Chapter 8 Advanced Swing Components

1. 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.

2. 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.

3. 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.

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

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

6. 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.

7. 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.

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

9. You can initialize a table using the constructor of JTable. You cannot specify the maximum number of visible rows in a table without scrolling. You can specify the height of a table cell using the setRowHeight method. You can specify the horizontal margin of table cells using the setIntercellSpacing method.

10. To modify table contents visually from the UI, the table cells must be editable with an associated editor for the cell. You must also save the change through the data model. To add or remove a row, you must use a TableModel. The DefaultTableModel class provides methods for adding and removing rows. To add a column, you may use the

addColumn method in DefaultTableModel or in DefaultTableColumnModel. To remove a column, you must use the removeTableColumn method from the DefaultTableColumnModel class.

11. JTable has the autoResizingMode property that can be used to auto resize a table column. Possible values are:

JTable.AUTO_RESIZE_OFF
JTable.AUTO_RESIZE_LAST_COLUMN
JTable.AUTO_RESIZE_SUBSEQUENT_COLUMNS
JTable.AUTO_RESIZE_NEXT_COLUMN
JTable.AUTO_RESIZE_ALL_COLUMNS

12. The properties to show grids, horizontal grids, and vertical grids are showGrid, showHorizontalGrid, and showVerticalGrid. The properties to specify the table row height, and vertical margin are rowHeight and rowMargin. There are no methods to set horizontal margins between the cells, because this type of margin is flexible.

13. By default, a cell object's string representation is displayed and the string can be edited as it was in a text field. JTable maintains a set of predefined renderers and editors, listed in Table 8.1, which can be specified to replace default string renderers and editors. You can create a custom renderer by extending the DefaultTableCellRenderer class.

14. To create a tree, simply use the constructor of JTtree. To specify the row height of a tree node, use the rowHeight property. To obtain the default tree model and tree selection model from an instance of JTree, use getModel and getSelectionModel methods.

15. To initialize data in a tree using TreeModel, you need to create nodes using the DefaultMutableTreeNode class, and set the root with the TreeModel (or DefaultTreeModel). To add a child to an instance of DefaultMutableTreeNode, use the add method.

16. To add a node from a tree, use the add method from an instance of DefaultMutableTreeNode in a parent. To remove a node from a tree, use the removeNodeFromParent method from TreeModel.

17. To obtain a selected tree node, use the getleadSelectedPath method to get the path, then get the node from the path using the getLastPathComponent method.