

Chapter 16

JavaFX UI Controls



Motivations

A graphical user interface (GUI) makes a system user-friendly and easy to use. Creating a GUI requires creativity and knowledge of how GUI components work. Since the GUI components in Java are very flexible and versatile, you can create a wide assortment of useful user interfaces.

Previous chapters briefly introduced several GUI components. This chapter introduces the frequently used GUI components in detail.

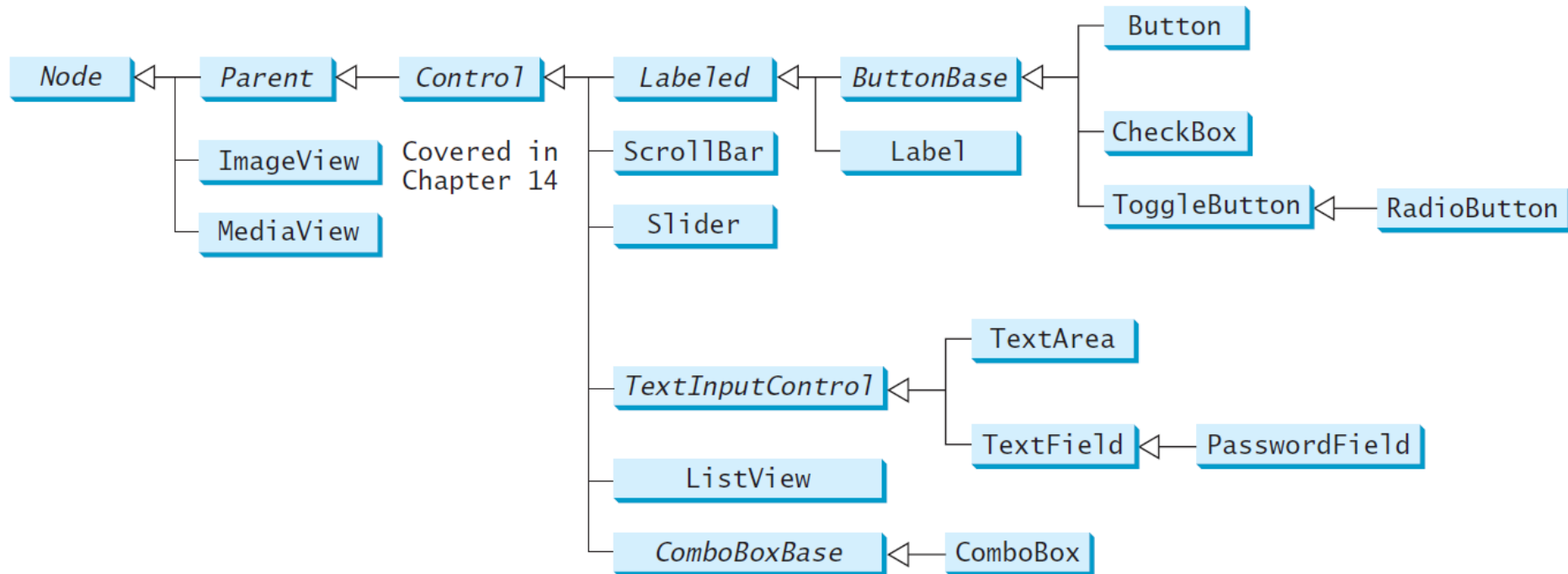


Objectives

- ☞ To create graphical user interfaces with various user-interface controls (§§16.2–16.11).
- ☞ To create a label with text and graphic using the **Label** class and explore properties in the abstract **Labeled** class (§16.2).
- ☞ To create a button with text and graphic using the **Button** class and set a handler using the **setOnAction** method in the abstract **ButtonBase** class (§16.3).
- ☞ To create a check box using the **CheckBox** class (§16.4).
- ☞ To create a radio button using the **RadioButton** class and group radio buttons using a **ToggleGroup** (§16.5).
- ☞ To enter data using the **TextField** class and password using the **PasswordField** class (§16.6).
- ☞ To enter data in multiple lines using the **TextArea** class (§16.7).
- ☞ To select a single item using **ComboBox** (§16.8).
- ☞ To select a single or multiple items using **ListView** (§16.9).
- ☞ To select a range of values using **ScrollBar** (§16.10).
- ☞ To select a range of values using **Slider** and explore differences between **ScrollBar** and **Slider** (§16.11).
- ☞ To develop a tic-tac-toe game (§16.12).
- ☞ To view and play video and audio using the **Media**, **MediaPlayer**, and **MediaView** (§16.13).
- ☞ To develop a case study for showing the national flag and play anthem (§16.14).



Frequently Used UI Controls



Throughout this book, the prefixes **lbl**, **bt**, **chk**, **rb**, **tf**, **pf**, **ta**, **cbo**, **lv**, **scb**, **sld**, and **mp** are used to name reference variables for **Label**, **Button**, **CheckBox**, **RadioButton**, **TextField**, **PasswordField**, **TextArea**, **ComboBox**, **ListView**, **ScrollBar**, **Slider**, and **MediaPlayer**.

Labeled

A *label* is a display area for a short text, a node, or both. It is often used to label other controls (usually text fields). Labels and buttons share many common properties. These common properties are defined in the **Labeled** class.

javafx.scene.control.Labeled

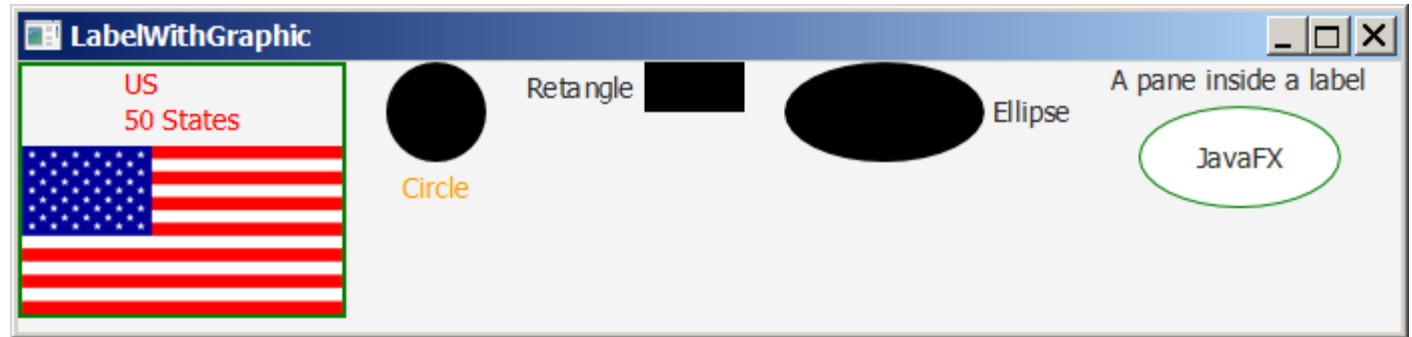
-alignment: ObjectProperty<Pos>
-contentDisplay:
 ObjectProperty<ContentDisplay>
-graphic: ObjectProperty<Node>
-graphicTextGap: DoubleProperty
-textFill: ObjectProperty<Paint>
-text: StringProperty
-underline: BooleanProperty
-wrapText: BooleanProperty

The getter and setter methods for property values and a getter for property itself are provided in the class, but omitted in the UML diagram for brevity.

Specifies the alignment of the text and node in the labeled.
Specifies the position of the node relative to the text using the constants TOP, BOTTOM, LEFT, and RIGHT defined in ContentDisplay.
A graphic for the labeled.
The gap between the graphic and the text.
The paint used to fill the text.
A text for the labeled.
Whether text should be underlined.
Whether text should be wrapped if the text exceeds the width.

Label

The Label class defines labels.



`javafx.scene.control.Labeled`

`javafx.scene.control.Label`

+Label()
+Label(text: String)
+Label(text: String, graphic: Node)

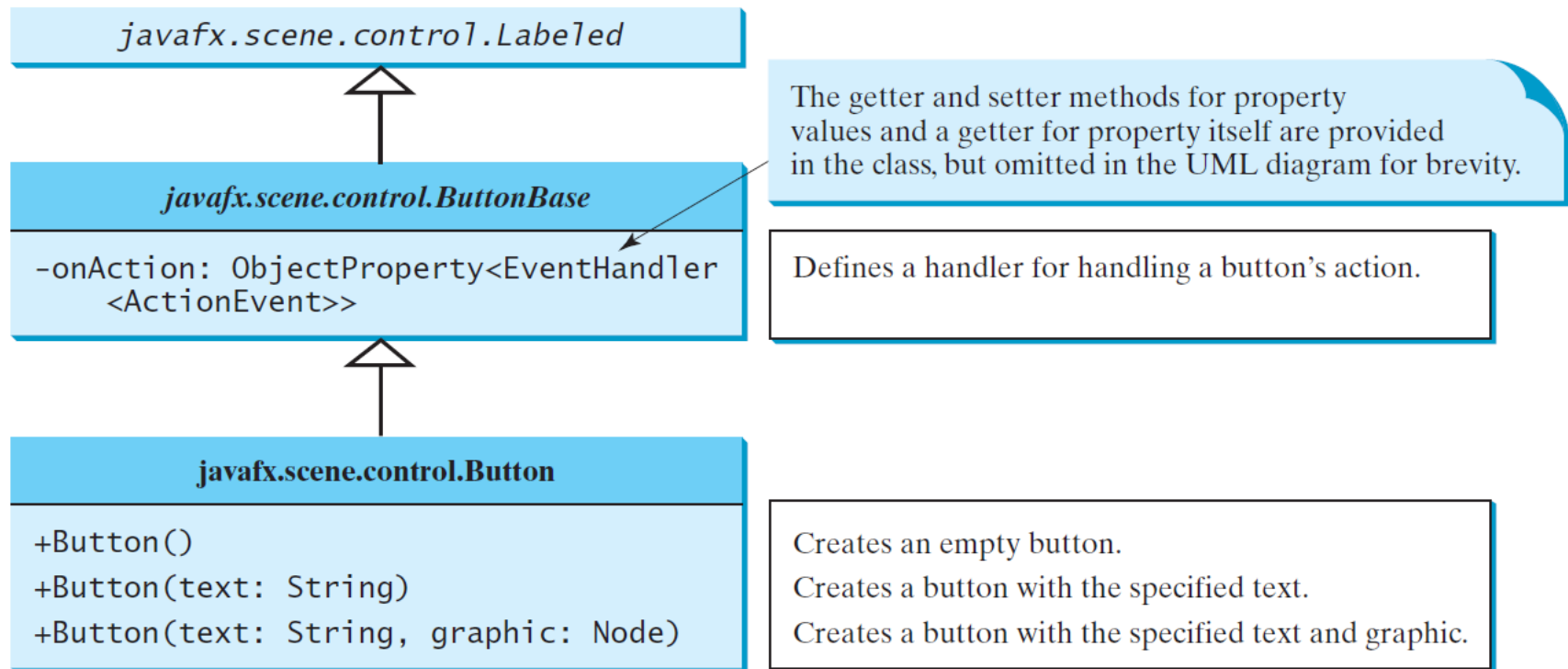
Creates an empty label.
Creates a label with the specified text.
Creates a label with the specified text and graphic.

LabelWithGraphic

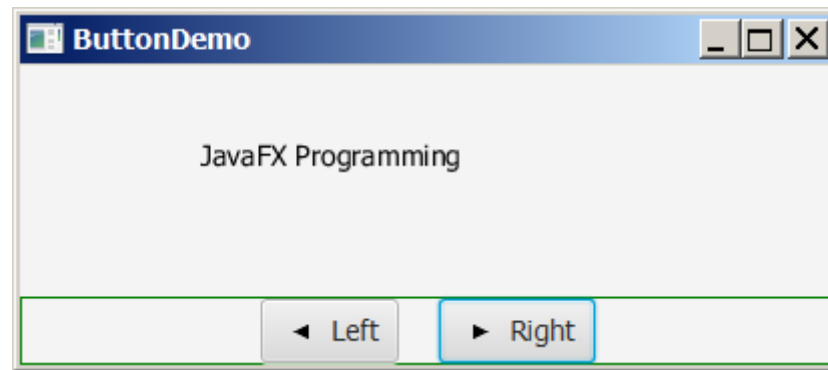
Run

ButtonBase and Button

A *button* is a control that triggers an action event when clicked. JavaFX provides regular buttons, toggle buttons, check box buttons, and radio buttons. The common features of these buttons are defined in **ButtonBase** and **Labeled** classes.



Button Example

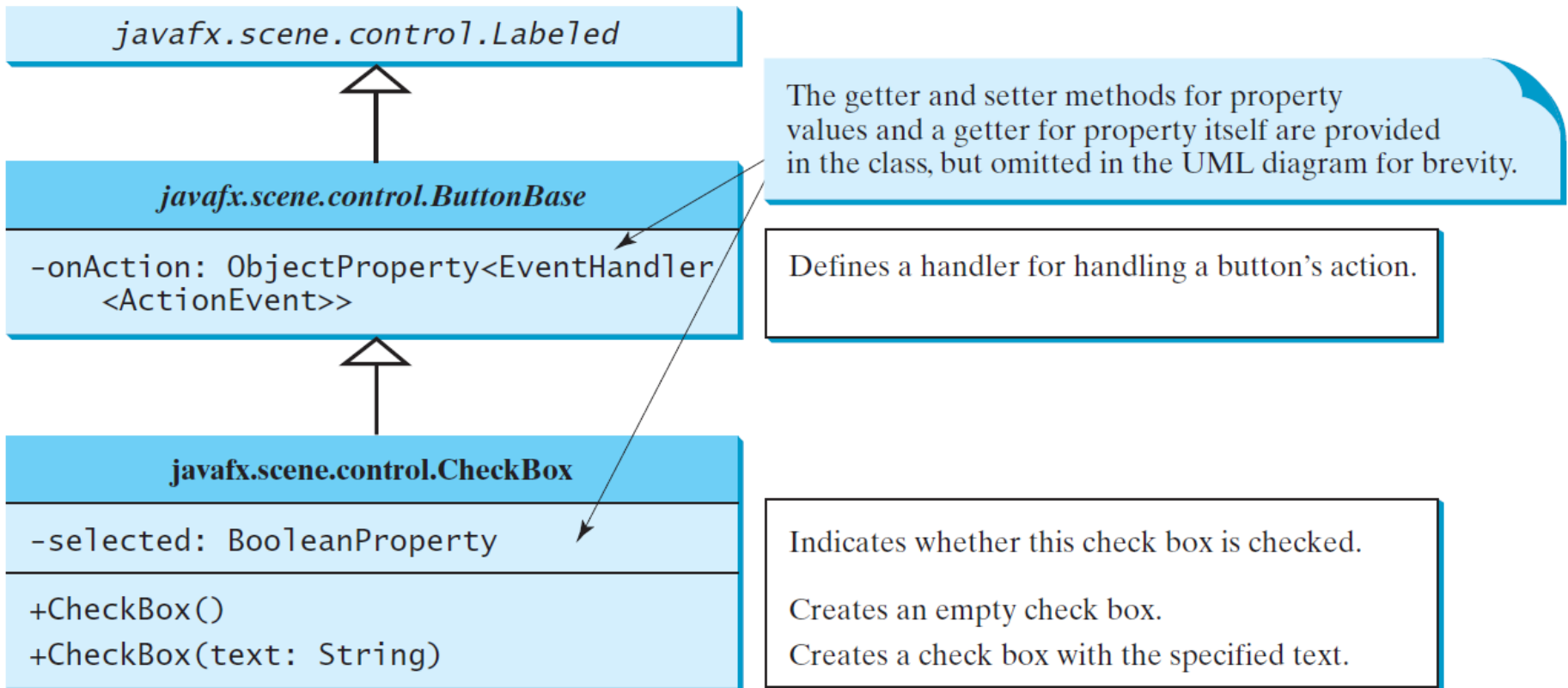


ButtonDemo

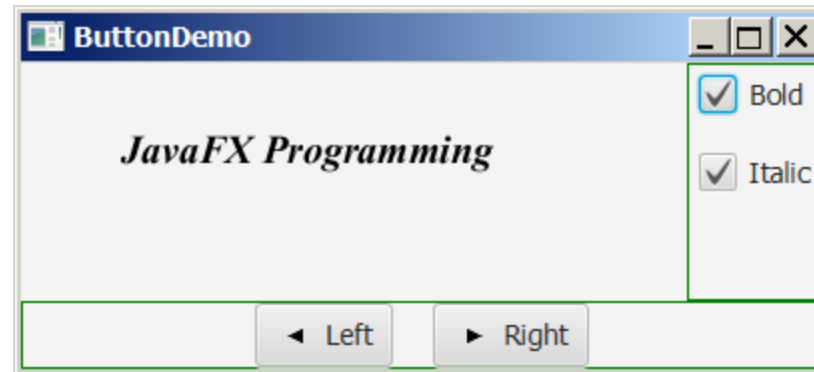
Run

CheckBox

A **CheckBox** is used for the user to make a selection. Like **Button**, **CheckBox** inherits all the properties such as **onAction**, **text**, **graphic**, **alignment**, **graphicTextGap**, **textFill**, **contentDisplay** from **ButtonBase** and **Labeled**.



CheckBox Example

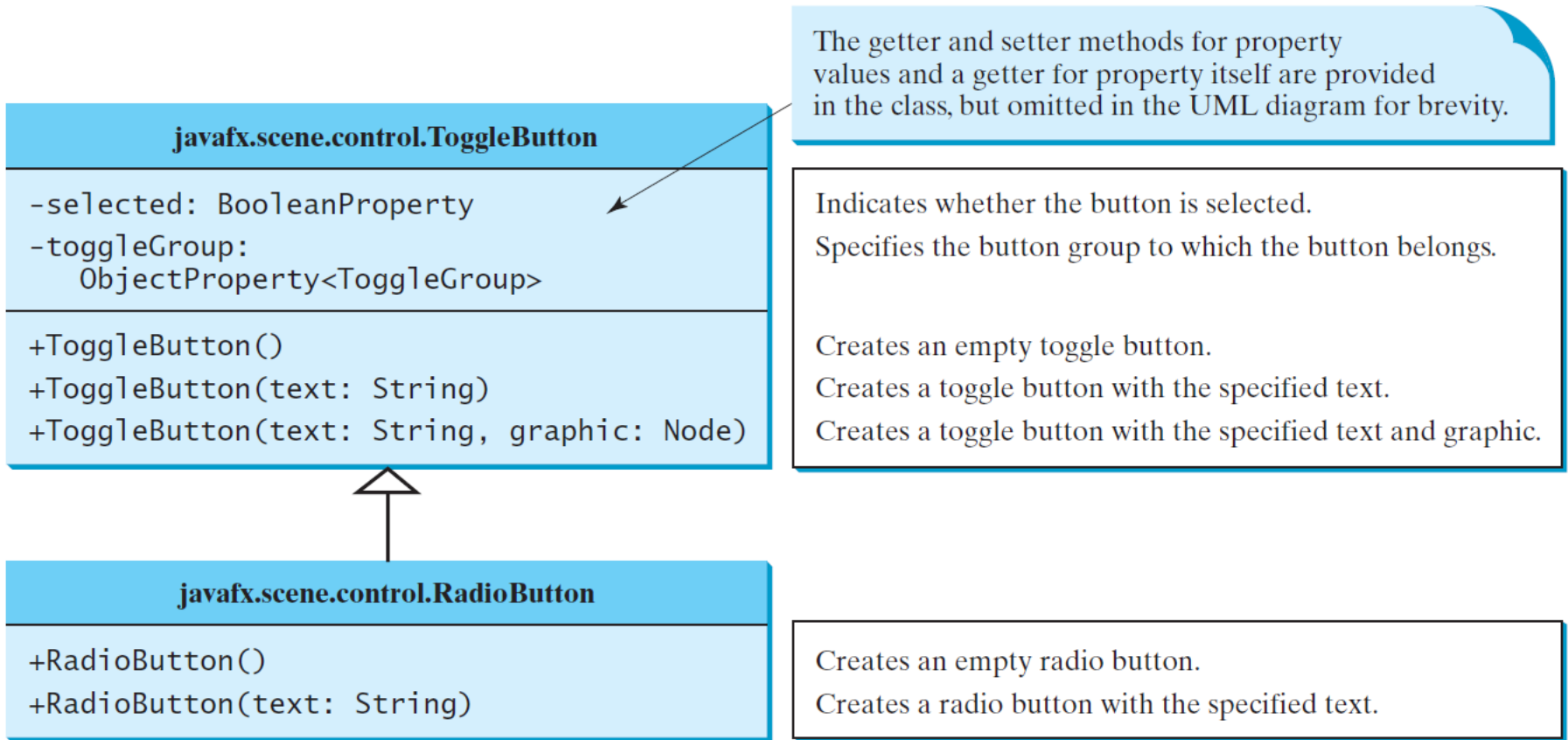


CheckBoxDemo

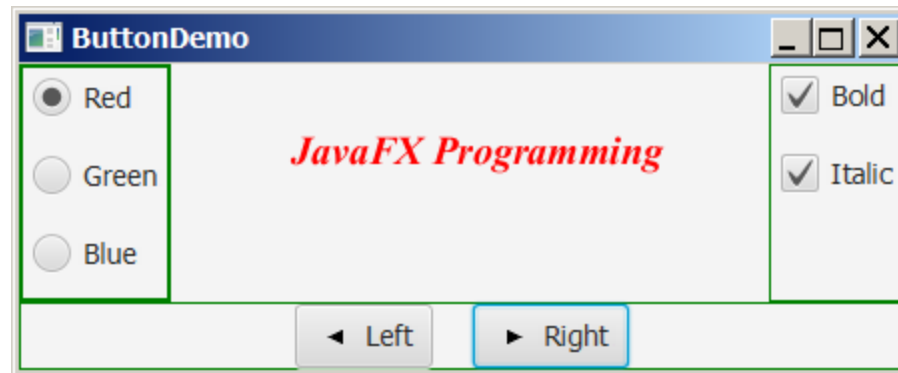
Run

RadioButton

Radio buttons, also known as *option buttons*, enable you to choose a single item from a group of choices. In appearance radio buttons resemble check boxes, but check boxes display a square that is either checked or blank, whereas radio buttons display a circle that



RadioButton Example

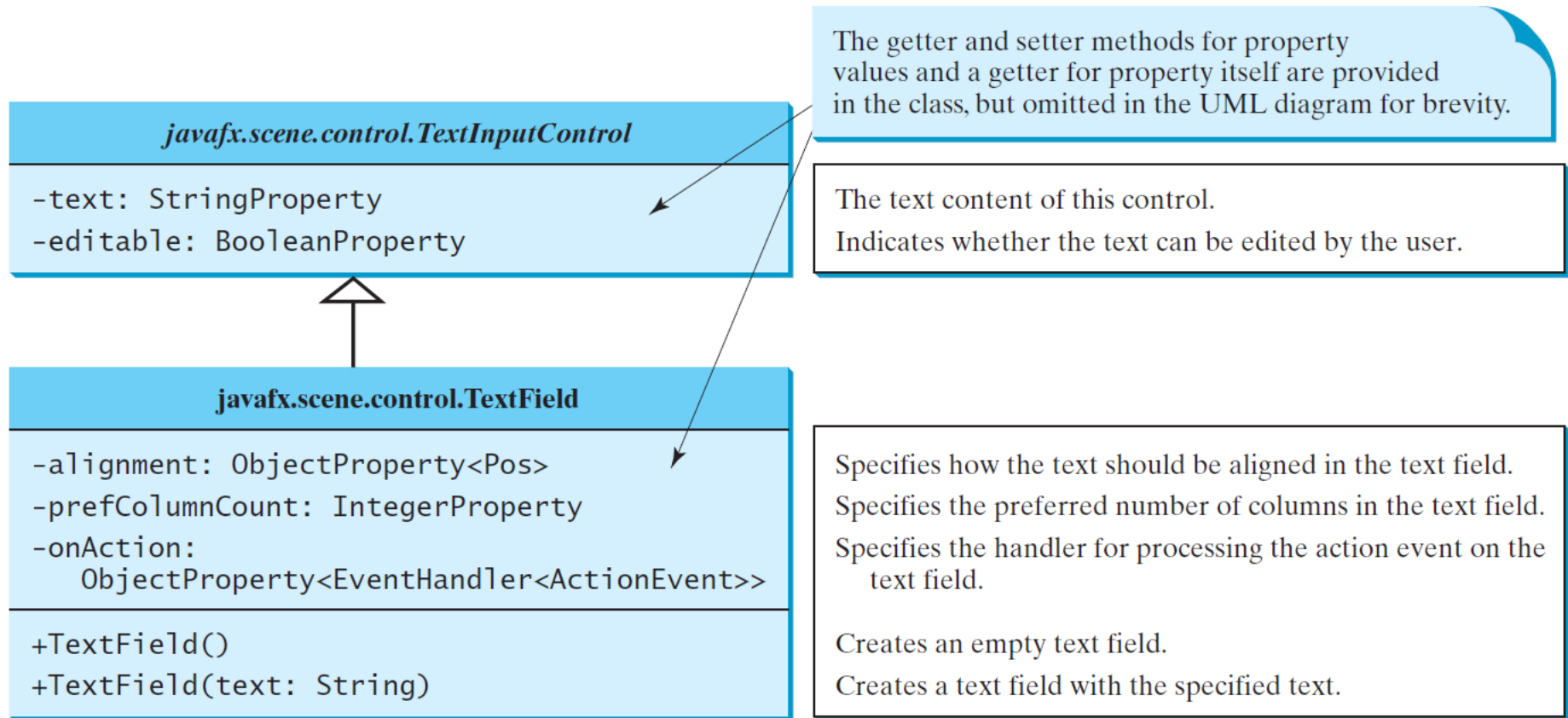


RadioButtonDemo

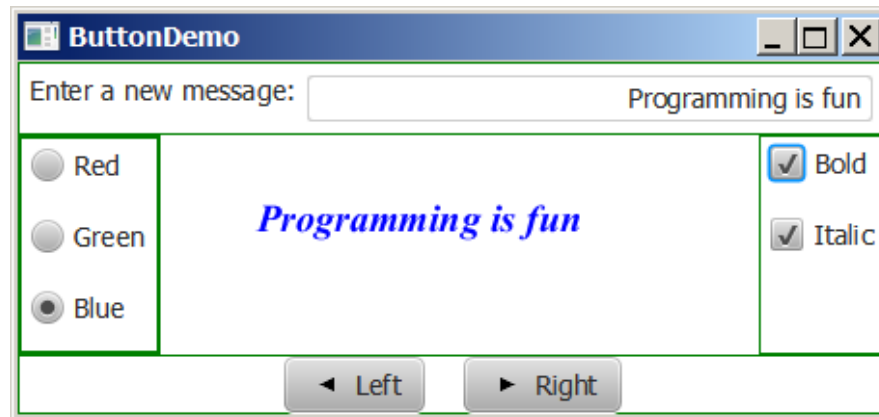
Run

TextField

A text field can be used to enter or display a string. **TextField** is a subclass of **TextInputControl**.



TextField Example

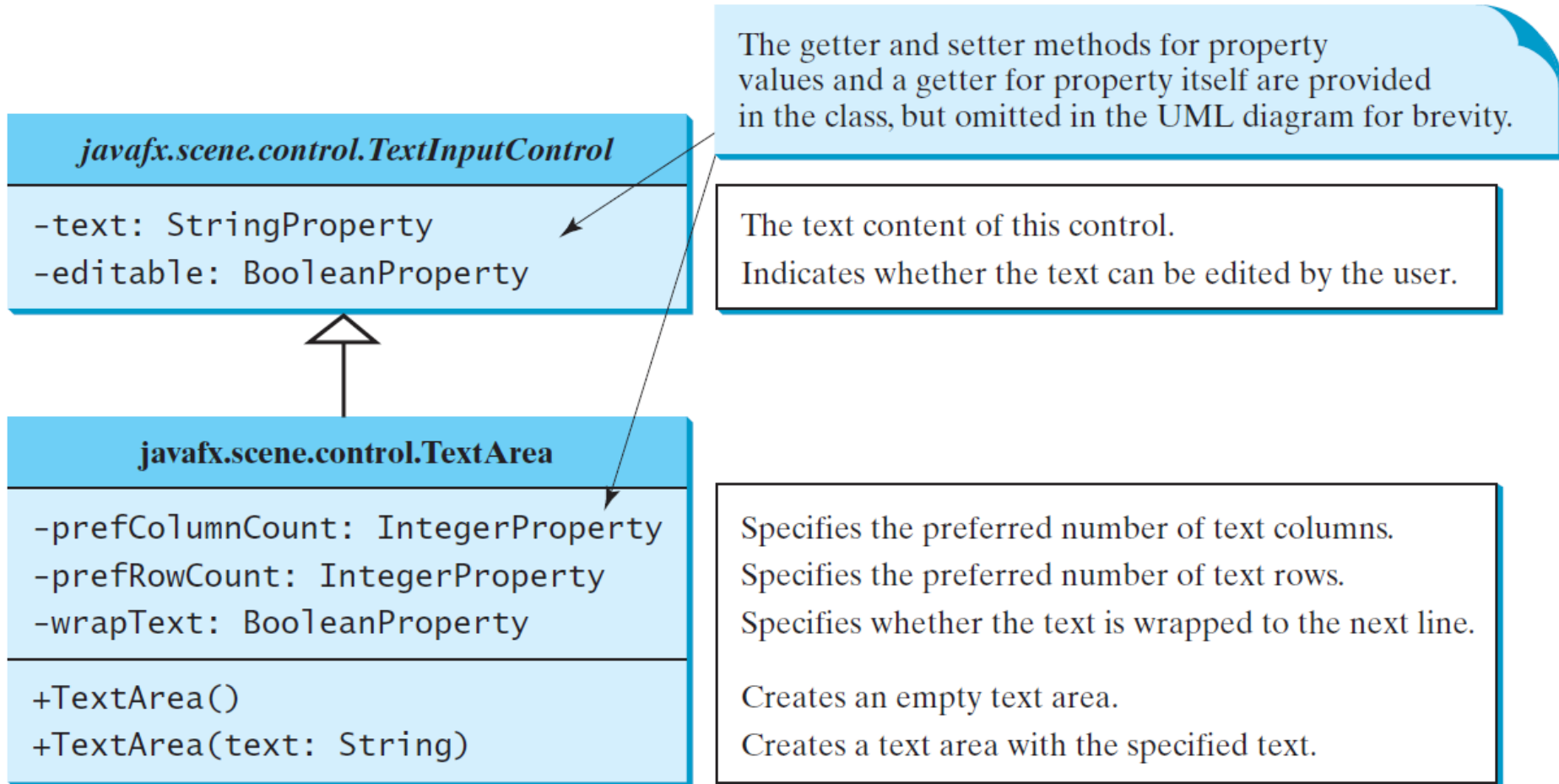


TextFieldDemo

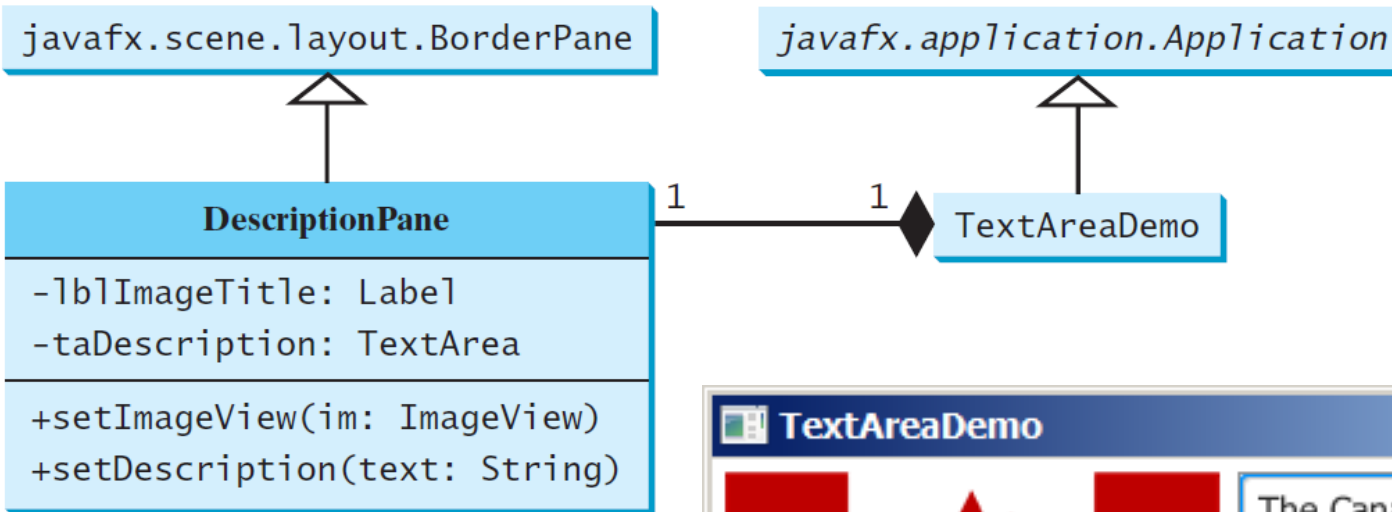
Run

TextArea

A **TextArea** enables the user to enter multiple lines of text.



TextArea Example



DescriptionPane

TextAreaDemo

Run