Java Programming

Arthur Hoskey, Ph.D. Farmingdale State College Computer Systems Department



Today's Lecture

Graphical User Interface (GUI) for Java

- Create windows type programs
- You can use buttons, textboxes, labels etc.

JavaFX is platform independent

- The code will run on both Windows and Linux platforms
- Other languages such as Visual Basic can only be run on one platform.
- A JavaFX program can be written on a Windows machine and run on a Linux machine (assuming the correct Java runtime is installed on both machines).



- File|New|Project. This will open the New Project dialog box.
- Choose JavaFX on the left.
- Use Maven as the Build system.
- Click Next. On the page that appears click Create.

	New Project			×	
	Q				
	New Project	Name: HelloJ	зvaFX		
	Empty Project	Location: C:\Hw	k		
	Generators	Project	will be created in: C:\Hwk\HelloJavaFX		
	Maven Archetype	Cre	ate Git repository		
	🖿 JavaFX		Kallin Carrie		
	Kotlin Multiplatform	Language: Java	Kotiin Groovy		
	Compose Multiplatform	Build system: May	en Gradle		
	IDE Plugin	Ground	vamula		
	🛎 Android	Group: Com.e	tampie		
		Artifact: HelloJ	avaFX		
			6 Oraçla Onan IDK varrian 16		
	2			Novt	
L	U			Cancel	
112.7				EV D-	
		2211	e Java		



• If you run the project as is, you will see the following:



Run the Project – Shows Window

- Scene Builder is an application used to edit .fxml files.
- You can drag and drop controls inside Scene Builder and make the window look the way you want it to.
- You must inform IntelliJ of the location of SceneBuilder.exe. If you do this then you will be able to open .fxml files in Scene Builder from inside IntelliJ.
- For example...

Scene Builder

- From IntelliJ, go to File|Settings.
- Choose Languages & Frameworks on the left. Press JavaFX on the right.
- Set the full path name of SceneBuilder.exe.



Intellij – Setup Intellij to Open .fxml Files in Scene Builder

Right click the .fxml file and choose Open in Scene Builder from the menu. Ele Edit View Nevigate Code Befactor Build Run Tools VCS Window Help HelioJavaFX-HelioAp HelioJavaFX (main) resources com) example' heliojavaFX (main)



Scene Builder

There are multiple tabs of items on the left that you can add to the window

Controls that you can put in the window





You can drag a label on to the window.

When you click the label its properties appear in the right side window.



The properties of the selected control appear in the window on the right

There are lots of values that can be updated. For example, the font, color, and text that appears on the label just to name a few

Add Control and Show Properties

You must set the fx:id (basically a variable name) of a control if you want to use it in code.



• A button was added below the label. The button's text was changed to "Show Message".







Add Button Event Handler

- The controller class contains the code that will run in the background.
- IntelliJ automatically creates a controller class for us when the project is created.
- The GUI event handlers for one screen go in one controller class that is associated with that screen.



• Set the button's event handler by setting the .



Set Button Event Handler

- When a button is pressed the method listed in On Action will be called automatically.
- You can put as much code as you want in the event handler (handleShowMessageButton() in this case).

public class PrimaryController {

```
@FXML
private void handleShowMessageButton()
{
    System.out.println("I love JavaFX");
}
// Other controller code is here...
}
The message will be
printed in the console
whenever the button
is pressed.
```

Add Code to Event Handler



 The following screenshots show before and after pressing the Show Message button:



Update Control in Code

End of Slides

End of Slides