Java Programming

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

Today's Lecture

- Java FX Overview

JavaFX

- JavaFX definition from the Oracle website:
- JavaFX is a set of graphics and media packages that enables developers to design, create, test, debug, and deploy rich client applications that operate consistently across diverse platforms.
- Link:

http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784

JavaFX

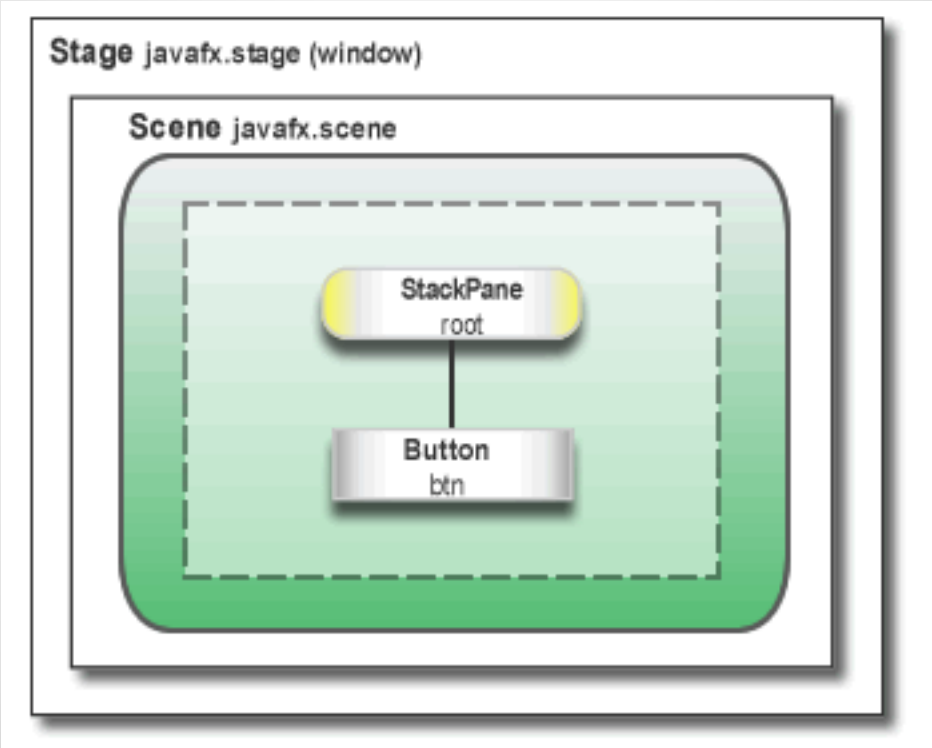
- Graphical User Interface (GUI) for Java
  - Create windows type programs
  - You can use buttons, textboxes, labels etc.
- JavaFX is platform independent
  - The code will run on both Windows and Linux platforms
  - Other languages such as Visual Basic can only be run on one platform.
  - A JavaFX program can be written on a Windows machine and run on a Linux machine (assuming the correct Java runtime is installed on both machines).

- Now a description of the components of a simple JavaFX application…

JavaFX Hierarchy

JavaFX Hierarchy

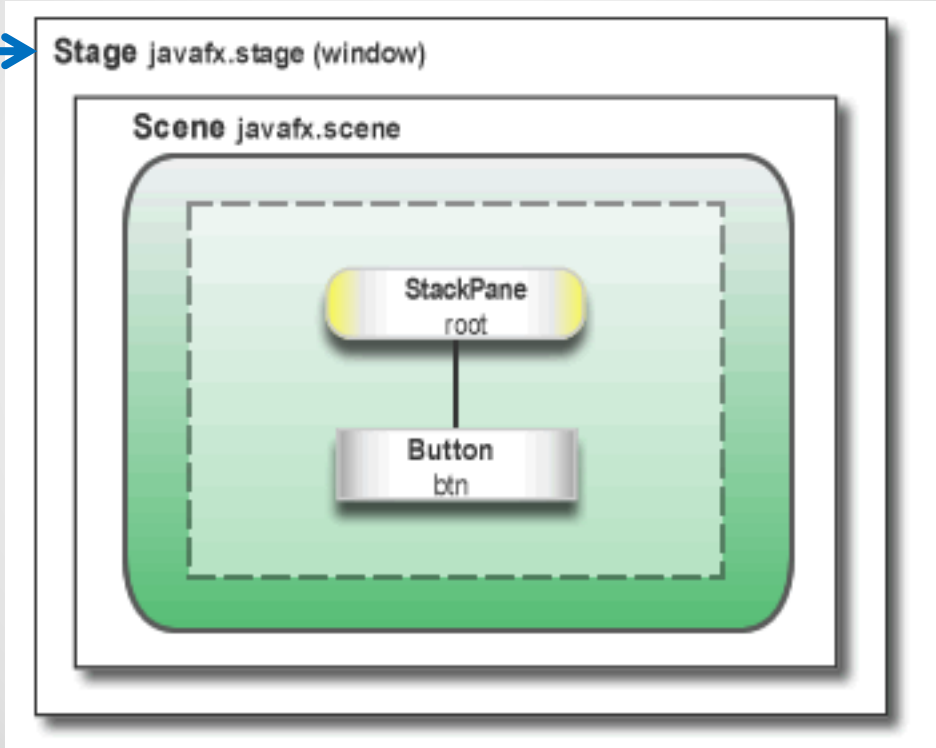
- Here is the JavaFX Hierarchy for a simple application that has a window and a button.
- This diagram shows the relationship between the main parts of a JavaFX application (taken from Oracle docs).



Stage

- Basically a window.
- Top-level JavaFX container.

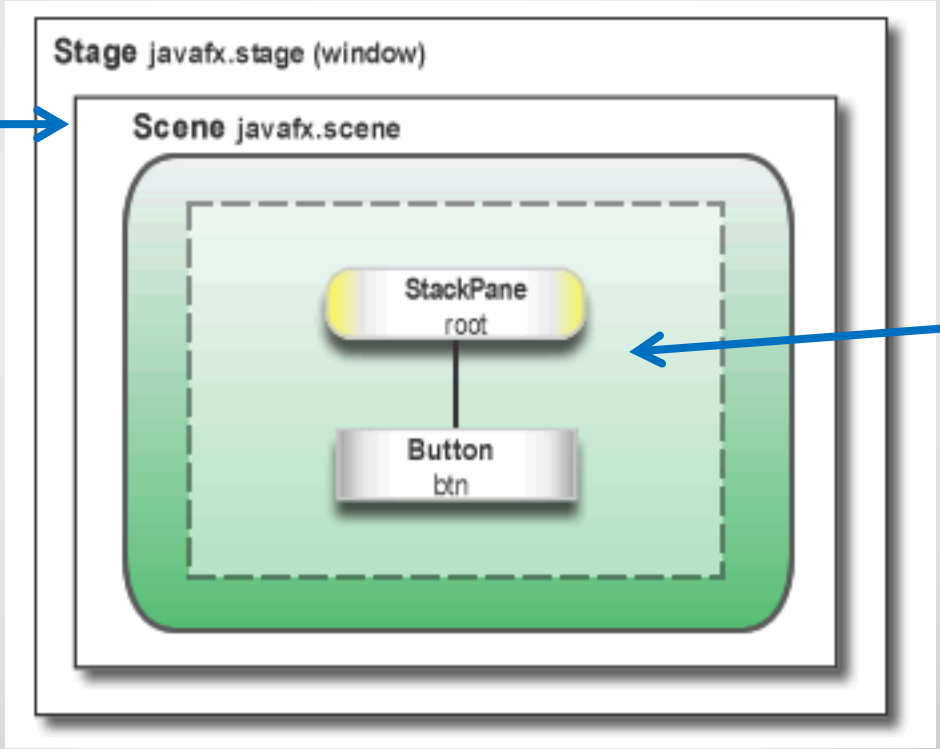
Stage →



JavaFX Stage

Scene

- The stage can have different scenes placed inside of it.
- The scene graph contains the GUI's layouts and controls (inside the dotted line).

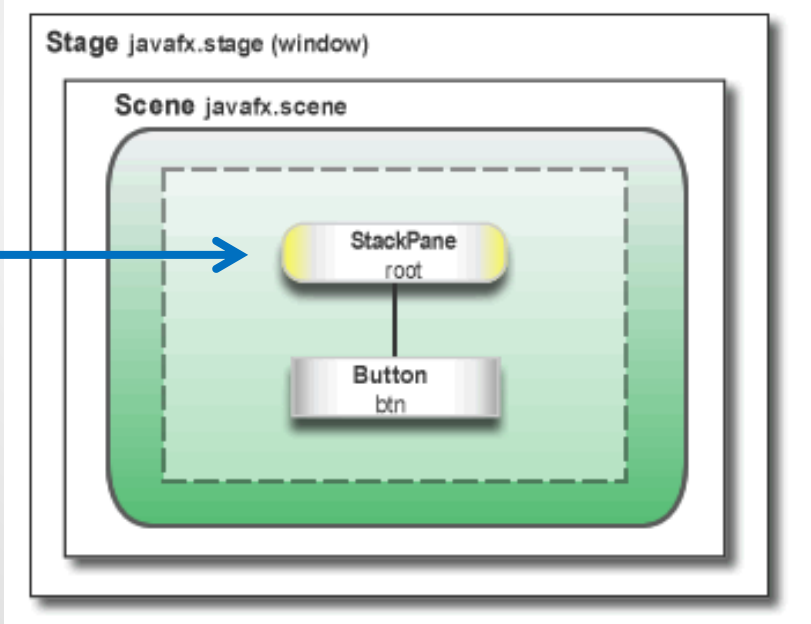


JavaFX Scene

Scene Graph

- A StackPane is the root node of the scene graph in this example.
- The StackPane contains and arranges child controls (one button in this case).
- Can use other types of root nodes to contain/arrange controls (VBox, Hbox, BorderPane, etc…).

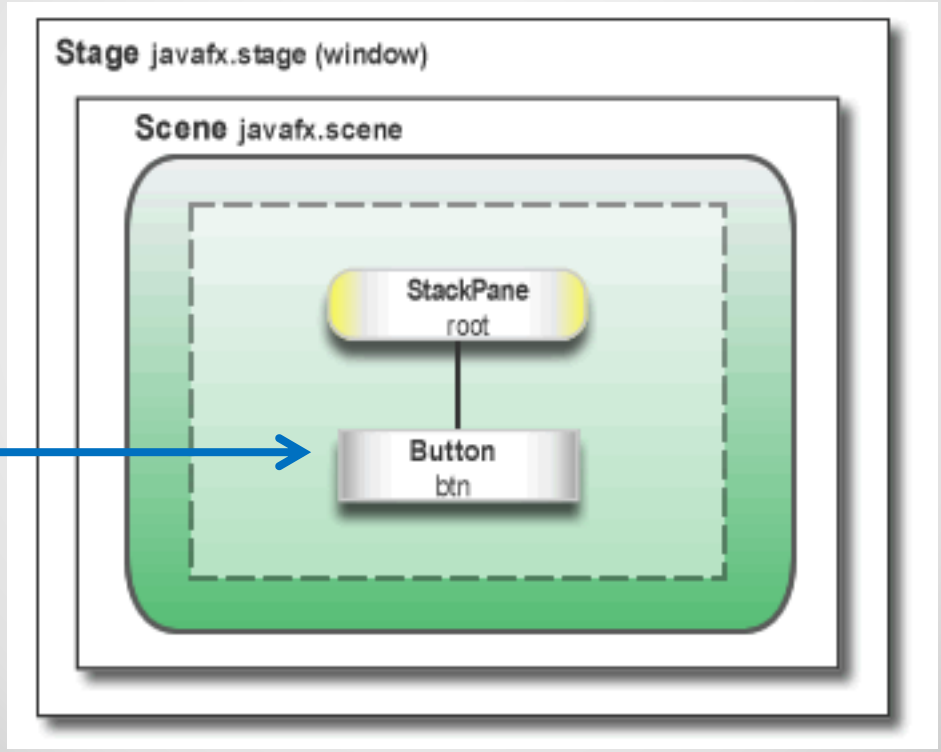
StackPane (root of scene graph in this example)



JavaFX Scene Graph

Button

- A button can be pressed and generate an event.
- The application can then respond to the event (do something interesting).



JavaFX Button

FXML

- XML used for JavaFX.
- Used to define user interfaces in JavaFX.
- Using FXML to define GUI is good design because it separates the presentation of data from the application logic.
- This is good because changes in either will minimally effect the other.

- Link – Why Use FXML?:

https://docs.oracle.com/javase/8/javafx/fxml-tutorial/why_use_fxml.htm

- Link - FXML:

https://docs.oracle.com/javase/8/javafx/api/javafx/fxml/doc-files/introduction_to_fxml.html#overview

FXML

Java Annotations

- Provide data about a program that is not part of the program itself.
- Annotations have no direct effect on the operation of the code they annotate.

- For example:

```
@FXML
private Label tipPercentageLabel;
```

- The @FXML annotation connects Java code to FXML code. In this case, an FXML Label is being associated with a Java variable.
- You can also associate FXML controls with Java event handlers.

- Taken from:

https://docs.oracle.com/javase/tutorial/java/annotations/

Java Annotations

Java Annotations

Java Annotation Uses:

- **Information for the compiler** — Annotations can be used by the compiler to detect errors or suppress warnings.
- **Compile-time and deployment-time processing** — Software tools can process annotation information to generate code, XML files, and so forth.
- **Runtime processing** — Some annotations are available to be examined at runtime.

- Taken from:

https://docs.oracle.com/javase/tutorial/java/annotations/

End of Slides