Chapter 31 — JavaFX Basics

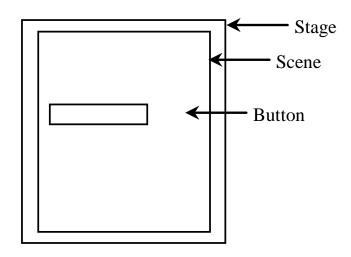
Lia10, C14

About JavaFX

- JavaFX started in Java 8.
- JavaFX is newer Swing.

Basic Structure of JavaFX

- Application
- Override the start(Stage) method
- Stage (one or more), Scene, and Nodes

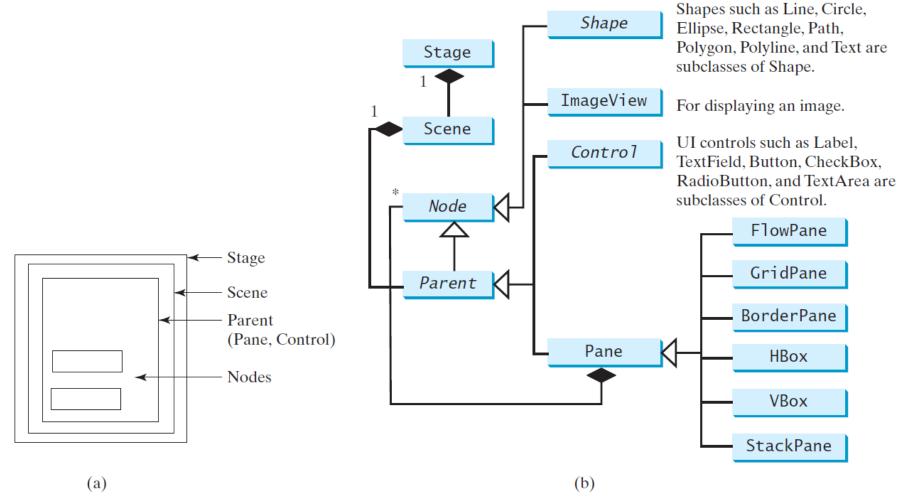


Example MyJavaFX

Note the difference in red between Swing & JavaFX programs

```
import javafx.application.Application
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.control.Button;
public class MyJavaFX extends Application {
    @Override // Override the start method in the Application class
   public void start(Stage primaryStage) {
        // Create a scene and place a button in the scene
        Button btOK = new Button("OK");
        Scene scene = new Scene (btOK, 200, 250);
        primaryStage.setTitle("MyJavaFX"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
    /**
    * The main method is only needed for the IDE with limited
    * JavaFX support. Not needed for running from the command line.
    */
   public static void main(String[] args) {
      Application.launch(args);
```

Panes, UI Controls, and Shapes



A Scene can contain a Pane or Control, but not a Shape or ImageView

The Color Class

javafx.scene.paint.Color

```
-red: double
-green: double
-blue: double
-opacity: double
+Color(r: double, g: double, b:
   double, opacity: double)
+brighter(): Color
+darker(): Color
+color(r: double, g: double, b:
   double): Color
+color(r: double, g: double, b:
   double, opacity: double): Color
+rgb(r: int, g: int, b: int):
   Color
+rgb(r: int, g: int, b: int,
   opacity: double): Color
```

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

The red value of this Color (between 0.0 and 1.0).

The green value of this Color (between 0.0 and 1.0).

The blue value of this Color (between 0.0 and 1.0).

The opacity of this Color (between 0.0 and 1.0).

Creates a Color with the specified red, green, blue, and opacity values.

Creates a Color that is a brighter version of this Color.

Creates a Color that is a darker version of this Color.

Creates an opaque Color with the specified red, green, and blue values.

Creates a Color with the specified red, green, blue, and opacity values.

Creates a Color with the specified red, green, and blue values in the range from 0 to 255.

Creates a Color with the specified red, green, and blue values in the range from 0 to 255 and a given opacity.

```
Color c = new Color(0,0,1.0); // constructor call
Color c = Color.color(0,0,1.0); vs. Color c = Color.rgb(0,0,255);
All pre-defined colors are in uppercase; e.g., Color.BEIGE
```

LIA10-C14

The Font Class

javafx.scene.text.Font

```
-size: double
-name: String
-family: String

+Font(size: double)
+Font(name: String, size:
    double)
+font(name: String, size:
    double)
+font(name: String, w:
    FontWeight, size: double)
+font(name: String, w: FontWeight,
    p: FontPosture, size: double)
+getFamilies(): List<String>
+getFontNames(): List<String>
```

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

The size of this font.

The name of this font.

The family of this font.

Creates a Font with the specified size.

Creates a Font with the specified full font name and size.

Creates a Font with the specified name and size.

Creates a Font with the specified name, weight, and size.

Creates a Font with the specified name, weight, posture, and size.

Returns a list of font family names.

Returns a list of full font names including family and weight.

The Image Class

javafx.scene.image.Image

-error: ReadOnlyBooleanProperty

-height: ReadOnlyBooleanProperty

-width: ReadOnlyBooleanProperty

-progress: ReadOnlyBooleanProperty

+Image(filenameOrURL: String)

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

Indicates whether the image is loaded correctly?

The height of the image.

The width of the image.

The approximate percentage of image's loading that is completed.

Creates an Image with contents loaded from a file or a URL.

Note: a URL ("http://www.csCourse.net/image/us.gif") can be passed to constructor Image

The ImageView Class

javafx.scene.image.ImageView

-fitHeight: DoubleProperty

-fitWidth: DoubleProperty

-x: DoubleProperty

-y: DoubleProperty

-image: ObjectProperty<Image>

+ImageView()

+ImageView(image: Image)

+ImageView(filenameOrURL: String)

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.

The height of the bounding box within which the image is resized to fit.

The width of the bounding box within which the image is resized to fit.

The x-coordinate of the ImageView origin.

The y-coordinate of the ImageView origin.

The image to be displayed in the image view.

Creates an ImageView.

Creates an ImageView with the specified image.

Creates an ImageView with image loaded from the specified file or URL.

FlowPane: fill row 1, then 2, etc.

javafx.scene.layout.FlowPane

```
-alignment: ObjectProperty<Pos>
-orientation:
    ObjectProperty<Orientation>
-hgap: DoubleProperty
-vgap: DoubleProperty
```

```
-hgap: DoubleProperty
-vgap: DoubleProperty

+FlowPane()
+FlowPane(hgap: double, vgap: double)
+FlowPane(orientation: ObjectProperty<Orientation>)
+FlowPane(orientation: ObjectProperty<Orientation>, hgap: double, vgap: double
```

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.

The overall alignment of the content in this pane (default: Pos.LEFT). The orientation in this pane (default: Orientation.HORIZONTAL).

The horizontal gap between the nodes (default: 0). The vertical gap between the nodes (default: 0).

Creates a default FlowPane.

Creates a FlowPane with a specified horizontal and vertical gap.

Creates a FlowPane with a specified orientation.

Creates a FlowPane with a specified orientation, horizontal gap and vertical gap.

GridPane

javafx.scene.layout.GridPane

-alignment: ObjectProperty<Pos>

```
-gridLinesVisible:
   BooleanProperty
-hgap: DoubleProperty
-vgap: DoubleProperty
+GridPane()
+add(child: Node, columnIndex:
   int. rowIndex: int): void
+addColumn(columnIndex: int,
   children: Node...): void
+addRow(rowIndex: int,
   children: Node...): void
+getColumnIndex(child: Node):
   int
+setColumnIndex(child: Node.
   columnIndex: int): void
+getRowIndex(child:Node): int
+setRowIndex(child: Node,
   rowIndex: int): void
+setHalighnment(child: Node,
   value: HPos): void
+setValighnment(child: Node,
   value: VPos): void
```

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.

```
The overall alignment of the content in this pane (default: Pos.LEFT). Is the grid line visible? (default: false)
```

The horizontal gap between the nodes (default: 0).

The vertical gap between the nodes (default: 0).

Creates a GridPane.

Adds a node to the specified column and row.

Adds multiple nodes to the specified column.

Adds multiple nodes to the specified row.

Returns the column index for the specified node.

Sets a node to a new column. This method repositions the node.

Returns the row index for the specified node.

Sets a node to a new row. This method repositions the node.

Sets the horizontal alignment for the child in the cell.

Sets the vertical alignment for the child in the cell.

BorderPane

javafx.scene.layout.BorderPane

-top: ObjectProperty<Node>

-right: ObjectProperty<Node>

-bottom: ObjectProperty<Node>

-left: ObjectProperty<Node>

-center: ObjectProperty<Node>

+BorderPane()

+setAlignment(child: Node, pos:
 Pos)

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.

The node placed in the top region (default: null).

The node placed in the right region (default: null).

The node placed in the bottom region (default: null).

The node placed in the left region (default: null).

The node placed in the center region (default: null).

Creates a BorderPane.

Sets the alignment of the node in the BorderPane.

HBox

javafx.scene.layout.HBox

-alignment: ObjectProperty<Pos>
-fillHeight: BooleanProperty
-spacing: DoubleProperty

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.

The overall alignment of the children in the box (default: Pos.TOP_LEFT). Is resizable children fill the full height of the box (default: true).

The horizontal gap between two nodes (default: 0).

Creates a default HBox.

Creates an HBox with the specified horizontal gap between nodes.

Sets the margin for the node in the pane.

VBox

javafx.scene.layout.VBox

-alignment: ObjectProperty<Pos>

-fillWidth: BooleanProperty

-spacing: DoubleProperty

+VBox()

+VBox(spacing: double)

+setMargin(node: Node, value:

Insets): void

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.

The overall alignment of the children in the box (default: Pos.TOP_LEFT).

Is resizable children fill the full width of the box (default: true).

The vertical gap between two nodes (default: 0).

Creates a default VBox.

Creates a VBox with the specified horizontal gap between nodes.

Sets the margin for the node in the pane.

Text

javafx.scene.text.Text

```
-text: StringProperty
-x: DoubleProperty
-y: DoubleProperty
-underline: BooleanProperty
-strikethrough: BooleanProperty
-font: ObjectProperty
+Text()
+Text(text: String)
+Text(x: double, y: double, text: String)
```

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.

Defines the text to be displayed.

Defines the x-coordinate of text (default 0).

Defines the y-coordinate of text (default 0).

Defines if each line has an underline below it (default false).

Defines if each line has a line through it (default false).

Defines the font for the text.

Creates an empty Text.

Creates a Text with the specified text.

Creates a Text with the specified x-, y-coordinates and text.

Line

javafx.scene.shape.Line

-startX: DoubleProperty-startY: DoubleProperty-endX: DoubleProperty-endY: DoubleProperty

```
+Line()
+Line(startX: double, startY:
    double, endX: double, endY:
    double)
```

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.

The x-coordinate of the start point.

The y-coordinate of the start point.

The x-coordinate of the end point.

The y-coordinate of the end point.

Creates an empty Line.

Creates a Line with the specified starting and ending points.

```
(0, 0) (getWidth(), 0)
```

```
(startX, startY)

(endX, endY)
```

```
(0, getHeight()) (getWidth(), getHeight())
```

Rectangle

javafx.scene.shape.Rectangle

-x: DoubleProperty

-y:DoubleProperty

-width: DoubleProperty

-height: DoubleProperty

-arcWidth: DoubleProperty

-arcHeight: DoubleProperty

+Rectangle()

+Rectanlge(x: double, y:
 double, width: double,

height: double)

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.

The x-coordinate of the upper-left corner of the rectangle (default 0).

The y-coordinate of the upper-left corner of the rectangle (default 0).

The width of the rectangle (default: 0).

The height of the rectangle (default: 0).

The arcWidth of the rectangle (default: 0). arcWidth is the horizontal diameter of the arcs at the corner (see Figure 14.31a).

The arcHeight of the rectangle (default: 0). arcHeight is the vertical diameter of the arcs at the corner (see Figure 14.31a).

Creates an empty Rectangle.

Creates a Rectangle with the specified upper-left corner point, width, and height.

Circle

javafx.scene.shape.Circle

```
-centerX: DoubleProperty
-centerY: DoubleProperty
-radius: DoubleProperty
+Circle()
+Circle(x: double, y: double)
+Circle(x: double, y: double, radius: double)
```

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.

The x-coordinate of the center of the circle (default 0).

The y-coordinate of the center of the circle (default 0).

The radius of the circle (default: 0).

Creates an empty Circle.

Creates a Circle with the specified center.

Creates a Circle with the specified center and radius.

Ellipse

javafx.scene.shape.Ellipse

-centerX: DoubleProperty
-centerY: DoubleProperty
-radiusX: DoubleProperty
-radiusY: DoubleProperty
+Ellipse()

+Ellipse()
+Ellipse(x: double, y: double)
+Ellipse(x: double, y: double,
 radiusX: double, radiusY:
 double)

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.

The x-coordinate of the center of the ellipse (default 0).

The y-coordinate of the center of the ellipse (default 0).

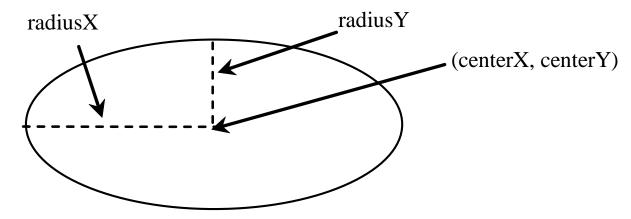
The horizontal radius of the ellipse (default: 0).

The vertical radius of the ellipse (default: 0).

Creates an empty Ellipse.

Creates an Ellipse with the specified center.

Creates an Ellipse with the specified center and radiuses.



Arc

javafx.scene.shape.Arc

-centerX: DoubleProperty
-centerY: DoubleProperty
-radiusX: DoubleProperty
-radiusY: DoubleProperty
-startAngle: DoubleProperty
-length: DoubleProperty

-type: ObjectProperty<ArcType>

+Arc()

+Arc(x: double, y: double,
 radiusX: double, radiusY:
 double, startAngle: double,
 length: double)

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.

The x-coordinate of the center of the ellipse (default 0).

The y-coordinate of the center of the ellipse (default 0).

The horizontal radius of the ellipse (default: 0).

The vertical radius of the ellipse (default: 0).

The start angle of the arc in degrees.

The angular extent of the arc in degrees.

The closure type of the arc (ArcType.OPEN, ArcType.CHORD, ArcType.ROUND).

Creates an empty Arc.

Creates an Arc with the specified arguments.