JavaFX Basics

Motivations

JavaFX is a new framework for developing Java GUI programs.

The JavaFX API is an excellent example of how the object-oriented principle is applied.

JavaFX vs Swing and AWT

Swing and AWT are replaced by the JavaFX platform for developing rich Internet applications.

Abstract Windows Toolkit (AWT)

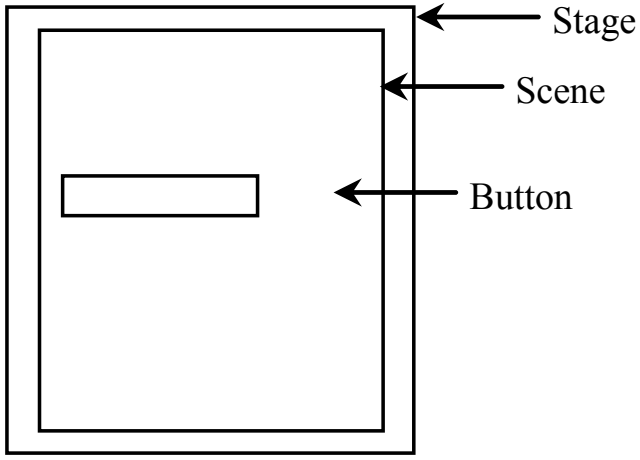
- fine for developing simple graphical user interfaces
- prone to platform-specific bugs
- replaced by a more robust, versatile, and flexible library known as *Swing components*

Swing components

- paint directly on canvases using Java code.
- Swing components depend less on the target platform and uses less of the native GUI resource.
- With the release of Java 8, Swing is replaced by a completely new GUI platform known as *JavaFX*.

Basic Structure of JavaFX

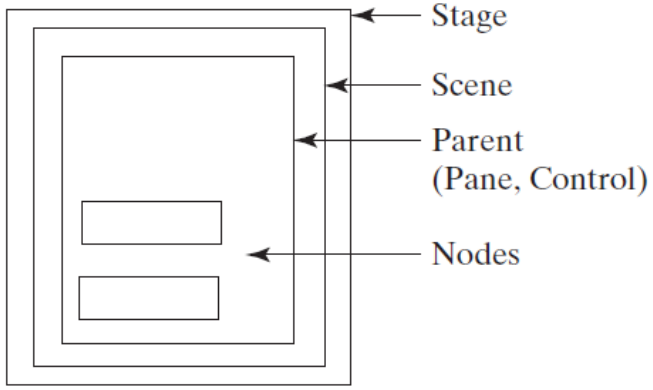
- Application
- Override the start(Stage) method
- Stage, Scene, and Nodes



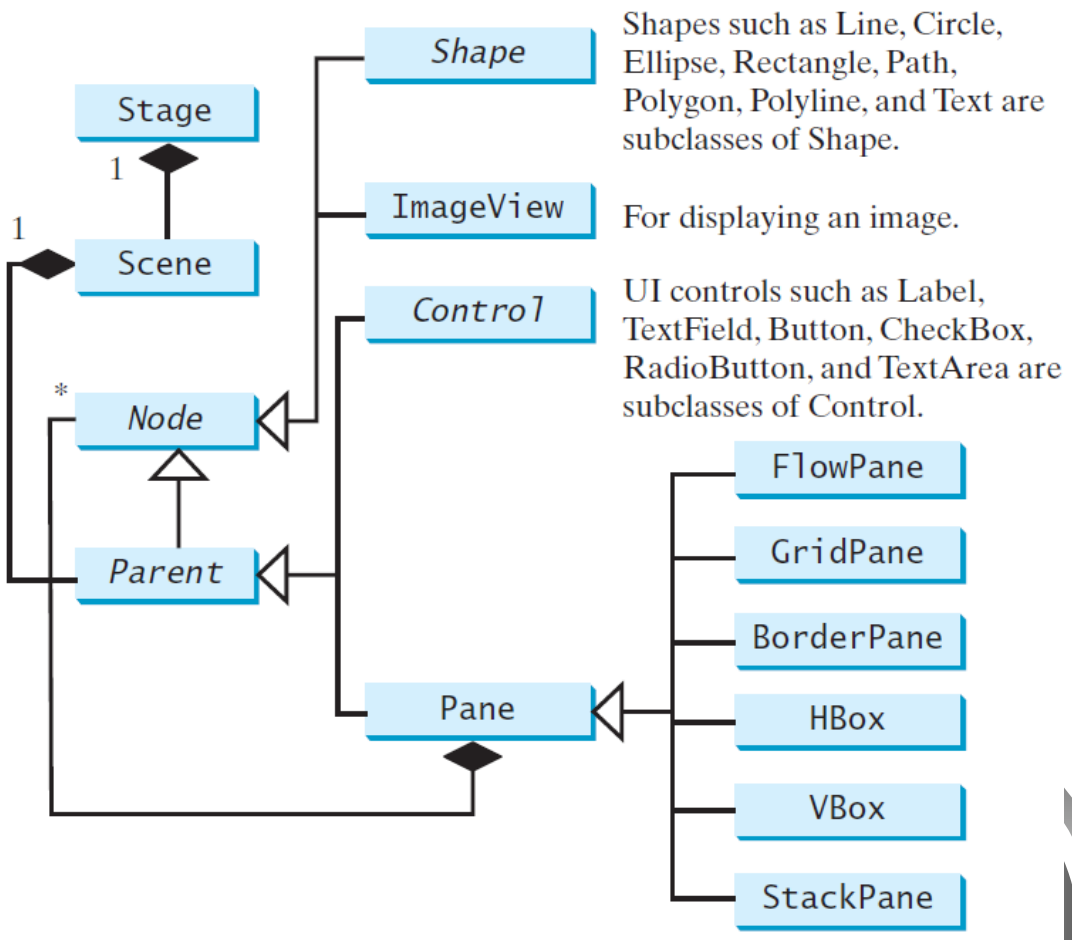
MyJavaFX

MultipleStageDemo

Panes, UI Controls, and Shapes



(a)

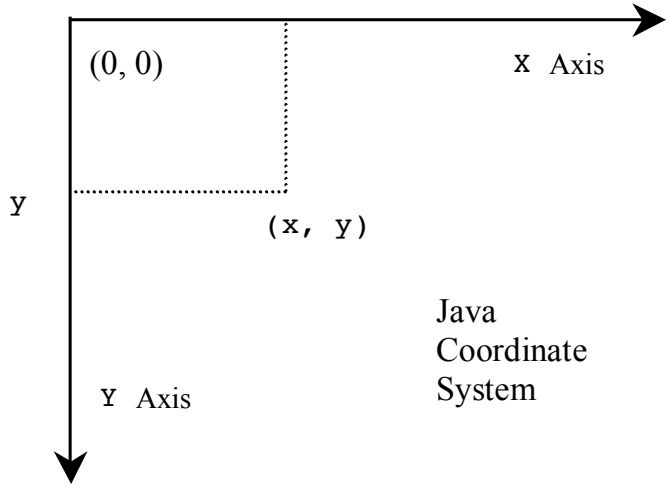


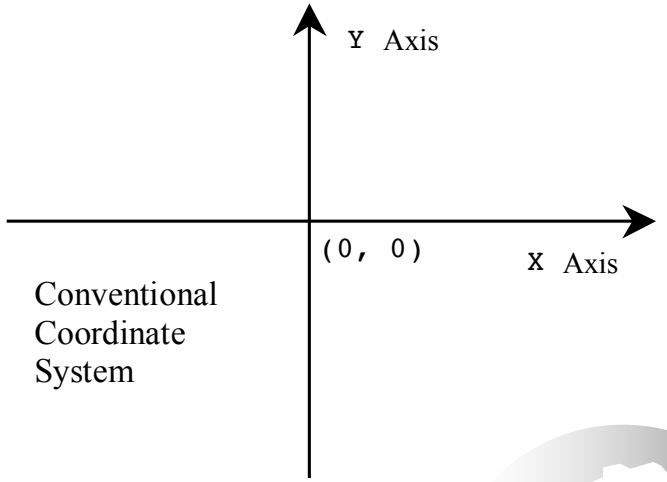
(b)

ButtonInPane

Display a Shape

This example displays a circle in the center of the pane.





ShowCircle

Binding Properties

JavaFX introduces a new concept called *binding property*

- Enables a *target object* to be bound to a *source object*
- If the value in the source object changes, the target property is also changed automatically.
- The target object is simply called a *binding object* or a *binding property*.

ShowCircleCentered

Binding Property: getter, setter, and property getter

```
public class SomeClassName {

  private PropertyType x;

  /** Value getter method */
  public propertyValueType getX() { ... }

  /** Value setter method */
  public void setX(propertyValueType value) { ... }

  /** Property getter method */
  public PropertyType
    xProperty() { ... }
}
```

(a) x is a binding property

```
public class Circle {

  private DoubleProperty centerX;

  /** Value getter method */
  public double getCenterX() { ... }

  /** Value setter method */
  public void setCenterX(double value) { ... }

  /** Property getter method */
  public DoubleProperty centerXProperty() { ... }
}
```

(b) centerX is binding property

Uni/Bidirectional Binding

BindingDemo

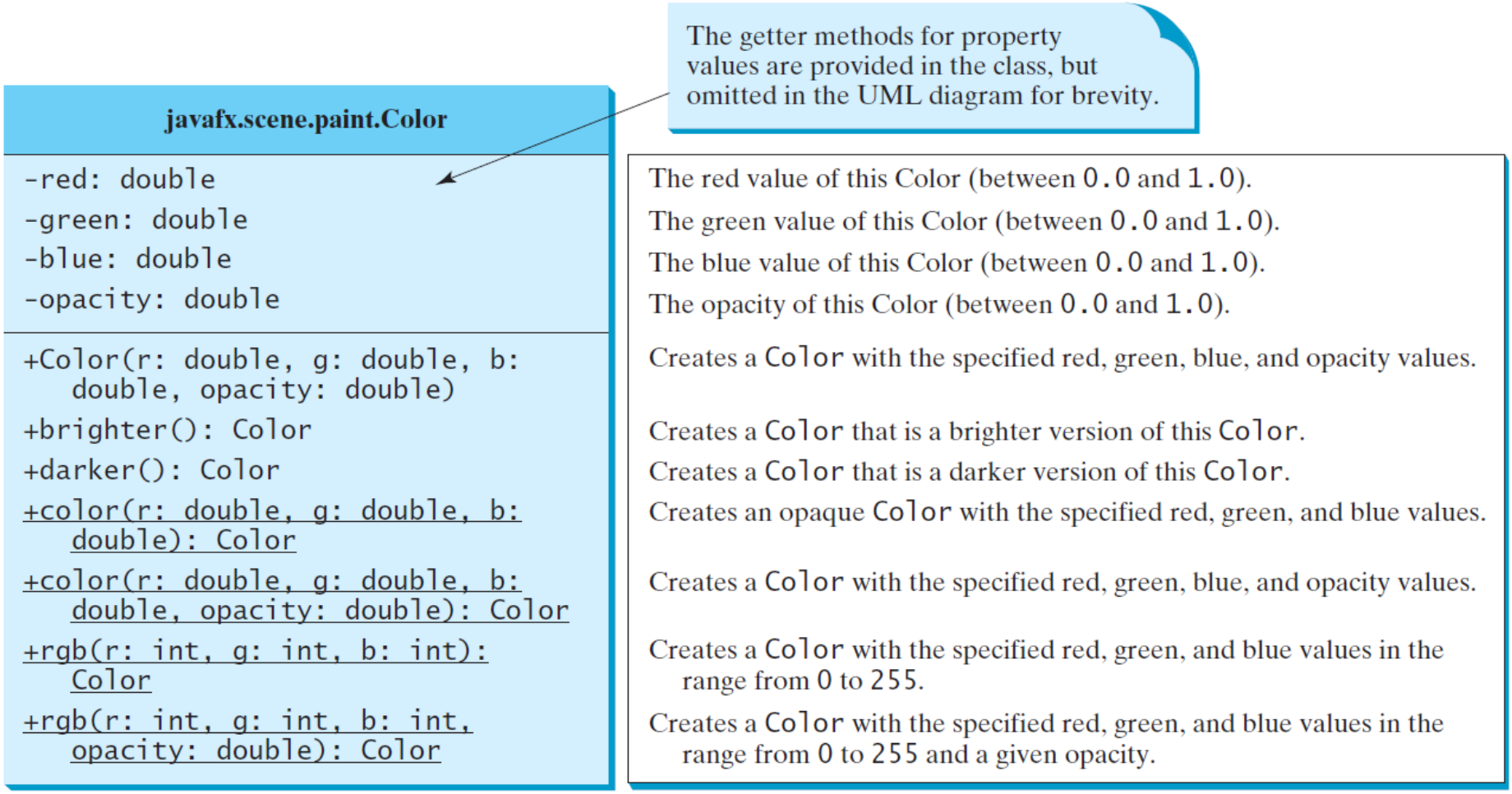
BidirectionalBindingDemo

Common Properties and Methods for Nodes

- style: set a JavaFX CSS style
- rotate: Rotate a node

NodeStyleRotateDemo

The Color Class



The Font Class

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

javafx.scene.text.Font

-size: double	The size of this font.
-name: String	The name of this font.
-family: String	The family of this font.
+Font(size: double)	Creates a Font with the specified size.
+Font(name: String, size: double)	Creates a Font with the specified full font name and size.
+font(name: String, size: double)	Creates a Font with the specified name and size.
+font(name: String, w: FontWeight, size: double)	Creates a Font with the specified name, weight, and size.
+font(name: String, w: FontWeight, p: FontPosture, size: double)	Creates a Font with the specified name, weight, posture, and size.
+getFamilies(): List<String>	Returns a list of font family names.
+getFontNames(): List<String>	Returns a list of full font names including family and weight.

FontDemo

The Image Class

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

javafx.scene.image.Image

```
-error: ReadOnlyBooleanProperty
-height: ReadOnlyBooleanProperty
-width: ReadOnlyBooleanProperty
-progress: ReadOnlyBooleanProperty
+Image(filenameOrURL: String)
```

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.

The ImageView Class

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.

javafx.scene.image.ImageView

```
-fitHeight: DoubleProperty
-fitWidth: DoubleProperty
-x: DoubleProperty
-y: DoubleProperty
-image: ObjectProperty<Image>
```

```
+ImageView()
+ImageView(image: Image)
+ImageView(filenameOrURL: String)
```

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.

ShowImage