Chapter 30 MVC and Swing MVC Components

1. The model-view-controller (MVC) approach is a way of developing components by separating data storage and handling from the visual representation of the data. The component for storing and handling data, known as a model, contains the actual contents of the component. The component for presenting the data, known as a view, handles all essential component behaviors. It is the view that comes to mind when you think of the component. It does all the displaying of the components. The controller is a component that is usually responsible for obtaining data.

2. The JDK event delegation model provides a superior architecture for supporting MVC component development. The model can be implemented as a source with appropriate event and event listener registration methods. The view can be implemented as a listener. Thus, if data are changed in the model, the view will be notified. To enable the selection of the model from the view, simply add the model as a property in the view with a set method.

3. A common variation of the model-view-controller architecture is to combine the controller with the view. In this case, a view not only presents the data, but is also used as an interface to interact with the user and accept user input.

4. Yes. Each Swing GUI component (except some containers and dialog boxes such as JPanel, JsplitPane, JFileChooser, and JColorChooser) have a property named model. No. The type of the model property is dependent on the component. For JList, the model type is ListModel and JSpinner, the model type is SpinnerModel. Generally, the model interface is named XModel for the X component.

5. Yes in most cases and no in some rare cases. Most model interface has a default implementation class that is commonly named DefaultX, where X is its model interface name. For example, the default implementation class for ListModel is DefaultListModel.

6. The default model is SpinnerNumberModel if you don’t specify a model when creating a JSpinner.

7. The internal data structure for storing data in SpinnerListModel is java.util.Vector. A simple way to convert an array to a vector is to use the static method Arrays.asList to obtain a list, then use the list to construct a vector.

8. No. You have to add items to a list using a ListModel. You can display icons and custom GUI objects in list. List items cannot be edited. You can initialize data from JList constructor. To specify the maximum number of rows without scrolling, set the JList’s visibleRowCount property. To specify the height of a list cell, set the JList’s fixedCellHeight property. To specify the horizontal margin of list cells, set the JList’s fixedCellWidth property.
9. A simple way to create a list model is using the DefaultListModel class. To add items to a list model, use the various add method in the DefaultListModel. To remove items from a list model, use the various remove method in the DefaultListModel.

10. The three list-selection modes are single selection, single-interval selection, and multiple-interval selection. You can set the selection modes directly in an instance of JList using the setSelectionMode method. To obtain the selected item(s), use the getSelectedValue or getSelectedValues methods.

11. To create a custom list cell renderer, implement the ListCellRenderer interface and its getListCellRendererComponent method.

12. The handler for handling the ListSelectionEvent is valueChanged(ListSelectionEvent e).

13. Only a single item can be selected from a combo box. A combo box item can be edited. To specify the maximum number of visible rows in a combo box without scrolling, set the maximumRowCount method. There are no methods in JComboBox that use can use to specify the height of a combo box cell. To obtain the selected item in a combo box, use the getSelectedItem method.

14. To add or remove items from a combo box, you may use JComboBox’s add and remove methods, or use the add and remove methods from the ComboBoxModel.

15. The cell renderer for a combo box is the same as the renderer for a list, since they both implement ListCellRenderer.