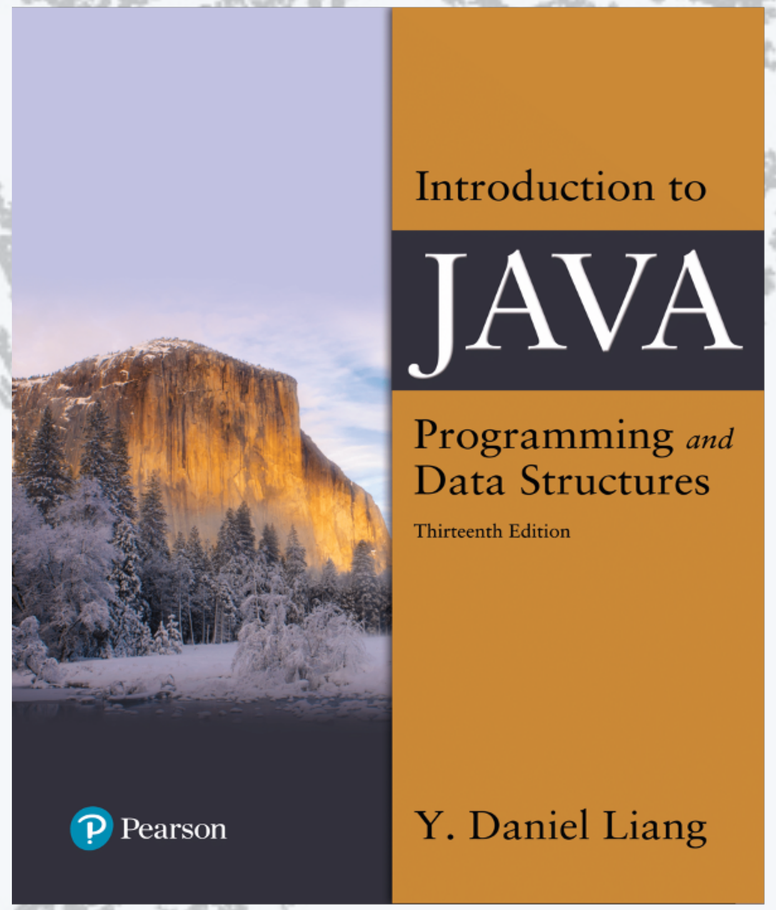


JavaFX Basics

JavaFX Basics

- JavaFX Basics
-

GUI Applications

- So far, you've probably only worked on console applications
  - Provide input from keyboard
  - Read input using java.util.Scanner
  - Do something
  - Print result to System.out
- It'll be nice to have a **GUI application**
  - Examples: Microsoft Word, Apps on your phone, Your browser
- We cover the basic material before covering GUI programming because it requires use of all the basic knowledge you've learned so far
- We will cover **JavaFX** in this class

Why JavaFX

- JavaFX is a new framework for developing Java GUI Programs
  - Graphical functionality is provided by the library, no need to write your own
- Some Java History
  - Ancient code: *AWT*
  - Until Java 7: *Swing* (Will never die, most current application still use it)
  - Java 8 and later: ***JavaFX***
  - Do not use Swing/AWT examples copied from online sources
- Good way to review and use all the knowledge you acquired so far
  - Object Oriented Programming

JavaFX HelloWorld

- **MyJavaFX**

```
//import javafx.scene.control.Button, not java.awt.Button!!!!!
public class HelloWorld extends Application {
	public static void main(String[] args) {
		launch(args);
	}
	// Override the start method in the Application class
	@Override
	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
		primaryStage.show();
	}
}
```

JavaFX HelloWorld Example: Controls

```
public class HelloWorld extends Application {
    public static void main(String[] args) {
        launch(args);
    }
    // Override the start method in the Application class
    @Override
    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
        primaryStage.show();
    }
}
```

- Starting Point of a JavaFX application
- Main Method can be omitted when running from console/with e(fx)clispe installed
- A primary stage is created automatically

Some Terminologies

- Stage
  - Represents windows, top level container
  - Many setter methods: setTitle(), setWidth()
  - You can create multiple stages and use one or another
- Scene
  - Each stage has a scene
  - Scene holds controls (buttons, labels, etc)
- Pane
  - You can put controls in Scenes directly, but we usually Panes for better layout
  - Examples: StackPane, BorderPane, HBox, VBox

JavaFX HelloWorld Example: Creating the Stage

```
public class HelloWorld extends Application {
    public static void main(String[] args) {
        launch(args);
    }
    // Override the start method in the Application class
    @Override
    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
        primaryStage.show();
    }
}
```

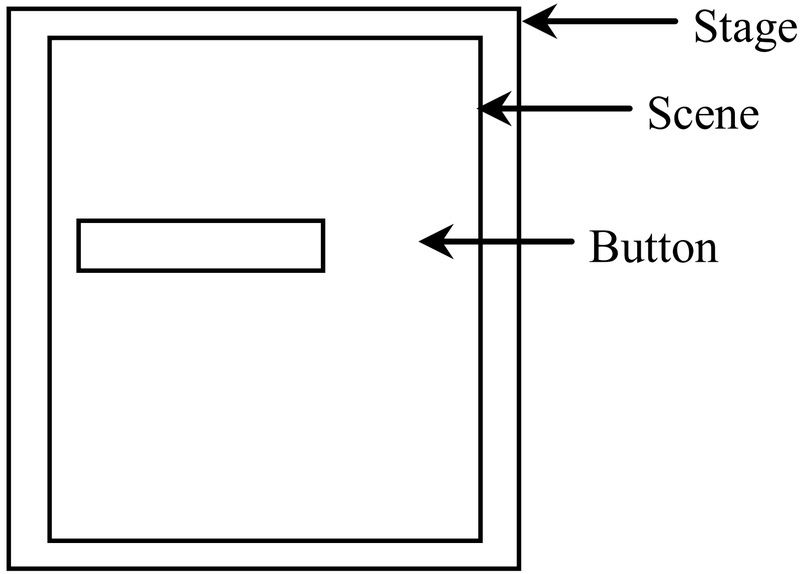
- Place the scene in the Stage
- Stage.show() makes window appear

JavaFX Example: Multiple Stages

```
public class MultiStageDemo extends Application {
	public static void main(String[] args) {
		launch(args);
	}
	// Override the start method in the Application class
	@Override
	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
		primaryStage.show();
		Stage secondStage = new Stage();
		secondStage.setTitle("Second Stage");
		secondStage.setScene(new Scene(new Label("This is exmaple of label")));
		secondStage.show();
	}
}
```

Basic Structure

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



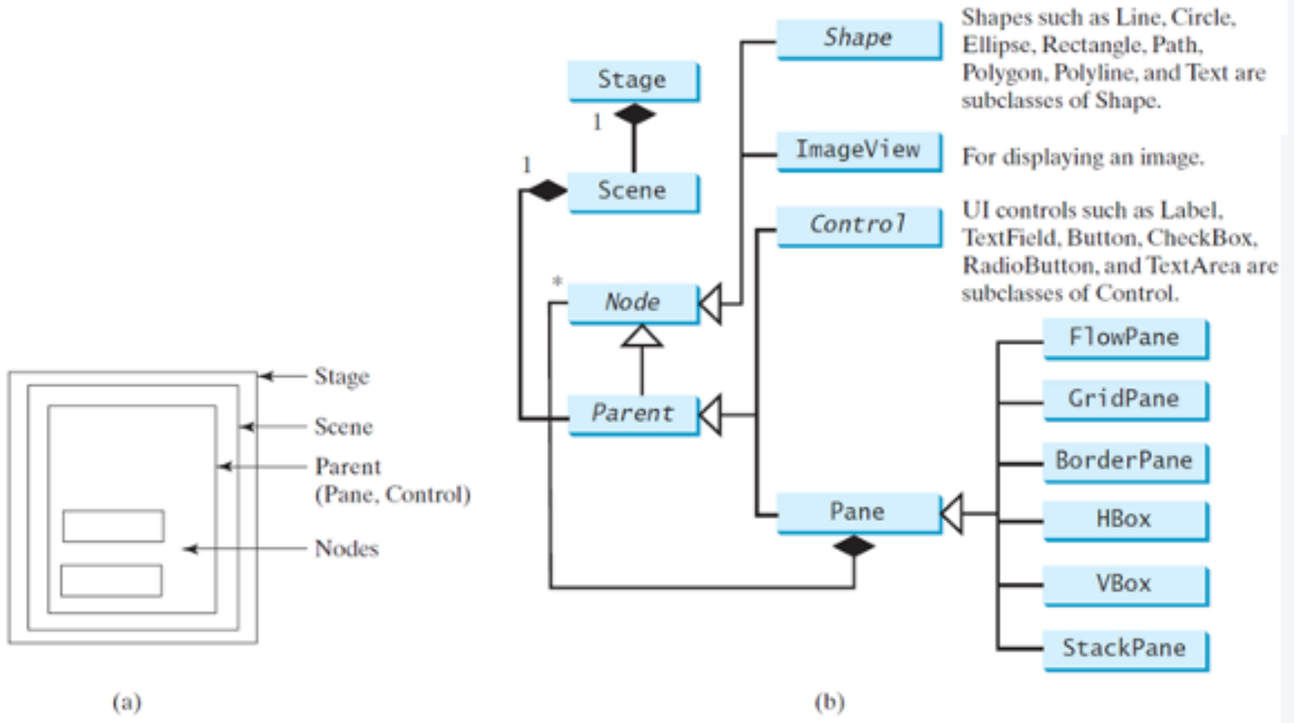
Layout Panes

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

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

Using Panes

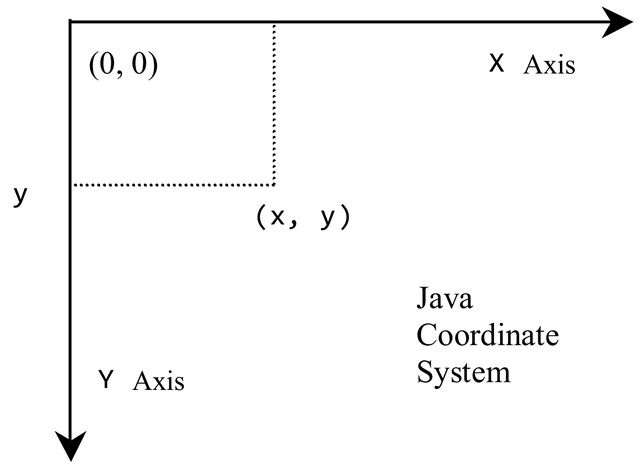
- Reading: http://docs.oracle.com/javafx/2/layout/builtin_layouts.htm
- Familiar yourself with the built in layouts
- You can mix and match different types of layouts
  - Panes are also nodes
  - You can have an HBox in a BorderPane, a VBox in a StackPane, etc

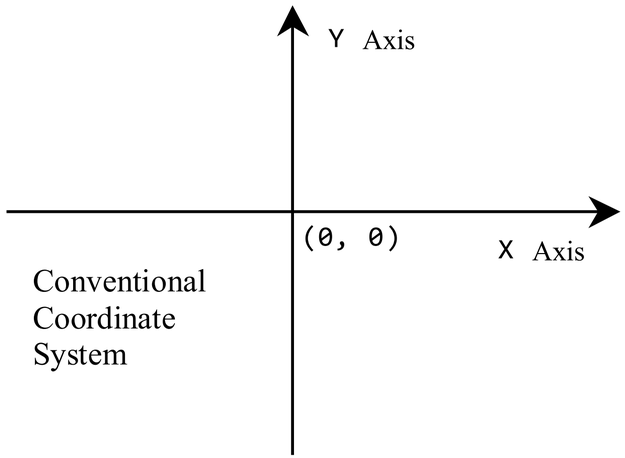


Display a Shape

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

Final Exam Info

- Final Exam Info

Final Exam Arrangement

- 8 **May, 10:30-12:30 (Wed)** in total 120 minutes
- Liwen Hall (礼文堂)
- **Closed-book exam**
- **No electronic devices (including calculators)** are allowed

- Scope: from 1.Basics to 11.ExceptionHandling

- Around 20% choice questions
- Around 50% read the program, give the expected output/complete the program
- Around 30% short answer questions
- The above are rough numbers, the proportion can be changed.