Controls Summary

- Output controls are usually used only to display information to the user. Other controls respond to user actions in some way.
    - Label
    - ProgressBar
    - ProgressIndicator
    - Tooltip
- Command controls allow the user to initiate actions defined by the application.
    - Button
    - Hyperlink
- Boolean input controls allow the user to input boolean values.
    - CheckBox
    - RadioButton
    - ToggleButton
    - Tooltip
- Numerical input controls allow the user to input numeric values.
    - Slider
    - Spinner
- Text input controls allow the user to input String values.
    - TextField
    - PasswordField
    - TextArea
- Selection controls present a list from which the user can select input values.
    - ChoiceBox
    - ComboBox
    - Pagination
    - ColorPicker
    - DatePicker
    - ListView
    - TableView
    - TreeView
    - TreeTableView

Behavior Properties

There are two important types of control properties involved in implementing application behavior.

- *Event handler properties* determine how controls respond to user actions.
- *State properties* capture information entered or selected by users or, for output controls, information presented to the user.

Event handler properties are not relevant for all controls. When they are relevant it is most often the `onAction` property. This property can be set with a function to implement some kind of application behavior whenever the user performs a certain type of action with the control.

All of the controls except the ones whose name ends in "View" have state properties. The name of the property depends on the control. Those properties play different roles depending on whether the control is initiating behavior or is just part of the response.

If the control is initiating behavior, but not using an onAction handler, then an invalidation listener is registered with the state property of the control.

The controls whose name ends in "View" do not directly have a state property. Instead they have a `selectionModel` property that represents the selection by the user. Its `getSelectedItems()` methods returns an observable list of selected items. Behavior can be initiated by the control by registering an invalidation listener with the observable list.

The state property of a control can also be used when the control is part of the response. Then you just need to invoke the setter for the state property.