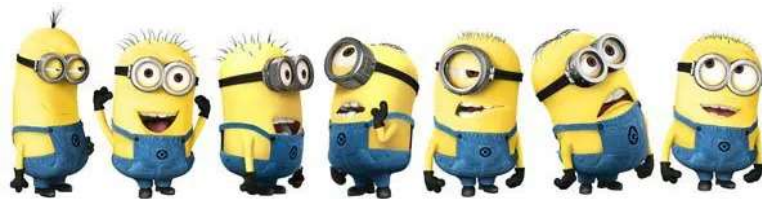


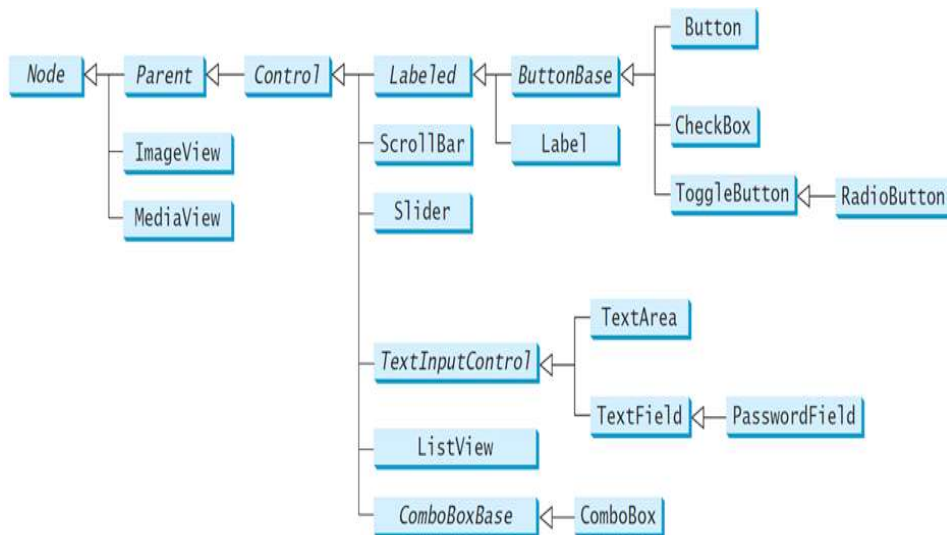
# JavaFX UI Controls

Liang, Introduction to Java programming, 11<sup>th</sup> Edition, © 2017 Pearson Education, Inc.  
All rights reserved



By: Mamoun Nawahdah (Ph.D.)  
2020

## Frequently Used UI Controls



## 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

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.



3

## Label

- ❖ The **Label** class defines labels.

*javafx.scene.control.Labeled*



*javafx.scene.control.Label*

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

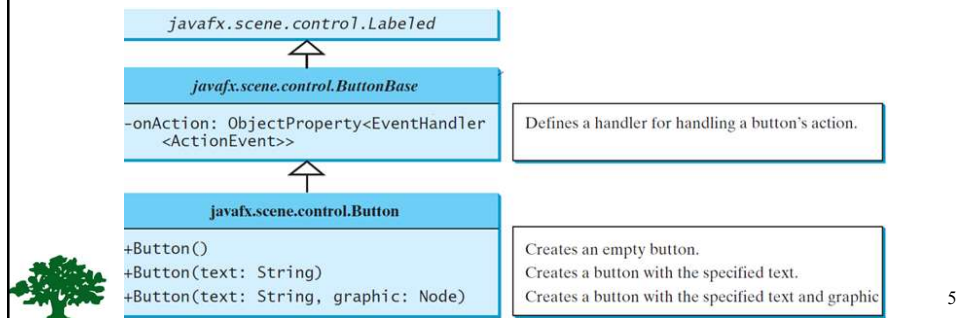
Creates an empty label.  
 Creates a label with the specified text.  
 Creates a label with the specified text and graphic.



4

## ButtonBase and Button Say 'Hello World'

- ❖ A **Button** is a UI component 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.



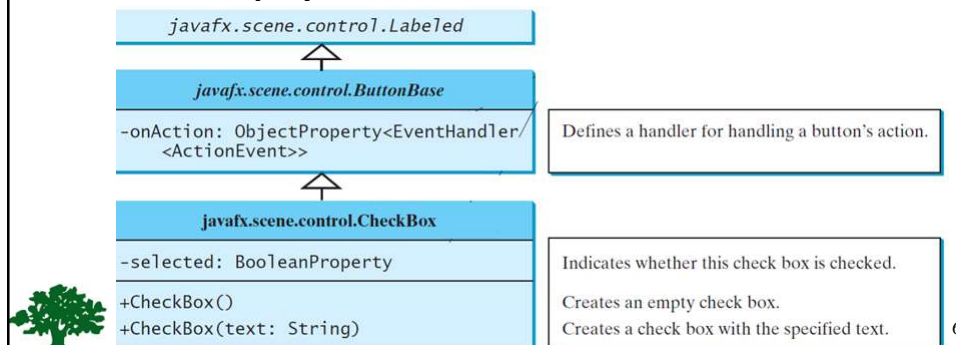
## CheckBox

- ❖ A **CheckBox** is used for the user to make a selection (**0 or more**).

Which pets do you have?

- Dog
- Cat
- Lizard
- Bird

- ❖ **CheckBox** inherits all the properties such as **onAction**, **text**, **graphic**, **alignment**, **graphicTextGap**, **textFill**, **contentDisplay** from **ButtonBase** and **Labeled**.



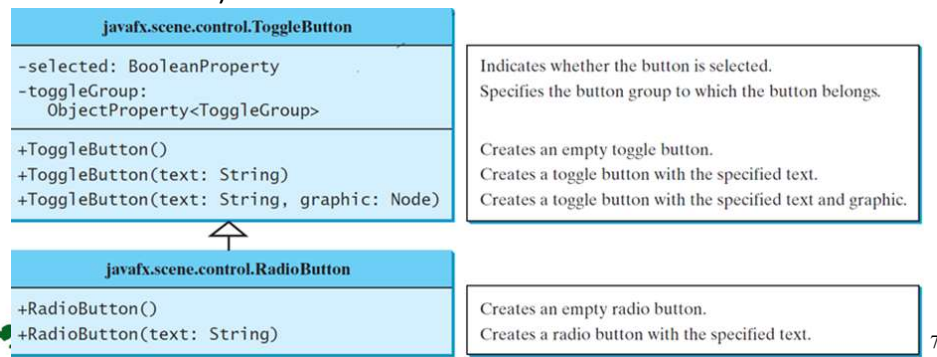
# RadioButton

Do you have pets?

Yes

No

- ❖ 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 is either filled (if selected) or blank (if not selected).

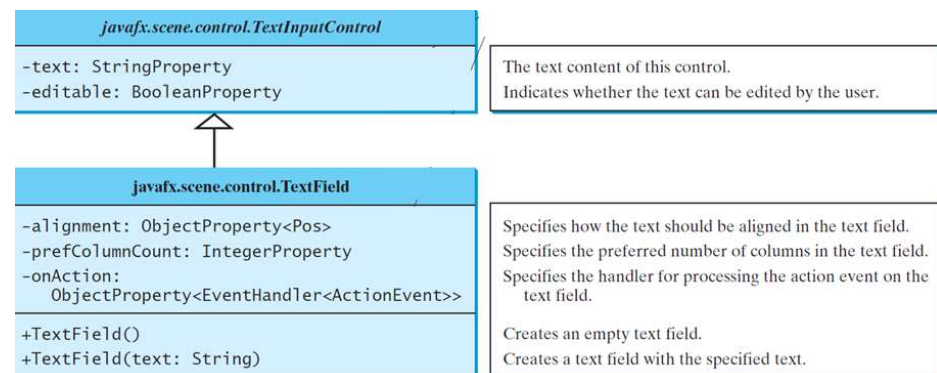


7

# TextField

Enter first your name.

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



8

# TextArea

Text Area 1

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

*javafx.scene.control.TextInputControl*

-text: StringProperty  
-editable: BooleanProperty

The text content of this control.  
Indicates whether the text can be edited by the user.

*javafx.scene.control.TextArea*

-prefColumnCount: IntegerProperty  
-prefRowCount: IntegerProperty  
-wrapText: BooleanProperty

Specifies the preferred number of text columns.  
Specifies the preferred number of text rows.  
Specifies whether the text is wrapped to the next line.

+TextArea()  
+TextArea(text: String)

Creates an empty text area.  
Creates a text area with the specified text.



9

# ComboBox

- ❖ A combo box, also known as a choice list or drop-down list, contains a list of items from which the user can choose.

*javafx.scene.control.ComboBoxBase<T>*

-value: ObjectProperty<T>  
-editable: BooleanProperty  
-onAction: ObjectProperty<EventHandler<ActionEvent>>/

The value selected in the combo box.  
Specifies whether the combo box allows user input.  
Specifies the handler for processing the action event.

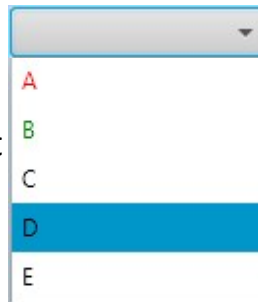
*javafx.scene.control.ComboBox<T>*

-items: ObjectProperty<ObservableList<T>>  
-visibleRowCount: IntegerProperty

The items in the combo box popup.  
The maximum number of visible rows of the items in the combo box popup.

+ComboBox()  
+ComboBox(items: ObservableList<T>)

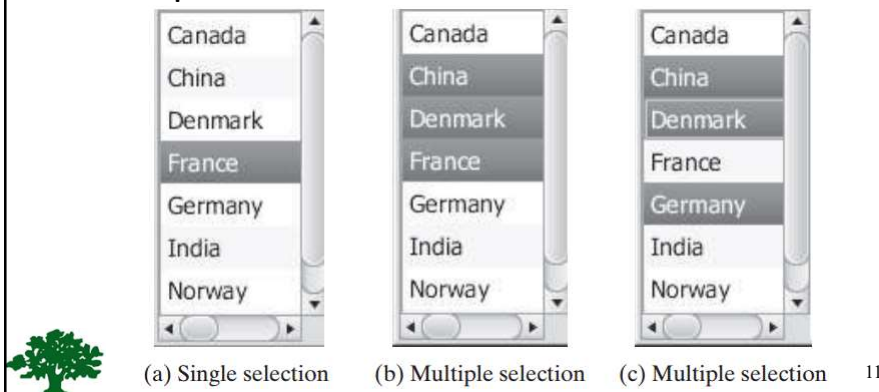
Creates an empty combo box.  
Creates a combo box with the specified items.



10

## ListView

❖ A **list view** is a component that performs basically the same function as a combo box, but it enables the user to choose a single value or multiple values.



11

## ListView

`javafx.scene.control.ListView<T>`

```
-items: ObjectProperty<ObservableList<T>>
-orientation: BooleanProperty

-selectionModel:
  ObjectProperty<MultipleSelectionModel<T>>

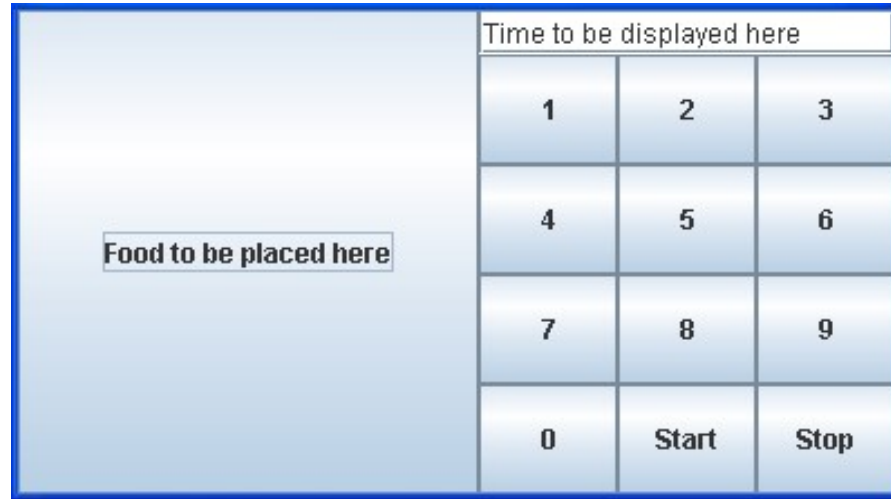
+ListView()
+ListView(items: ObservableList<T>)
```

The items in the list view.  
Indicates whether the items are displayed horizontally or vertically in the list view.  
Specifies how items are selected. The `SelectionModel` is also used to obtain the selected items.  
Creates an empty list view.  
Creates a list view with the specified items.

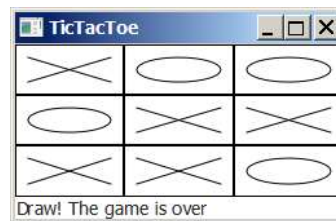
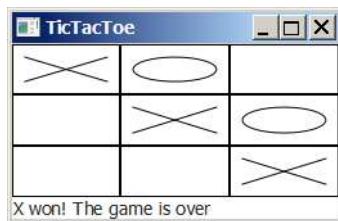
12



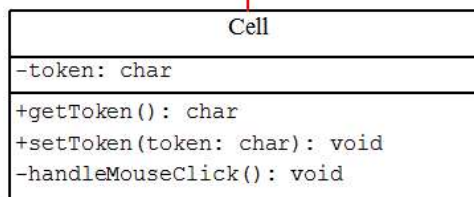
## Building a Microwave Interface



## Case Study: TicTacToe



`javafx.scene.layout.Pane`



Token used in the cell (default: ' ').  
 Returns the token in the cell.  
 Sets a new token in the cell.  
 Handles a mouse click event.



## Case Study: TicTacToe cont.

