25

GUI Components: Part 2, Solutions

Objectives

In this chapter you'll learn:

- To create and manipulate sliders, menus, pop-up menus and windows.

- To programatically change the look-and-feel of a GUI, using Swing's pluggable look-and-feel.

- To create a multiple-document interface with `JDesktopPane` and `JInternalFrame`.

- To use additional layout managers.

Self-Review Exercises

25.1   Fill in the blanks in each of the following statements:
   a) The _____ class is used to create a menu object.
   ANS: JMenu.
   b) The _____ method of class JMenu places a separator bar in a menu.
   ANS: addSeparator.
   c) JSlider events are handled by the _____ method of interface _____.
   ANS: stateChanged, ChangeListener.
   d) The GridBagConstraints instance variable _____ is set to CENTER by default.
   ANS: anchor.

25.2   State whether each of the following is *true* or *false*. If *false*, explain why.
   a) When the programmer creates a JFrame, a minimum of one menu must be created and
        added to the JFrame.
   ANS: False. A JFrame does not require any menus.
   b) The variable fill belongs to the GridBagLayout class.
   ANS: False. The variable fill belongs to the GridBagConstraints class.
   c) Drawing on a GUI component is performed with respect to the (0, 0) upper-left corner
        coordinate of the component.
   ANS: True.
   d) The default layout for a Box is BoxLayout.
   ANS: True.

25.3   Find the error(s) in each of the following and explain how to correct the error(s).
   a) `JMenubar b;`
   ANS: JMenubar should be JMenuBar.
   b) `mySlider = JSlider( 1000, 222, 100, 450 );`
   ANS: The first argument to the constructor should be either SwingConstants.HORIZONTAL
        or SwingConstants.VERTICAL, and the keyword new must be used after the = operator.
        Also, the minimum value should be less than the maximum and the initial value
        should be in range.
   c) `gbc.fill = GridBagConstraints.NORTHWEST;   // set fill`
   ANS: The constant should be either BOTH, HORIZONTAL, VERTICAL or NONE.
   d) `// override to paint on a customized Swing component`
      ```
      public void paintcomponent( Graphics g )
      {
         g.drawString( "HELLO", 50, 50 );
      } // end method paintComponent
      ```
   ANS: paintcomponent should be paintComponent, and the method should call
        super.paintComponent(g) as its first statement.
   e) `// create a JFrame and display it`
      ```
      JFrame f = new JFrame( "A Window" );
      f.setVisible( true );
      ```
   ANS: The JFrame's setSize method must also be called to establish the size of the window.

Exercises

*NOTE: Solutions to the programming exercises are located in the **ch25solutions** folder.
Each exercise has its own folder named **ex25_##** where ## is a two-digit number represent-
ing the exercise number. For example, exercise 25.14's solution is located in the folder
ex25_14.*

25.4   Fill in the blanks in each of the following statements:
    a) A JMenuItem that is a JMenu is called a(n) _____ .
ANS: submenu.
    b) Method _____ attaches a JMenuBar to a JFrame.
ANS: setJMenuBar.
    c) Container class _____ has a default BoxLayout.
ANS: Box.
    d) A(n) _____ manages a set of child windows declared with class JInternalFrame.
ANS: JDesktopPane.

25.5   State whether each of the following is *true* or *false*. If *false*, explain why.
    a) Menus require a JMenuBar object so they can be attached to a JFrame.
ANS: True.
    b) BoxLayout is the default layout manager for a JFrame.
ANS: False. BorderLayout is the default layout manager for JFrame.
    c) JApplets can contain menus.
ANS: True.

25.6   Find the error(s) in each of the following. Explain how to correct the error(s).
    a) x.add(new JMenuItem("Submenu Color")); // create submenu
ANS: A JMenu is used to create a submenu, not a JMenuItem.
    b) container.setLayout(new GridbagLayout());
ANS: GridbagLayout should be GridBagLayout.

25.9   Explore the effects of varying the weightx and weighty values of the program in Fig 25.21. What happens when a slot has a nonzero weight but is not allowed to fill the whole area (i.e., the fill value is not BOTH)?
    ANS: When the weight is non-zero but the component is not allowed to fill in both directions, blank space is created around the component. The amount of space allocated for the component in the panel is the same as if it could expand to fill the space, but the component does not stretch in that direction.