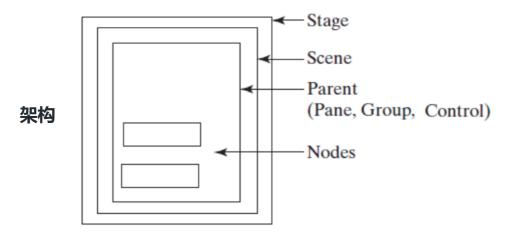
Lecture5 GUI 和 JavaFX

1. JavaFX 介绍

javafx.application.Application 定义了编写 JavaFX 程序的基本框架

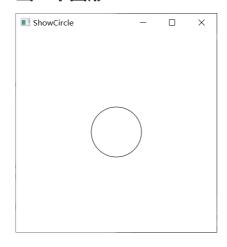
第一个 JavaFX 程序

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;
public class MyJavaFX extends Application { // 继承 Application
    @Override // 重写 Application 类的 start 方法
    public void start(Stage primaryStage) throws Exception {
        StackPane pane = new StackPane(); // 创建一个 Pane
        pane.getChildren().add(new javafx.scene.control.Button("ok")); // 向 Pane
添加一个 btn
        Scene scene = new Scene(pane, 500, 200); // 将 Pane 加入 Scene 里
        primaryStage.setTitle("MyJavaFX");
        primaryStage.setScene(scene); // 将 Scene 加入 Stage 里
        primaryStage.show(); // 展示 Stage
    // main 方法执行 Application
    public static void main(String[] args) {
        Application.launch();
```



- Stage 是一个窗口,它可以展示一些包含了 Node 的 Scene
- Node 可以是 Shape , ImageView , Control , Group 和 Pane
- 一个 JavaFX 程序可以展示多个 Stage

画一个圆形



```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;

public class ShowCircle extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception {

        // Node
        Circle circle = new Circle();
        circle.setCenterX(200);
        circle.setCenterY(200);
        circle.setRadius(50);
        circle.setStroke(Color.BLACK);
```

```
circle.setFill(Color.WHITE);

// Pane
// Pane
// Pane pane = new Pane();
pane.getChildren().add(circle);

// Scene
Scene scene = new Scene(pane, 400, 400);

// Stage
primaryStage.setTitle("ShowCircle");
primaryStage.setScene(scene);
primaryStage.show();

// Stage
primaryStage.show();

// Stage
// Application.launch();

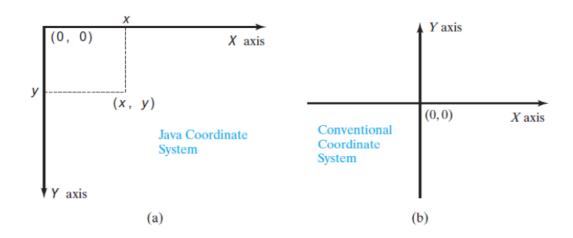
// Application.launch();

// Application.launch();

// Application.launch();

// Pane
// Stage
// PrimaryStage.setTitle("ShowCircle");
// Stage
/
```

• 在 Java 中, 坐标系系统的度量单位是像素, (0,0) 是在左上角



2. Pane 和 Group 的布局

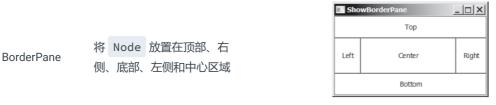
JavaFX提供了多种类型的 Pane ,用于在所需的位置和大小中自动布局 Node

Pane 和 Group 是保存节点的容器

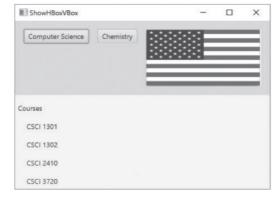
- Group 类通常用于将节点分组,并作为一个组执行转换和扩展
- Pane 和 UI Control 对象是可调整大小的
- Group 、Shape 和 Text 对象是不可调整大小的
- JavaFX 提供了许多类型的 Pane 来组织容器中的 Node



and row indices.



The **BorderPane** places the nodes in five regions of the pane.



The HBox places the nodes in one row, and the VBox places the nodes in one column.

HBox 将 Node 放在单行中

3. JavaFX 相关类

Color 类

Color 类用来创建颜色

JavaFX 定义了一个抽象的 Paint 类来描述绘制一个 Node 的情况, javafx.scene.paint.Color 是 Paint 的一个具体实现子类

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.

除了 RGB 颜色系统之外, Color 类也支持一种 HSB 系统,在 HSB 系统中,一个颜色由 3 种属性定义

	色彩 Hue	饱和度 Saturation	亮度 Brightness
数据类型	double	double	double
取值范围	0.0 - 360.0	0.0 - 1.0	0.0 - 1.0

• 色彩的值是以度来表示的,颜色被看作是沿着一个圆排列的

Color 类提供静态方法 Color.hsb(h,s,b) 和 Color.hsb(h,s,b,a) 来生成 HSB 颜色

```
1 Color randomColor = Color.hsb(360*Math.random(), 1.0, 1.0)
```

RGB系统和HSB系统只是描述同一组颜色的不同方式,可以在一个系统和另一个系统之间进行转换

Font 类

Font 类描述字体名称、字号和大小

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
-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)

(F (N) () | | | | | | |

+getFontNames(): List<String>

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 all font names installed on the user system.

Image 和 ImageView 类

Image 类表示一个图形图像, ImageView 类可以用来显示一个图像

```
1  Image image = new Image("image.gif");
2  ImageView img = new ImageView(image);
3  ImageView img2 = new ImageView("image.gif");
```

javafx.scene.image.Image

-error: ReadOnlyBooleanProperty
-height: ReadOnlyDoubleProperty
-width: ReadOnlyDoubleProperty
-progress: ReadOnlyDoubleProperty

+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.

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.

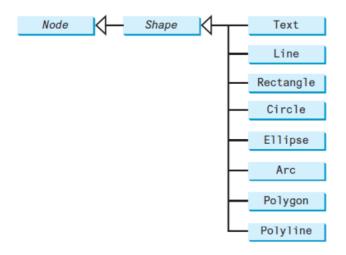
Creates an ImageView.

Creates an ImageView with the specified image.

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

Shape 类

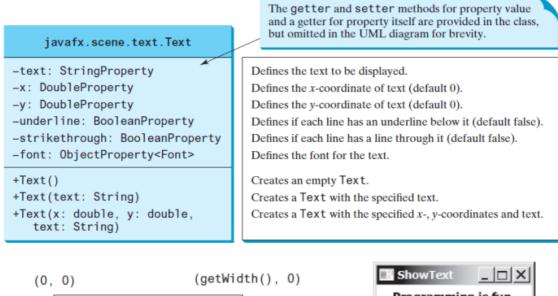
JavaFX 提供了许多 Shape 类,用于绘制文本、线条、圆、矩形、椭圆、圆弧、多边形和折线

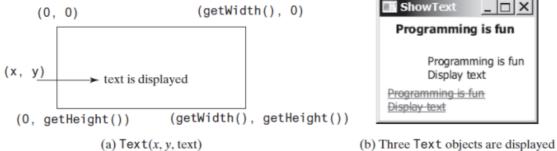


Shape 类是一个抽象基类,它表示了所有形状的公有特性,其中典型的有

fill():指定一个颜色填充形状的内部stroke():指定一个颜色填充形状的轮廓strokeWidth():指定形状边缘的宽度

Text 类





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 value 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.

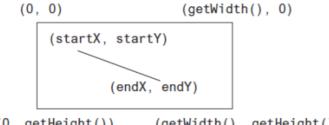
The y-coordinate of the start point.

The x-coordinate of the end point.

The y-coordinate of the end point.

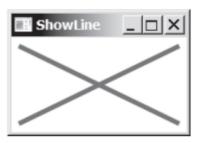
Creates an empty Line.

Creates a Line with the specified starting and ending points.



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

(a) Line(startX, startY, endX, endY)



(b) Two lines are displayed across the pane.

Rectangle 类

javafx.scene.shape.Rectangle

-x: DoubleProperty

-y: DoubleProperty

-width: DoubleProperty

-height: DoubleProperty -arcWidth: DoubleProperty

-arcHeight: DoubleProperty

+Rectangle()

+Rectangle(x: double, y: double, width: double, height: double)

The getter and setter me thods 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).

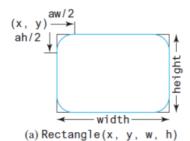
The height of the rectangle (default: 0).

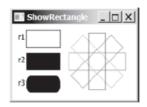
The arcWidth of the rectangle (default: 0). arcWidth is the horizontal diameter of the arcs at the corner (see Figure 14.31a).

The arcWidth 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.





(b) Multiple rectangles are displayed. (c) Transparent rectangles are displayed.

Circle 类

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 类

javafx.scene.shape.Ellipse

-centerX: DoubleProperty
-centerY: DoubleProperty
-radiusX: DoubleProperty
-radiusY: DoubleProperty

+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).

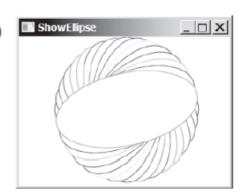
Creates an empty Ellipse.

Creates an Ellipse with the specified center.

Creates an Ellipse with the specified center and radiuses.

radiusX radiusY

(a) Ellipse(centerX, centerY, radiusX, radiusY)



(b) Multiple ellipses are displayed.

Arc 类

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).

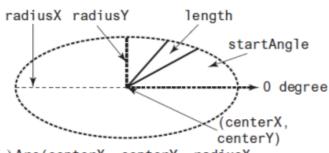
The start angle of the arc in degrees.

The angular extent of the arc in degrees.

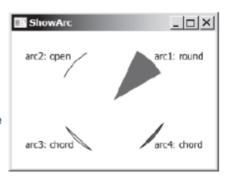
The closure type of the arc (ArcType . OPEN, ArcType . CHORD, ArcType . ROUND).

Creates an empty Arc.

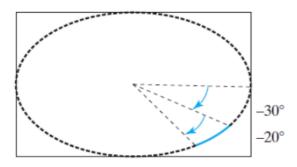
Creates an Arc with the specified arguments.



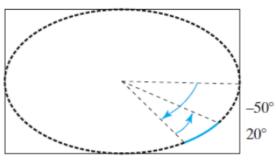
(a) Arc(centerX, centerY, radiusX, radiusY, startAngle, length)



(b) Multiple ellipses are displayed.



(a) Negative starting angle −30° and negative spanning angle −20°



(b) Negative starting angle –50° and positive spanning angle 20°

Angles may be negative.

Polygon 类和 Polyline 类

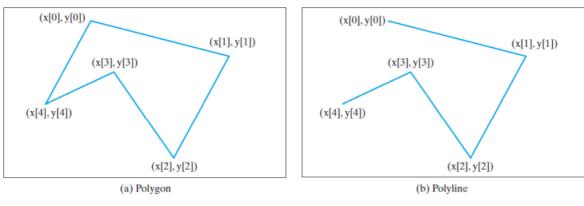
javafx.scene.shape.Polygon

- +Polygon()
- +Polygon(double... points)
- +getPoints():
 - ObservableList<Double>

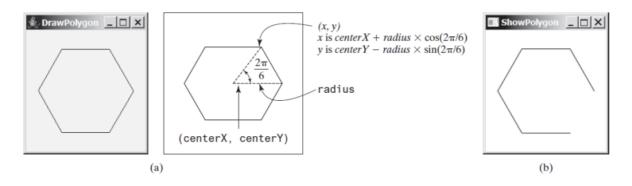
Creates an empty Polygon.

Creates a Polygon with the given points.

Returns a list of double values as x- and y-coordinates of the points.



Polygon is closed and Polyline is not closed.



(a) A Polygon is displayed. (b) A Polyline is displayed.

4. 属性绑定 Property Binding

可以将一个 target 对象绑定在一个 source 对象上

对 source 对象发生的改变将会自动影响到 target 对象

- target 对象也叫做: binding object 或者 binding property
- source 对象也叫做: bindable object 或者 observable object

在第一部分讲到的画一个圆形的代码,现在可以更改部分代码

```
// Scene
// Scene
// Scene scene = new Scene(pane, 400, 400);

// Stage

primaryStage.setTitle("ShowCircle");

primaryStage.setScene(scene);

primaryStage.show();

}
```

下面两条语句完全等价

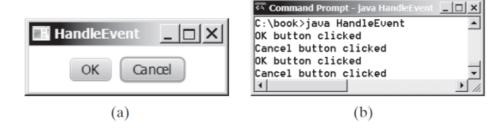
```
1 circle.centerXProperty().bind(pane.widthProperty().divide(2));
```

```
DoubleProperty centerX = circle.centerXproperty();
DoubleProperty width = pane.widthProperty();
centerX.bind(width.divide(2));
```

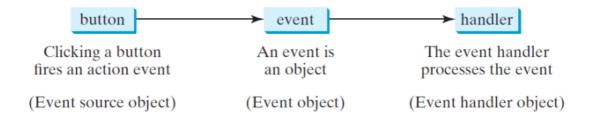
由于 centerX 与 width.divide(2) 绑定, 当 pane 的大小改变的时候, centerX 会自动将自己更新, 来匹配 pane 的宽度

5. 事件驱动的编程

基本介绍



- 程序展示了两个 btn
- 当点击 btn 的时候, 命令行会显示一些消息



当点击 btn 的时候,发送一个事件,这个事件被事件处理器处理

处理事件

并不是所有的对象都可以处理事件,为了能进行处理,必须满足下面两个条件

- 对象必须是 EventHandler<T extends Event> 接口的实例化, T 是 Event 的子类
- EventHandler 对象的 handler 必须与一个产生事件源的对象进行绑定,使用方法 source.setOnAction(hanlder)

注意,由于 EventHandler 接口是一个函数式接口,它里面只有一个抽象方法 handler() ,所以可以写成 lambda 表达式

示例

```
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.HBox;
import javafx.stage.Stage;
public class HandleEvent extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        HBox pane = new HBox(10);
        pane.setAlignment(Pos.CENTER);
        Button bt0k = new Button("OK");
        Button btCancel = new Button("Cancel");
        btOk.setOnAction(event -> System.out.println("Ok btn clicked"));
        btCancel.setOnAction(event -> System.out.println("cancel btn clicked"));
        pane.getChildren().addAll(bt0k, btCancel);
        Scene scene = new Scene(pane);
        primaryStage.setTitle("Handle Event");
        primaryStage.setScene(scene);
        primaryStage.show();
    public static void main(String[] args) {
       Application.launch();
```

Event 类

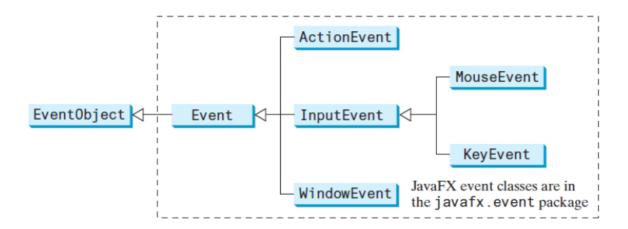


TABLE 15.1 User Action, Source Object, Event Type, Handler Interface, and Handler

User Action	Source Object	Event Type Fired	Event Registration Method
Click a button	Button	ActionEvent	setOnAction(EventHandler <actionevent>)</actionevent>
Press Enter in a text field	TextField	ActionEvent	setOnAction(EventHandler <actionevent>)</actionevent>
Check or uncheck	RadioButton	ActionEvent	setOnAction(EventHandler <actionevent>)</actionevent>
Check or uncheck	CheckBox	ActionEvent	setOnAction(EventHandler <actionevent>)</actionevent>
Select a new item	ComboBox	ActionEvent	setOnAction(EventHandler <actionevent>)</actionevent>
Mouse pressed	Node, Scene	MouseEvent	setOnMousePressed(EventHandler <mouseevent>)</mouseevent>
Mouse released			setOnMouseReleased(EventHandler <mouseevent>)</mouseevent>
Mouse clicked			setOnMouseClicked(EventHandler <mouseevent>)</mouseevent>
Mouse entered			setOnMouseEntered(EventHandler <mouseevent>)</mouseevent>
Mouse exited			<pre>setOnMouseExited(EventHandler<mouseevent>)</mouseevent></pre>
Mouse moved			setOnMouseMoved(EventHandler <mouseevent>)</mouseevent>
Mouse dragged			setOnMouseDragged(EventHandler <mouseevent>)</mouseevent>
Key pressed	Node, Scene	KeyEvent	setOnKeyPressed(EventHandler <keyevent>)</keyevent>
Key released			setOnKeyReleased(EventHandler <keyevent>)</keyevent>
Key typed			setOnKeyTyped(EventHandler <keyevent>)</keyevent>