

## Chapter 23 Menus, Toolbars, Dialogs, and Internal Frames

1. Use JMenuBar constructor.
2. You can add menu items into a menu item to create submenu items. To create a check box menu item, use JCheckBoxMenuItem. To create a radio button menu item, use JRadioButtonMenuItem.
3. Use the addSeparator method.
4. You can use the constructor JMenuItem(String, ImageIcon) to associate an icon with a menu item or use the setIcon method in the JMenuItem class to set an icon. To associate a menu with a keyboard mnemonics, use the setMnemonic method. To associate a menu with a keyboard accelerator, use setAccelerator(KeyStroke).
5. To create a popup menu, use JPopupMenu. To show a popup menu, use the show method of JPopupMenu. The popup menu usually contains the commands for an object.
6. Customarily, you display a popup menu by pointing to the object and clicking a certain mouse button, the so-called popup trigger. Popup triggers are system-dependent. In Windows, the popup menu is displayed when the right mouse button is released. In Motif, the popup menu is displayed when the third mouse button is pressed and held down.
7. By default, the layout manager for JToolBar is BorderLayout. However, you can change it if desired.
8. Use the add(Component) to add a component to a JToolBar. You can place a JToolBar into the content pane of a frame or an applet.
9. The Action interface establishes a protocol for processing the common action from buttons, menu items, and toolbar buttons. So you don't have to write separate code.
10. To add an Action object to a JToolBar or a JMenu, use the add(Action) method. To add it to a JButton, JRadioButton, and JCheckBox, you have to use their constructors with the Action object as the argument.
11. The JOptionPane class can be used to create four kinds of standard dialogs:
  - Message dialog: Shows a message and waits for the user to click OK.
  - Confirmation dialog: Shows a question and asks for confirmation, such as OK and Cancel.

- Input dialog: Shows a question and gets the user's input from a text field, combo box, or list.
- Option dialog: Shows a question and gets the user's answer from a set of options.

These dialogs are created using the static methods showXxxDialog.

12. To display a message dialog box, use the static showMessageDialog method in the JOptionPane class. The messageType is one of the following constants: JOptionPane.ERROR\_MESSAGE, JOptionPane.INFORMATION\_MESSAGE, JOptionPane.PLAIN\_MESSAGE, JOptionPane.WARNING\_MESSAGE, and JOptionPane.QUESTION\_MESSAGE. The button in a message dialog box is OK.
13. To display a confirmation dialog box, use the static method showConfirmDialog in the JOptionPane class. The optionType determines which buttons are displayed in the dialog. The possible values are JOptionPane.YES\_NO\_OPTION, JOptionPane.YES\_NO\_CANCEL\_OPTION, and JOptionPane.OK\_CANCEL\_OPTION.
14. Use JOptionPane.showInputDialog to create a dialog box to use text field as input. To create a dialog box using a combo box, use the showInputDialog method with an argument of an array of less than 20 elements. To create a dialog box using a list, use the showInputDialog method with an argument of an array of 20 or more elements.
15. To show an instance of JDialog, invoke its setVisible(true) method. A standard dialog box created using the static methods in JOptionPane is modal. An instance of JDialog box by default is not modal.
16. To show a JColorChooser dialog box, use its showDialog(Component, title, initialColor) method. A JColorChooser dialog box is modal. The showDialog method returns the selected color.
17. To show a JFileChooser dialog box, use its showOpenDialog method to display an Open dialog box, and use its showSaveDialog method to display a Save dialog box. A JFileChooser dialog box is modal. To obtain the selected file(s), use the methods getSelectedFile or getSelectedFiles. getSelectedFile() returns an instance of File. getSelectedDirectories() returns an instance of File[]. Use setCurrentDirectory(new File(".")) to set current directory as default directory for JFileChooser.
18.
  1. Use a JFrame or a JApplet to be the outer window.
  2. Create a JDesktopPane and add it to the content pane of a JFrame or JApplet. Usually, the JDesktopPane is added to the center of the content pane.

3. Create a JInternalFrame and add it to the JDesktopPane using the add method.
4. Use the setVisible(true) method to display the internal frame.