes - Radio Buttons* is shown in the Monospaced font.

7.3 RADIO BUTTONS

We create a *JRadioButton* instance.

```
ButtonGroup buttonGroup = new ButtonGroup();
JRadioButton sansSerifButton = new JRadioButton("Sans Serif");
JRadioButton serifButton = new JRadioButton("Serif");
```

We add it to a *RadioGroup*, which ensures that only one button at a time may be selected.

```
    buttonGroup.add(sansSerifButton, true);
    buttonGroup.add(serifButton, false);
```

The *true* argument specifies which button is selected initially.

We add the button to the panel in the usual way.

```
    add(sansSerifButton, "North");
```

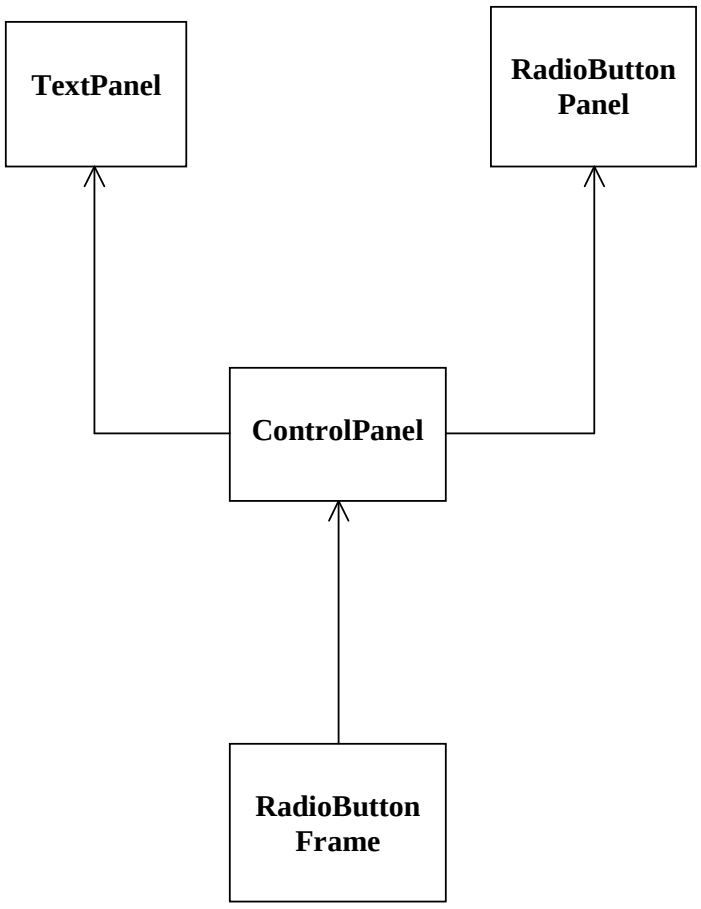
We register this panel to be the action listener for a button.

```
    radioButtonPanel.getSansSerifButton().
                     addActionListener(this);
```

We find out which button was pressed, and take some action.

```
  public void actionPerformed(ActionEvent e)
  {
    Object source = e.getSource();
    if (source == radioButtonPanel.getSansSerifButton())
      textPanel.setTextFont("SansSerif");
    else if (source == radioButtonPanel.getSerifButton())
      textPanel.setTextFont("Serif");
    else if (source == radioButtonPanel.getMonospacedButton())
      textPanel.setTextFont("Monospaced");
    repaint();
  }
```

7.4 CLASS DIAGRAM



A radio button frame has a control panel. A control panel has text panel and a radio button panel.

7.5 RADIO BUTTONS FRAME

The complete code fore the *RadioButtonsFrame* program is shown below.

```
/* RadioButtonsFrame.java
   Terry Marris 1 July 2001
*/

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;


class TextPanel extends JPanel {
  private JTextField text;

  public TextPanel()
  {
    Border etched = BorderFactory.createEtchedBorder();
    setBorder(etched);

    text = new JTextField(" Java Notes - Radio Buttons ", 20);
    add(text);
  }


  public void setTextFont(String fontName)
  {
    text.setFont(new Font(fontName, Font.PLAIN, 12));
    repaint();
  }
}
```

Exercise: Explain each line of the *TextPanel* class shown above.

```
class RadioButtonPanel extends JPanel {
  private ButtonGroup buttonGroup;
  private JRadioButton sansSerifButton;
  private JRadioButton serifButton;
  private JRadioButton monospacedButton;

  public RadioButtonPanel()
  {
    Border etched = BorderFactory.createEtchedBorder();
    setBorder(etched);

    setLayout(new BorderLayout());

    ButtonGroup buttonGroup = new ButtonGroup();

    sansSerifButton = new JRadioButton("SansSerif", true);
    buttonGroup.add(sansSerifButton);
    add(sansSerifButton, "North");

    serifButton = new JRadioButton("Serif", false);
    buttonGroup.add(serifButton);
    add(serifButton, "Center");

    monospacedButton = new JRadioButton("Monospaced", false);
    buttonGroup.add(monospacedButton);
    add(monospacedButton, "South");
  }


  public JRadioButton getSansSerifButton()
  {
    return sansSerifButton;
  }


  public JRadioButton getSerifButton()
  {
    return serifButton;
  }


  public JRadioButton getMonospacedButton()
  {
    return monospacedButton;
  }
}
```

Exercise: Explain each line of the *RadioButtonPanel* class shown above.

```
class ControlPanel extends JPanel implements ActionListener {
  private TextPanel textPanel;
  private RadioButtonPanel radioButtonPanel;

  public ControlPanel()
  {
    setLayout(new BorderLayout());

    textPanel = new TextPanel();
    add(textPanel, "North");

    radioButtonPanel = new RadioButtonPanel();
    add(radioButtonPanel, "South");

    radioButtonPanel.getSansSerifButton().
                     addActionListener(this);
    radioButtonPanel.getSerifButton().addActionListener(this);
    radioButtonPanel.getMonospacedButton().
                     addActionListener(this);
  }


  public void actionPerformed(ActionEvent e)
  {
    Object source = e.getSource();
    if (source == radioButtonPanel.getSansSerifButton())
      textPanel.setTextFont("SansSerif");
    else if (source == radioButtonPanel.getSerifButton())
      textPanel.setTextFont("Serif");
    else if (source == radioButtonPanel.getMonospacedButton())
      textPanel.setTextFont("Monospaced");
    repaint();
  }
}
```

Exercise: Explain each line of the *ControlPanel* class shown above.

```
public class RadioButtonsFrame extends JFrame {
  public RadioButtonsFrame()
  {
    setTitle("Radio Buttons Frame");

    Toolkit tk = Toolkit.getDefaultToolkit();
    Dimension screen = tk.getScreenSize();
    setLocation(screen.width/4, screen.height/4);

    addWindowListener(new WindowAdapter()
      {
        public void windowClosing(WindowEvent e)
        {
          System.exit(0);
        }
      });

    Container contentPane = getContentPane();
    contentPane.add(new ControlPanel());

    pack();
  }


  public static void main(String[] s)
  {
    JFrame aFrame = new RadioButtonsFrame();
    aFrame.show();
  }
}
```

Exercise: Explain each line of the *RadioButtonsFrame* class shown above.

7.6 DOCUMENTATION

javax.swing.ButtonGroup

This class is used to create a set of buttons with just one button turned on at any one time. Turning one button on turns off all the other buttons in the group. A button group can be used with sets of JButton, JRadioButton or JRadioButtonMenu objects.

There is no way to turn all the buttons off. To give the appearance of "none selected" add an invisible button to the group and then programmatically select that button to turn off all the displayed radio buttons.

Constructors	
ButtonGroup()	intialises a newly created, empty button group.

Methods		
void	add(AbstractButton b)	adds the given button to the group.

javax.swingJRadioButton

Constructors	
JRadioButton(String text)	initialises a new button which is unselected and has the given text as its label.
JRadioButton(String text, boolean selected)	initialises a new button with the given selection state and the giventext as its label.

setVisible() is defined in *javax.swing.JComponent*, the super class of *javax.swing.AbstractButton*.

7.7 FURTHER READING

HORSTMANN & CORNELL *Core Java 2 Volume 1* pp 411
www.java.sun.com/docs/books/tutorial

7.8 REVIEW

7.9 EXERCISES

1 Write and test a program that produces the following frame. The example text should change according to the button pressed.