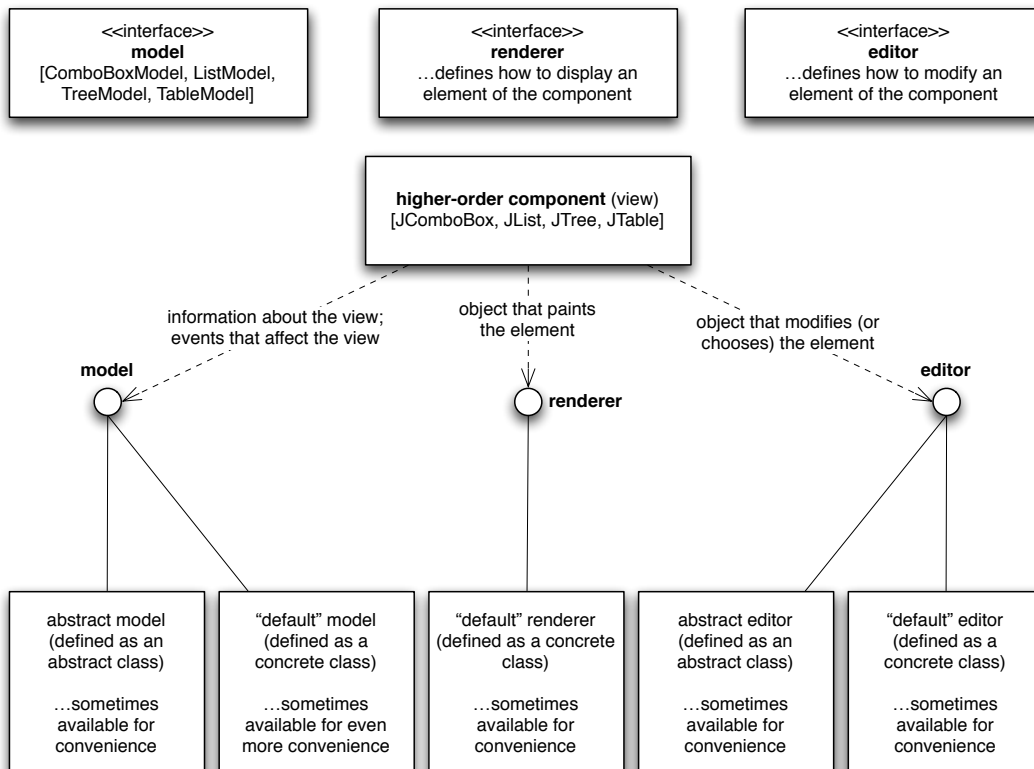


Higher-Order Components

- Some Swing components have inherently more complicated structure than others — I call these “higher-order” components
- For example, contrast JButton, JRadioButton, JCheckBox, JSlider, JProgressBar...
- ...with JComboBox, JList, JTree, and JTable
- The upshot: higher-order components are very generalized and customizable, but they require more work to truly do what you need



General Workflow: Models

- Define the information to be displayed
- Implement the model interface; if available, extend the abstract or default version
- Define how this information might change; have the models fire events as needed
- Plug the models into the Swing components
- Most of the time, you can use the Swing component classes as-is (i.e. no need to subclass JComboBox, JList, JTree, or JTable)

General Workflow: Renderers and Editors

- Make sure that “uncustomized” components work fine first
- Decide how information is to be displayed
- Extend or implement the appropriate renderer and/or editor
- Assign that renderer and/or editor to the Swing component
- Again, it is typically not necessary to extend the component itself; sufficient setter methods exist and can be called from the outside