



# COMP231

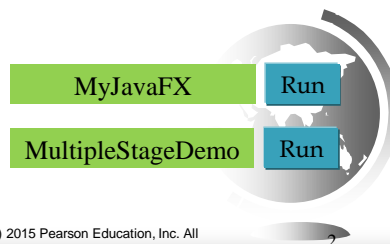
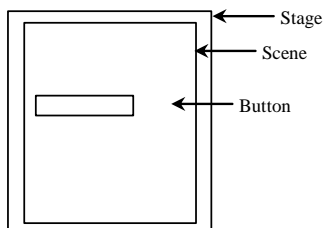
## Advanced Programming

### Chapter 14 JavaFX Basics

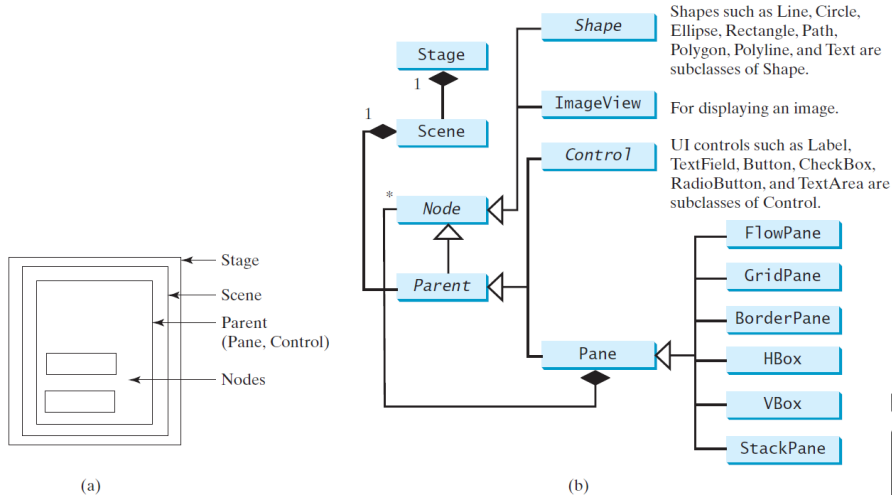
Compiled By: Dr. Majdi Mafarja  
Fall Semester 2017/2018

## Basic Structure of JavaFX

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



# Panes, UI Controls, and Shapes



## ButtonInPane

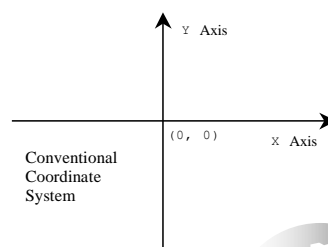
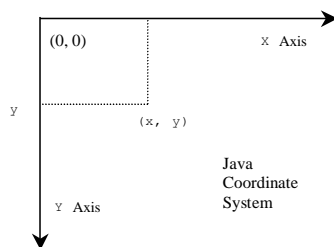
Run

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

3

## Display a Shape

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



ShowCircle

Rur

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

△

# Binding Properties

JavaFX introduces a new concept called *binding property* that 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

Run

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

5

## 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

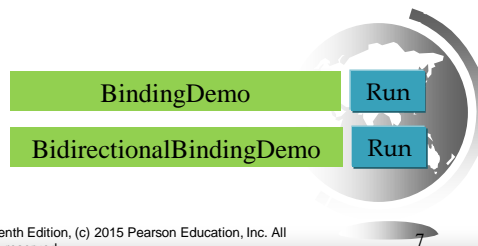
```
public class Circle {  
    private DoubleProperty centerX;  
    /** Value getter method */  
    public double getCenterX() { ... }  
    /** Value setter method */  
    public void setCenterX(double value) { ... }  
    /** Property getter method */  
    public DoubleProperty centerXProperty() { ... }  
}
```

(b) centerX is binding property

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

6

# Uni/Bidirectional Binding



Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

7

# Common Properties and Methods for Nodes

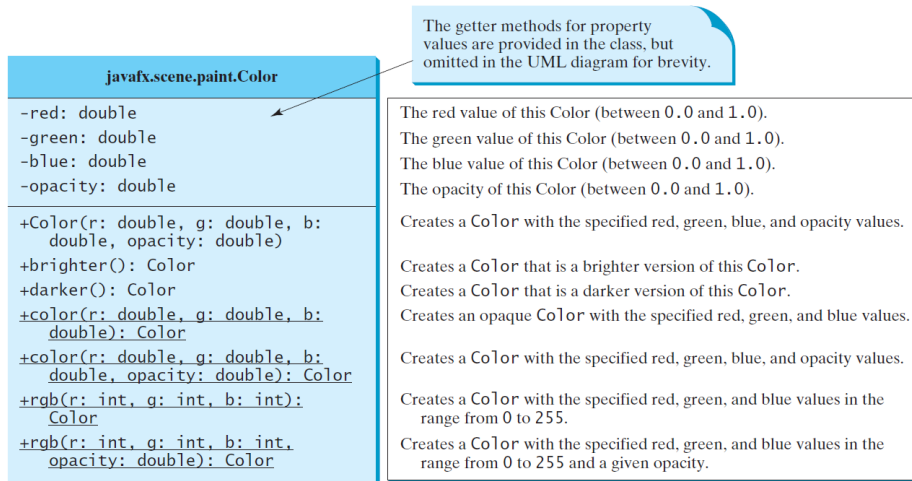
- style: set a JavaFX CSS style
- rotate: Rotate a node



Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

8

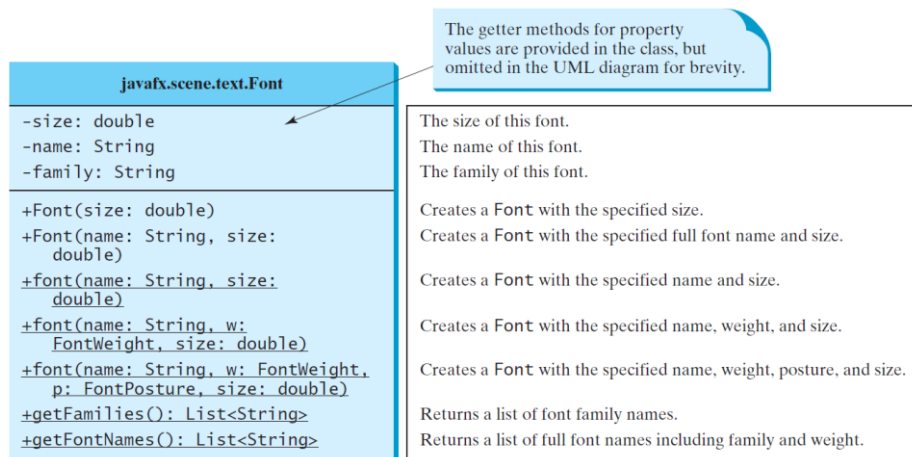
# The Color Class



Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

9

# The Font Class



FontDemo

Run

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

10

# 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.



Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

11

# 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.

ShowImage

Run



Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

12

# Layout Panes

JavaFX provides many types of panes for organizing nodes in a container.

Class	Description
<b>Pane</b>	Base class for layout panes. It contains the <code>getChildren()</code> method for returning a list of nodes in the pane.
<b>StackPane</b>	Places the nodes on top of each other in the center of the pane.
<b>FlowPane</b>	Places the nodes row-by-row horizontally or column-by-column vertically.
<b>GridPane</b>	Places the nodes in the cells in a two-dimensional grid.
<b>BorderPane</b>	Places the nodes in the top, right, bottom, left, and center regions.
<b>HBox</b>	Places the nodes in a single row.
<b>VBox</b>	Places the nodes in a single column.

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

13

## FlowPane

`javafx.scene.layout.FlowPane`

-alignment: `ObjectProperty<Pos>`  
-orientation: `ObjectProperty<Orientation>`  
-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.

MultipleStageDemo

Run

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

14

# GridPane

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.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  
+setHalignment(child: Node, value: HPos): void  
+setValignment(child: Node, value: VPos): void

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.

ShowGridPane

Run

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

15

# BorderPane

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.layout.BorderPane**

-top: ObjectProperty<Node>  
-right: ObjectProperty<Node>  
-bottom: ObjectProperty<Node>  
-left: ObjectProperty<Node>  
-center: ObjectProperty<Node>

+BorderPane()  
+setAlignment(child: Node, pos: Pos)

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.

ShowBorderPane

Run

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

16



# HBox

**javafx.scene.layout.HBox**

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

+HBox()  
+HBox(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 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.

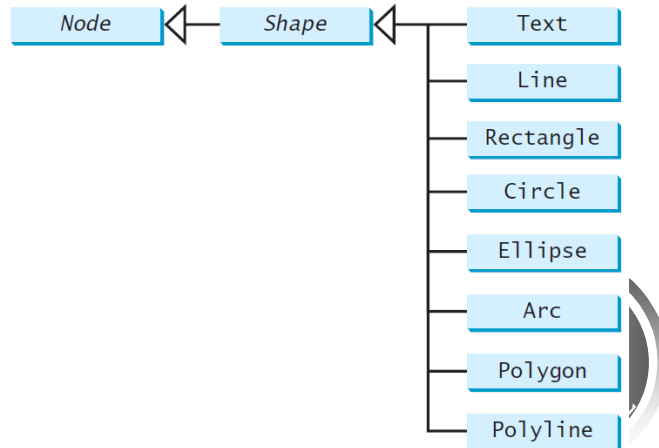


ShowHBoxVBox

Run

# Shapes

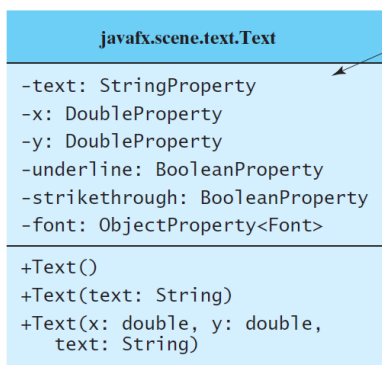
JavaFX provides many shape classes for drawing texts, lines, circles, rectangles, ellipses, arcs, polygons, and polylines.



Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

19

# Text



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.

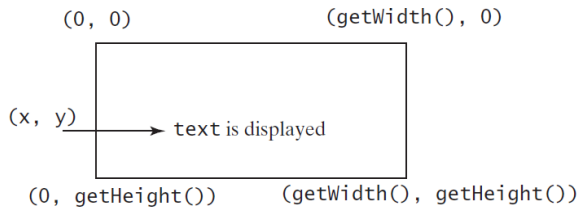
Defines the text to be displayed.  
 Defines the x-coordinate of text (default 0).  
 Defines the y-coordinate of text (default 0).  
 Defines if each line has an underline below it (default false).  
 Defines if each line has a line through it (default false).  
 Defines the font for the text.

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

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

20

# Text Example



(a) Text(x, y, text)



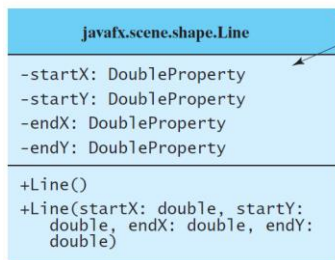
(b) Three Text objects are displayed



Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

21

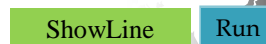
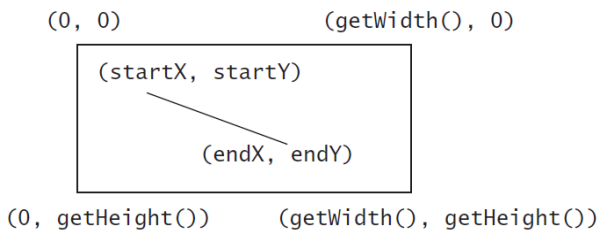
# Line



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.



Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

22

# Rectangle

`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 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).

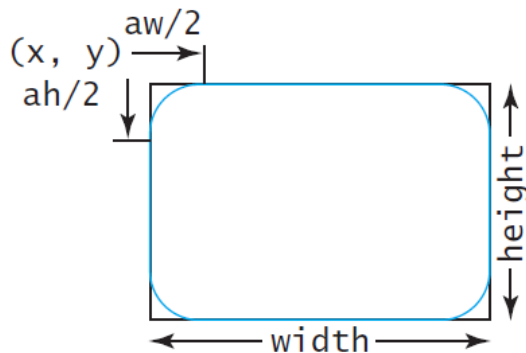
The `arcWidth` of the rectangle (default: 0). `arcWidth` is the horizontal diameter of the arcs at the corner (see Figure 14.31a).

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.

## Rectangle Example

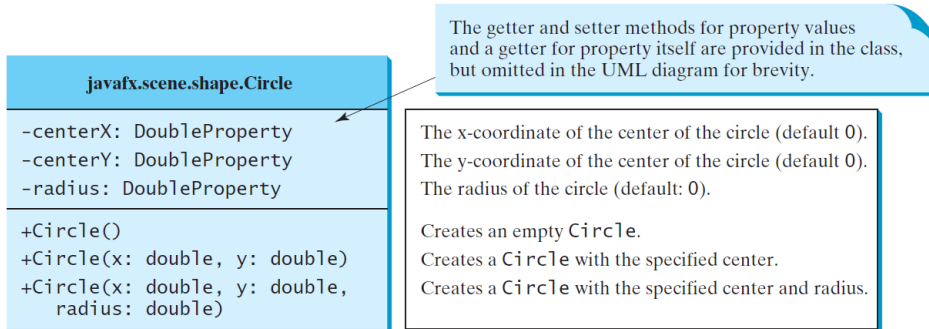


(a) `Rectangle(x, y, w, h)`

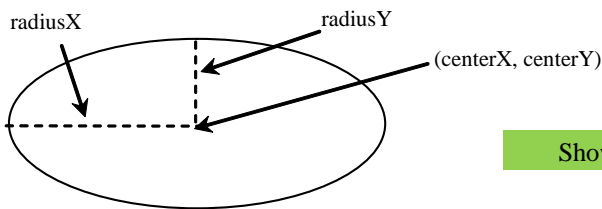
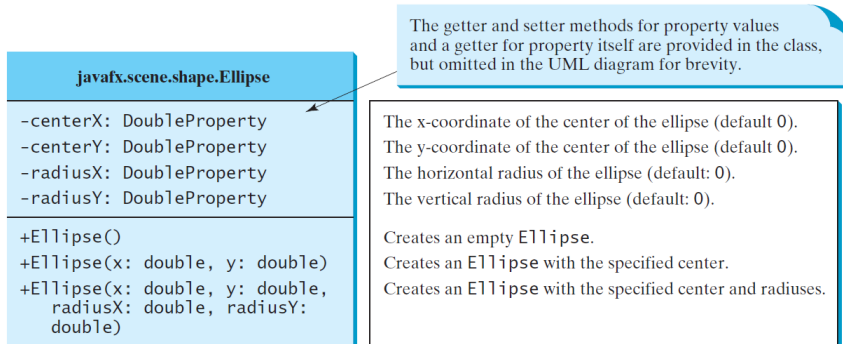
ShowRectangle

Run

# Circle



# Ellipse



ShowEllipse

Run



# Arc

**javafx.scene.shape.Arc**

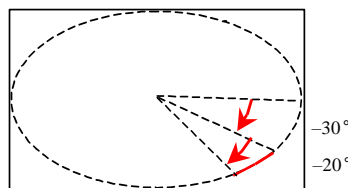
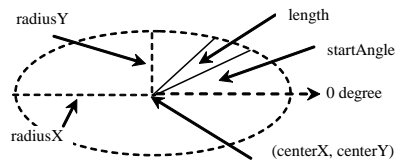
```
-centerX: DoubleProperty
-centerY: DoubleProperty
-radiusX: DoubleProperty
-radiusY: DoubleProperty
-startAngle: DoubleProperty
-length: DoubleProperty
-type: ObjectProperty<ArcType>

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

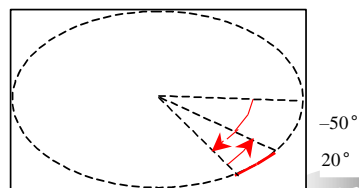
The getter and setter methods for property values and a getter for property itself are provided in the class, but omitted in the UML diagram for brevity.

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.

## Arc Examples



(a) Negative starting angle  $-30^\circ$  and negative spanning angle  $-20^\circ$

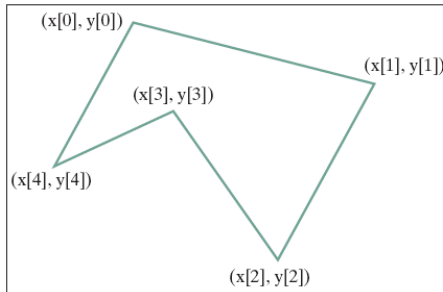


(b) Negative starting angle  $-50^\circ$  and positive spanning angle  $20^\circ$

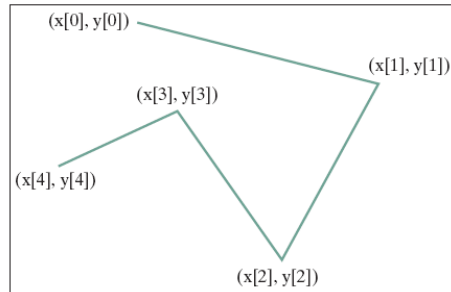
ShowArc

Run

# Polygon and Polyline



(a) Polygon



(b) Polyline



Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

29

# Polygon

javafx.scene.shape.Polygon
<pre>+Polygon () +Polygon (double... points) +getPoints () :     ObservableList&lt;Double&gt;</pre>

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.

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.

ShowPolygon

Run

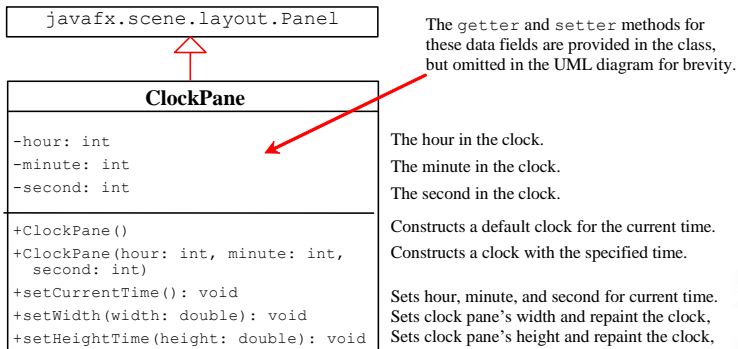


Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

30

# Case Study: The ClockPane Class

This case study develops a class that displays a clock on a pane.



ClockPane

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

31

## Use the ClockPane Class

DisplayClock

Run

Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.

32