JDK 8 – JDK – 10 Follow these steps
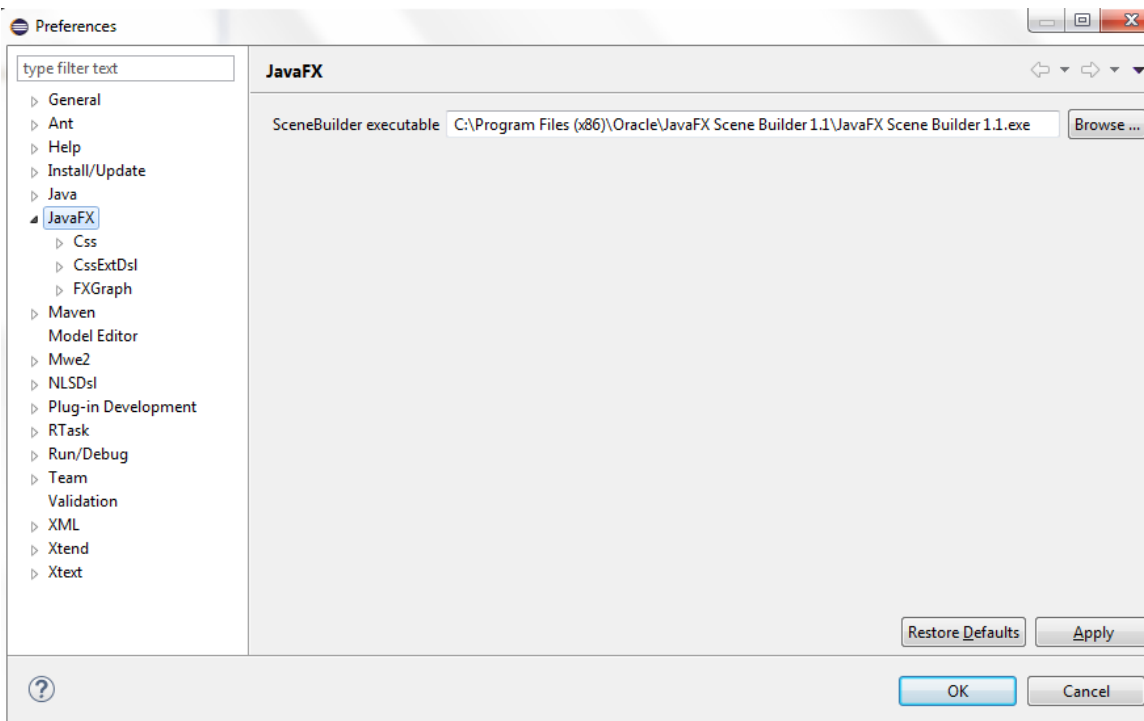
1. Download JavaFX Scene Builder from: http://www.oracle.com/technetwork/java/javafxscenebuilder-1x-archive-2199384.html

2. Change the path to Scene Builder in Eclipse to installation location, by clicking windows->preferences->JavaFX tab. If you install JavaFX Scene Builder on a 64-bit Windows machine, the default installation location is C:\Program Files (x86)\Oracle\JavaFX Scene Builder (version)

From JDK 10+ Follow the below procedure

Configuring JavaFX in Eclipse

1. Download JavaFX Scene-Builder from the following link and install it on your machine. https://www.oracle.com/java/technologies/javafxscenebuilder-1x-archive-downloads.html

2. In Eclipse from menu Help -> Eclipse Marketplace search for "e(fx)clipse" and install it and restart the Eclipse.
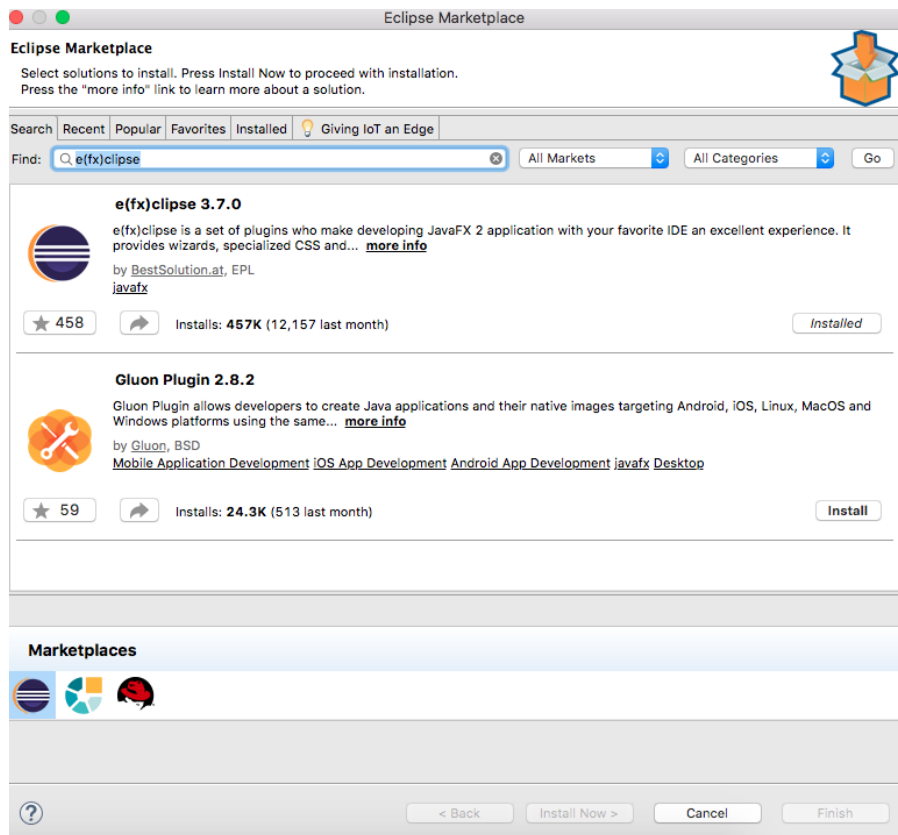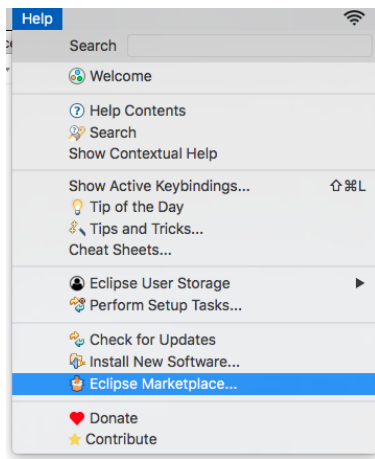
3. Download JavaFX SDK from the following URL and download it.

    https://gluonhq.com/products/javafx/

Product	Version	Platform	Download
JavaFX Windows x64 SDK	16	Windows x64	Download [SHA256]
JavaFX Windows x64 jmods	16	Windows x64	Download [SHA256]
JavaFX Windows x86 SDK	16	Windows x86	Download [SHA256]
JavaFX Windows x86 jmods	16	Windows x86	Download [SHA256]
JavaFX Mac OS X SDK	16	Mac	Download [SHA256]
JavaFX Mac OS X jmods	16	Mac	Download [SHA256]
JavaFX Linux SDK	16	Linux	Download [SHA256]
JavaFX Linux jmods	16	Linux	Download [SHA256]
JavaFX Documentation	16	Javadoc	Download [SHA256]

4. Unzip the file and place the **javafx-sdk-XX** folder in some static path in your local machine.

5. In Eclipse from menu Preferences click on JavaFX and paste the path where you placed the JavaFX SDK in the field named **JavaFX 11 + SDK** including the lib folder. **Click APPLY.**
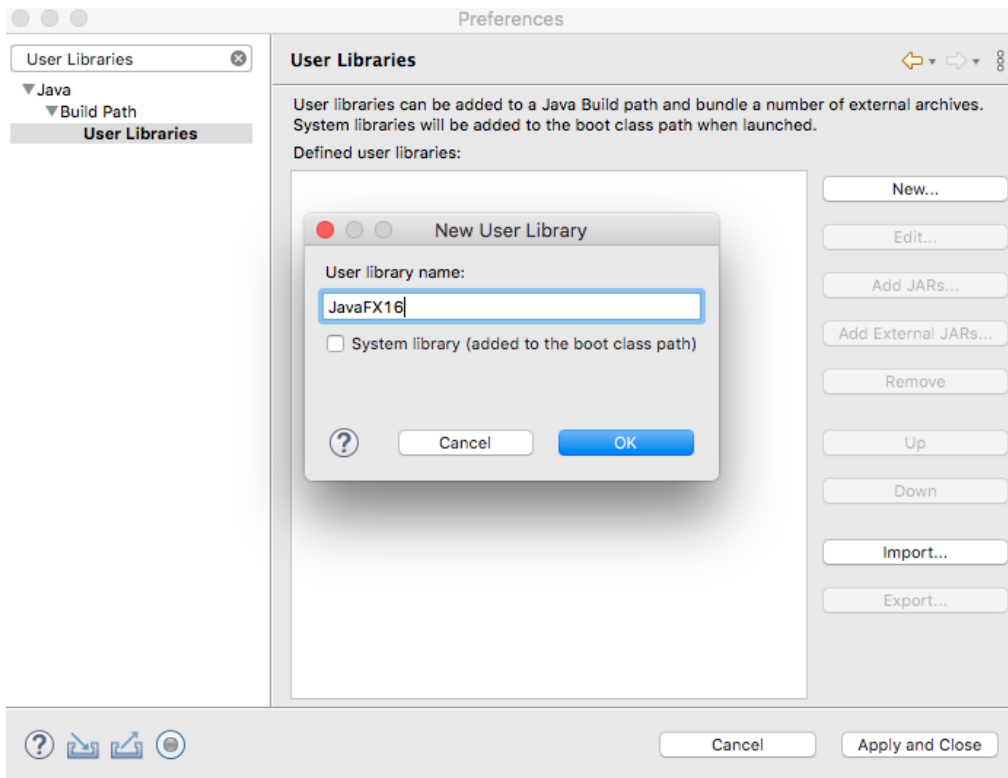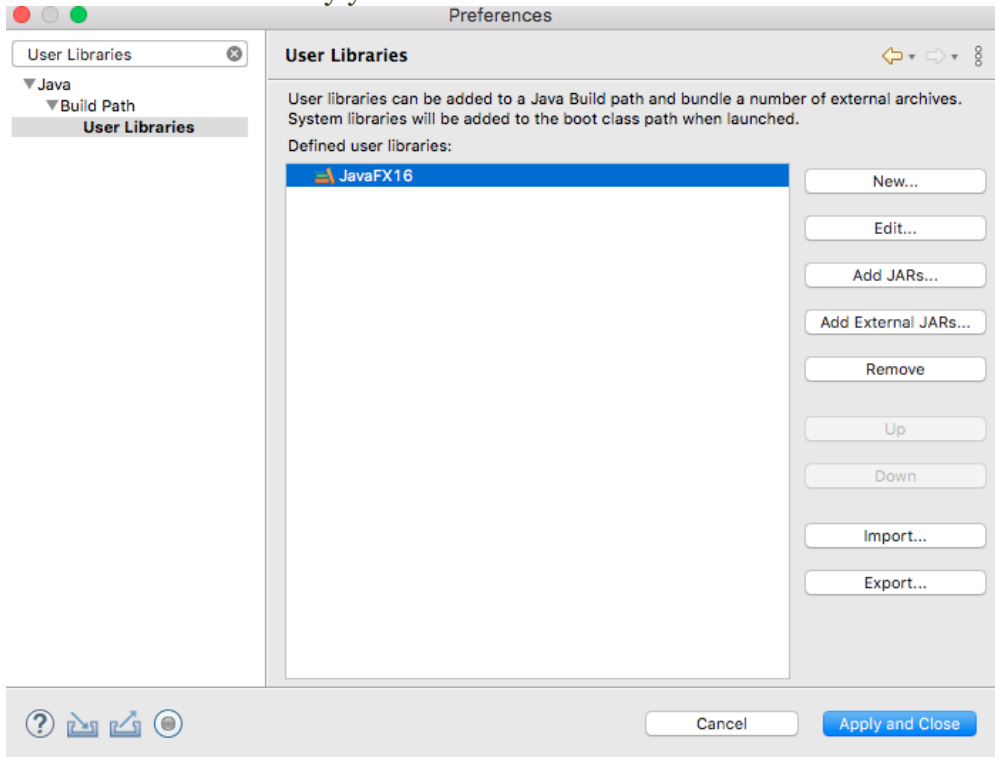
/YOUR_SYSTEM_PATH /javafx-sdk-16/lib
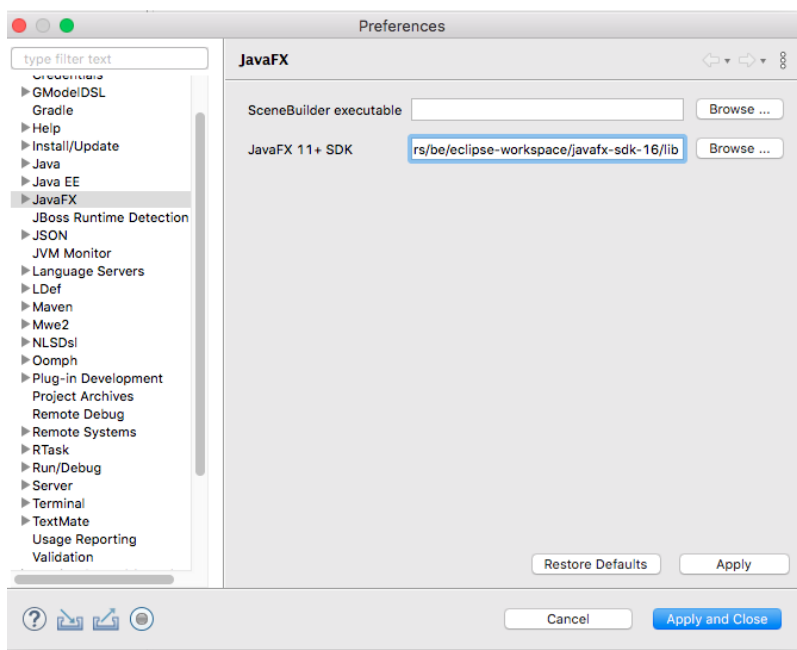
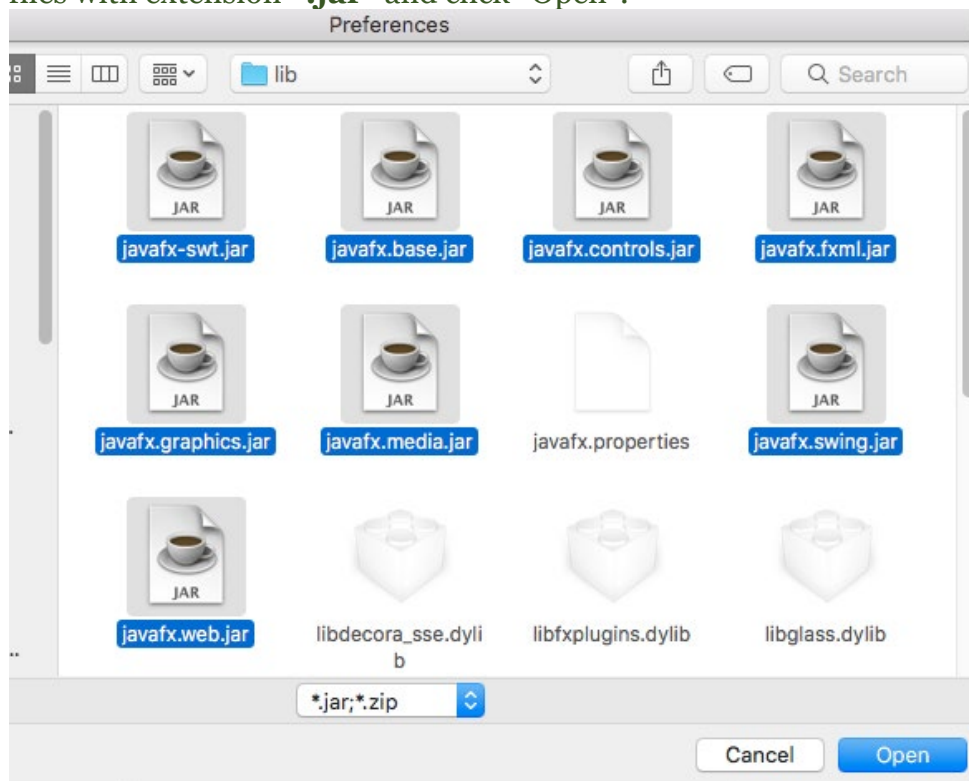6. In Eclipse from Menu Preferences search for "User Libraries".
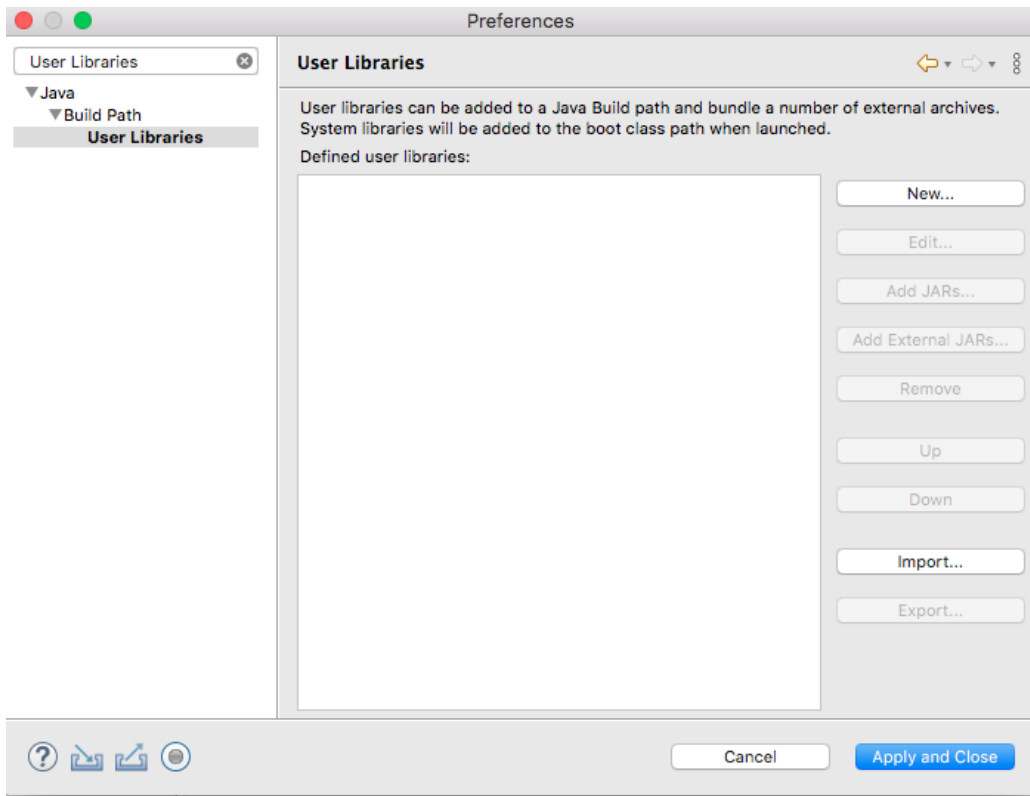
7. Click on "**New...**" button give name to the library and click "**OK**" button.
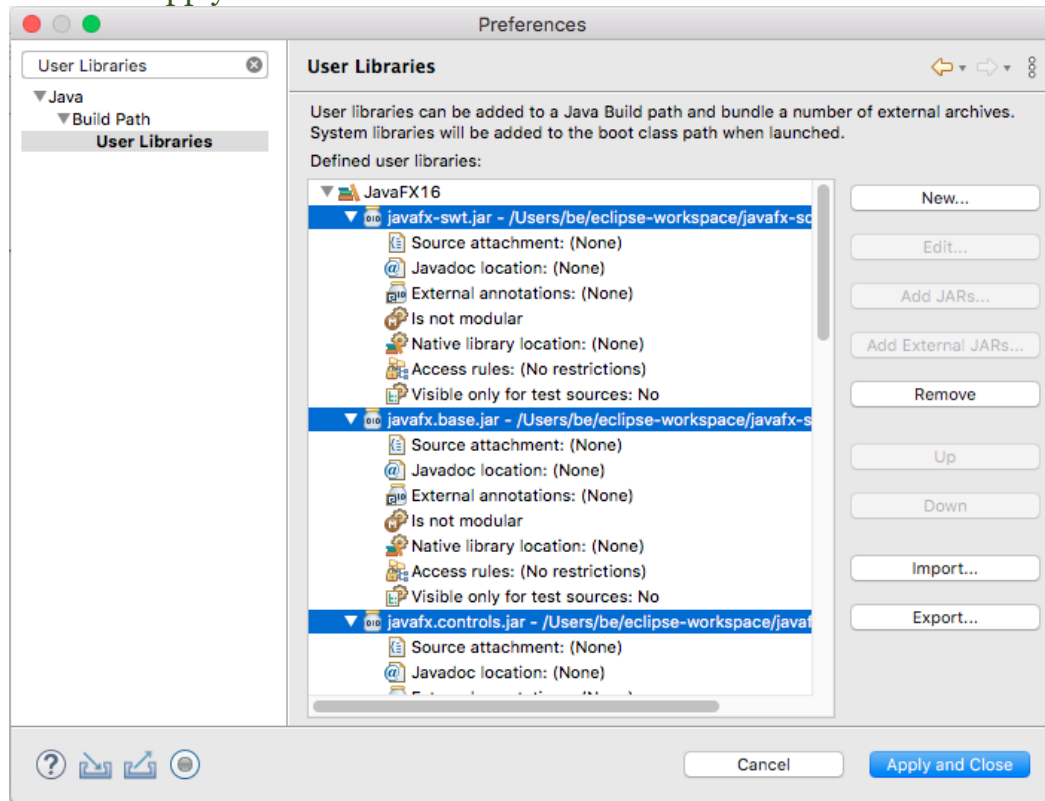
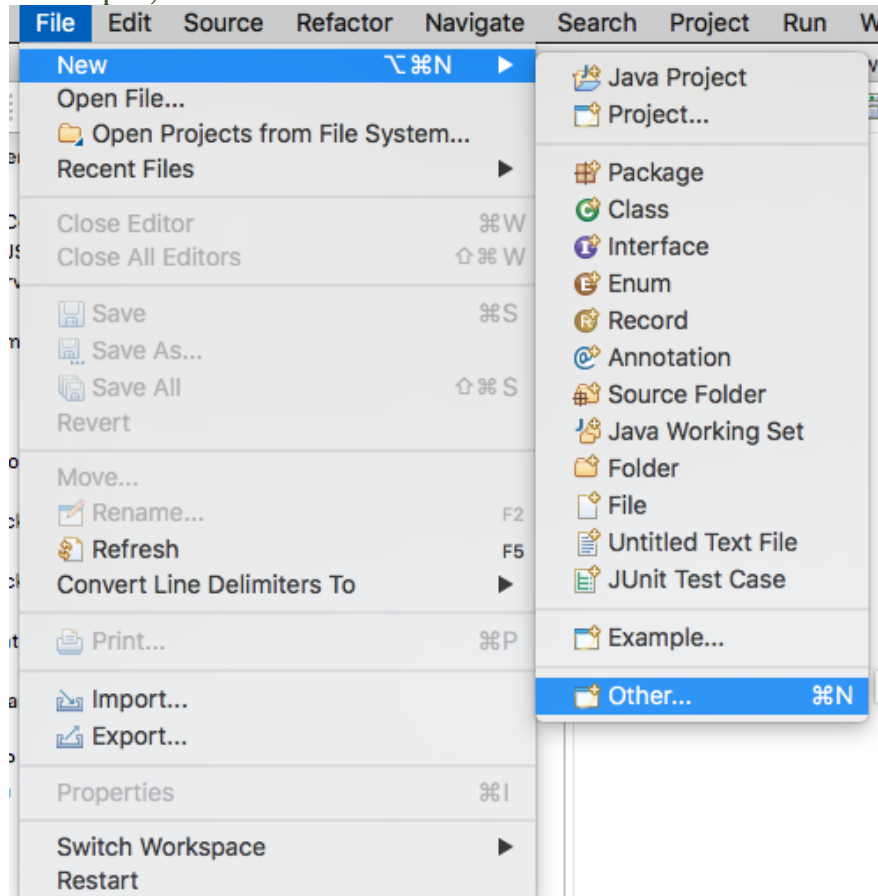8. Select the library you created and click on "Add External JARs" button.

9. Go to the path where you stored the **JavaFX SDK** and go to the **lib** folder and select all the files with extension "**.jar**" and click "Open".
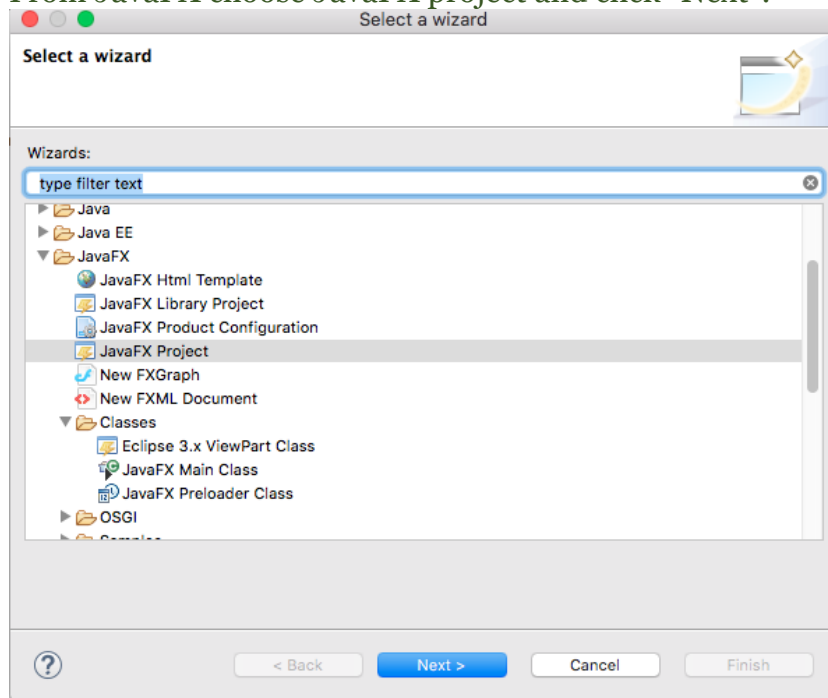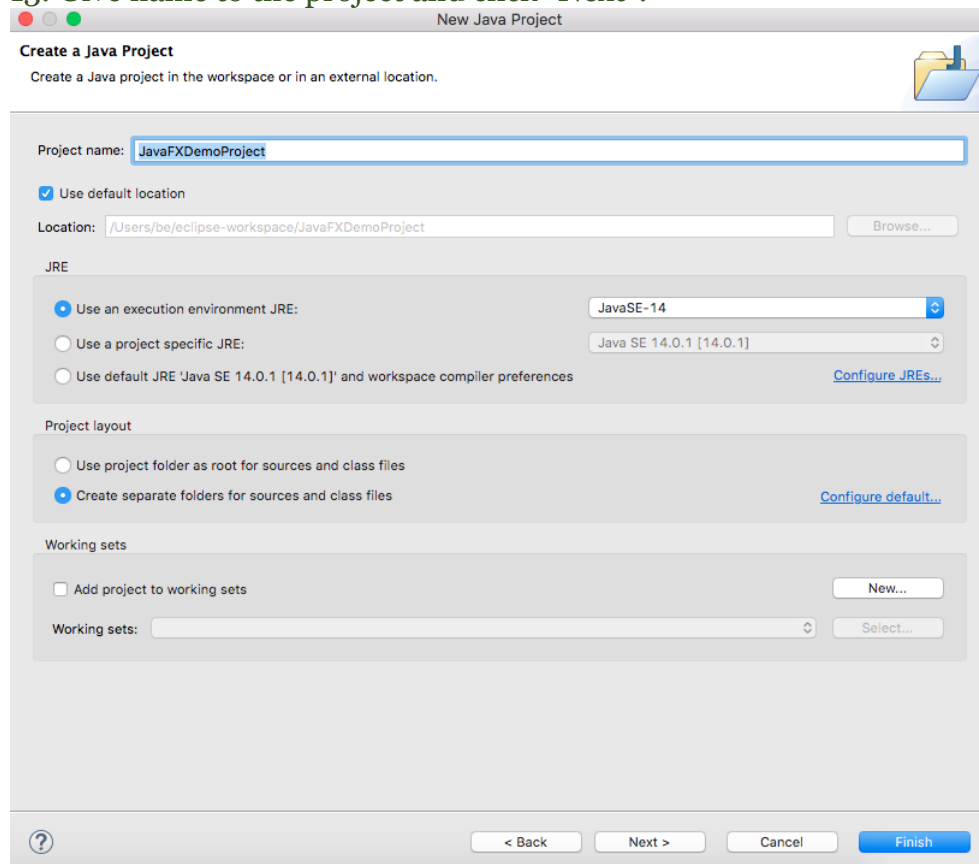
10. Click on "Apply and Close" button.

11. In Eclipse, from the menu File -> New -> Other… . Choose "Other…" menu.

12. From JavaFX choose JavaFX project and click "Next".

13. Give name to the project and click "Next".

14. Uncheck the "Create module-info.java file" checkbox.
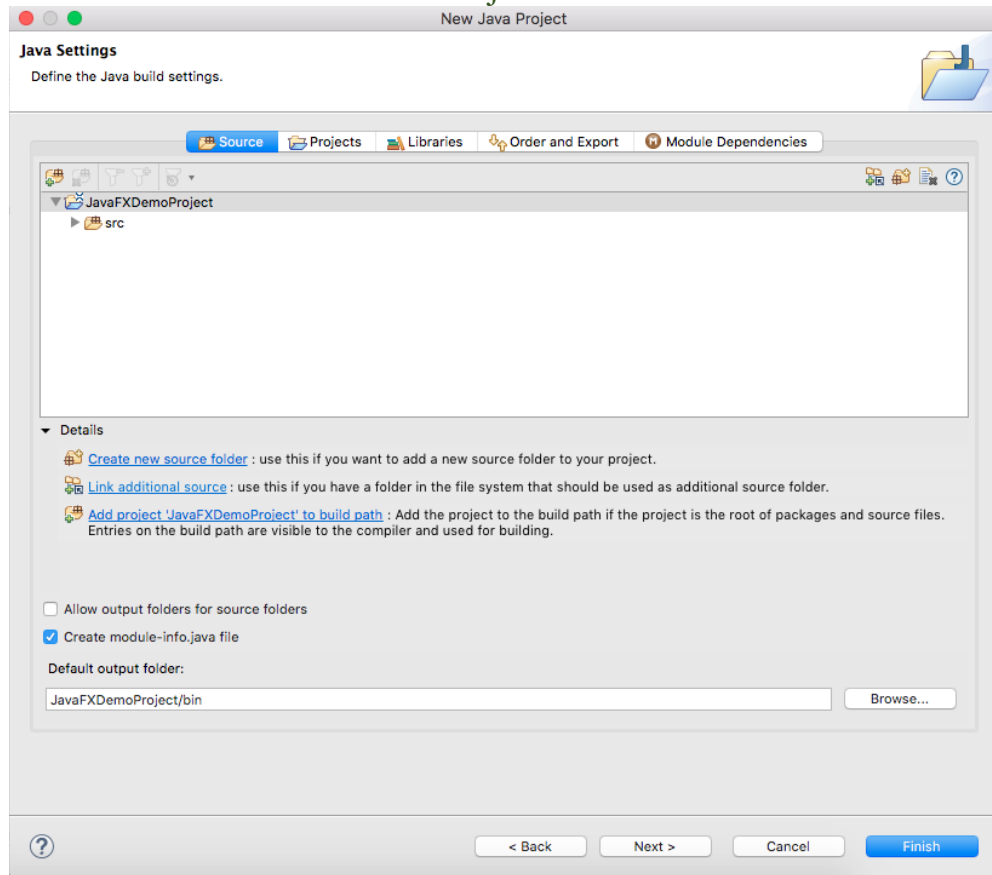
15. Select "Libraries" tab then select "Classpath" and click on "Add Library" button.
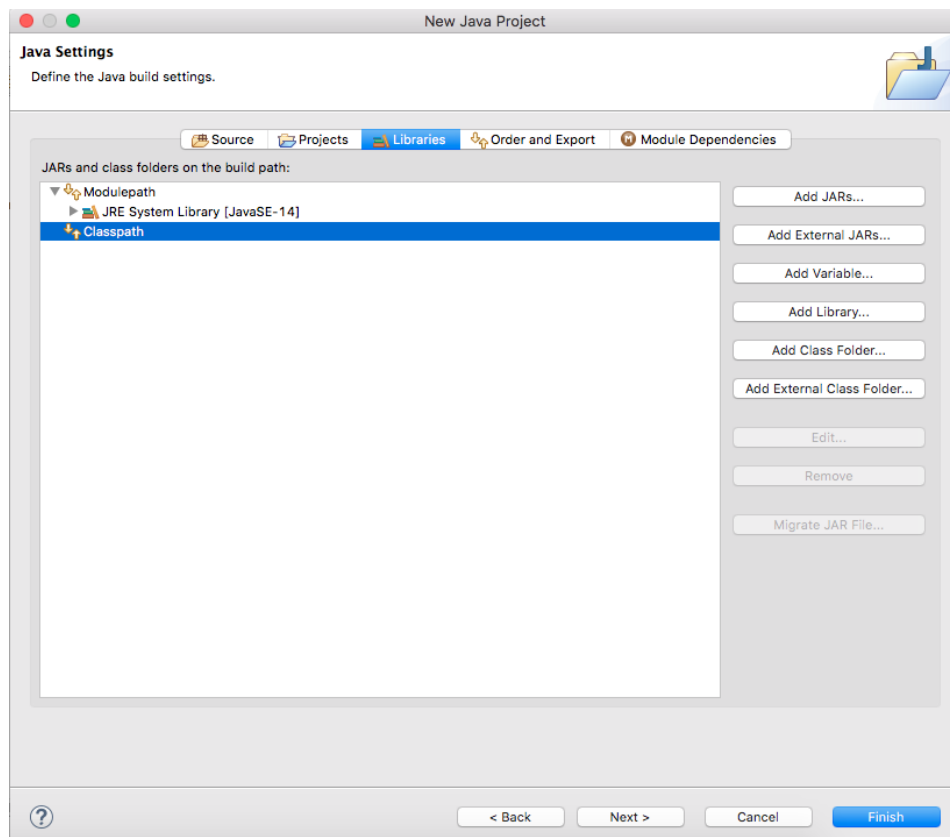
16. Select "User Library" and click "Next" button.
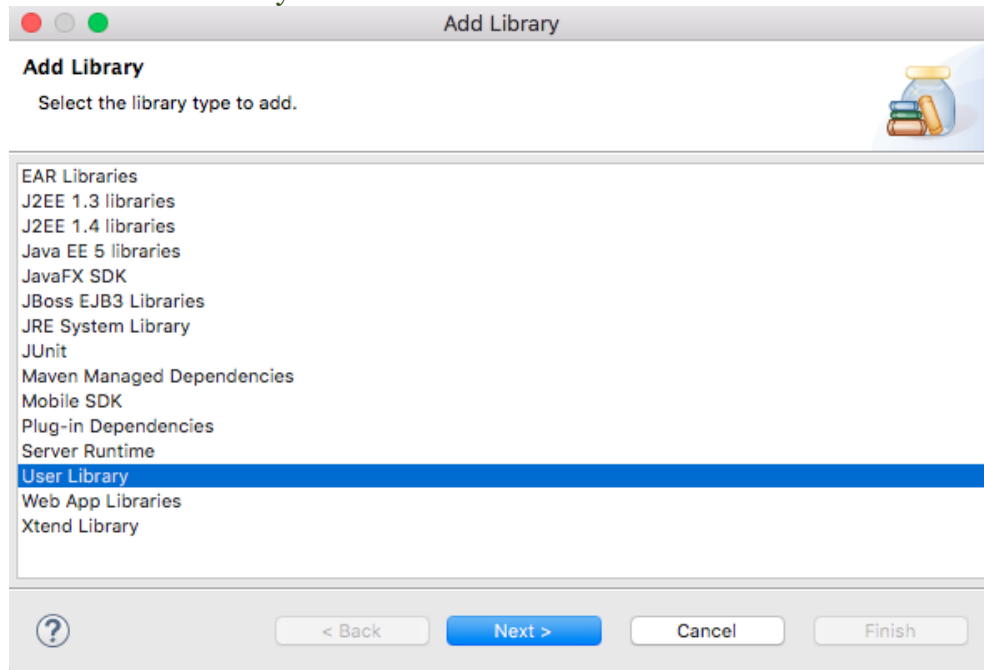
17. Check the library you created and click on "Finish" button.
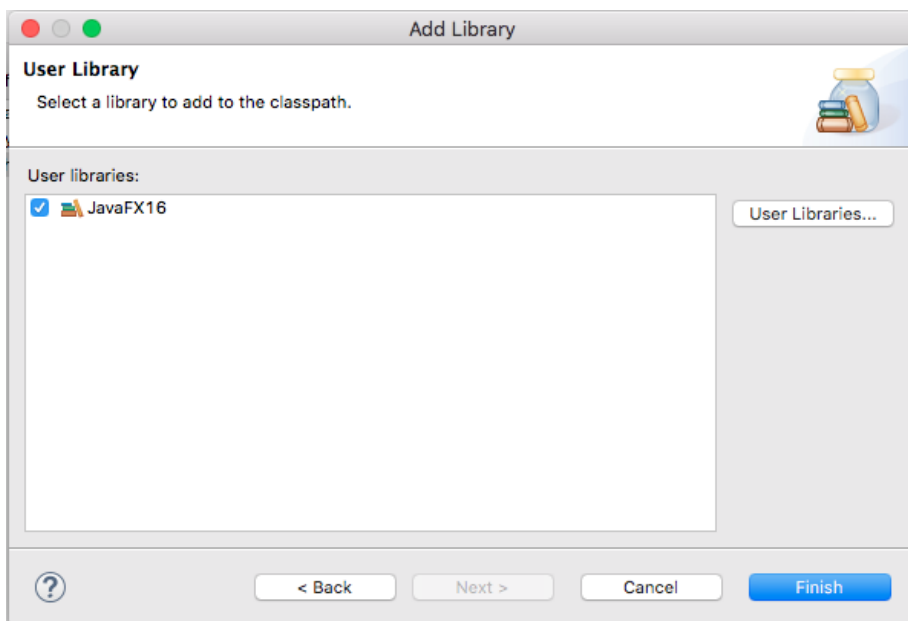
18. Click on "Next" button.
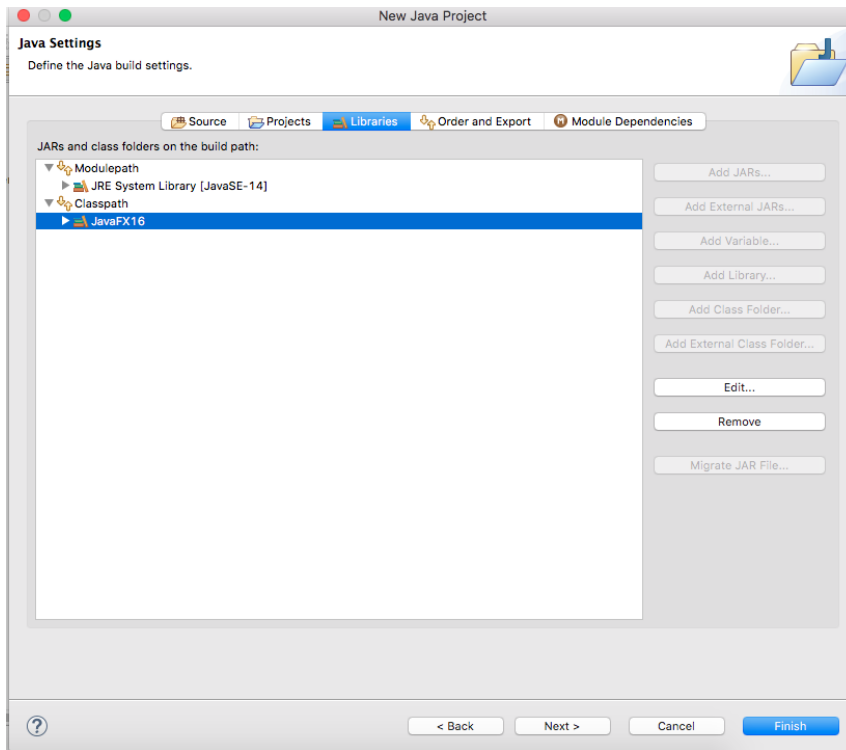
19. Select "FXML" from the language dropdown box.
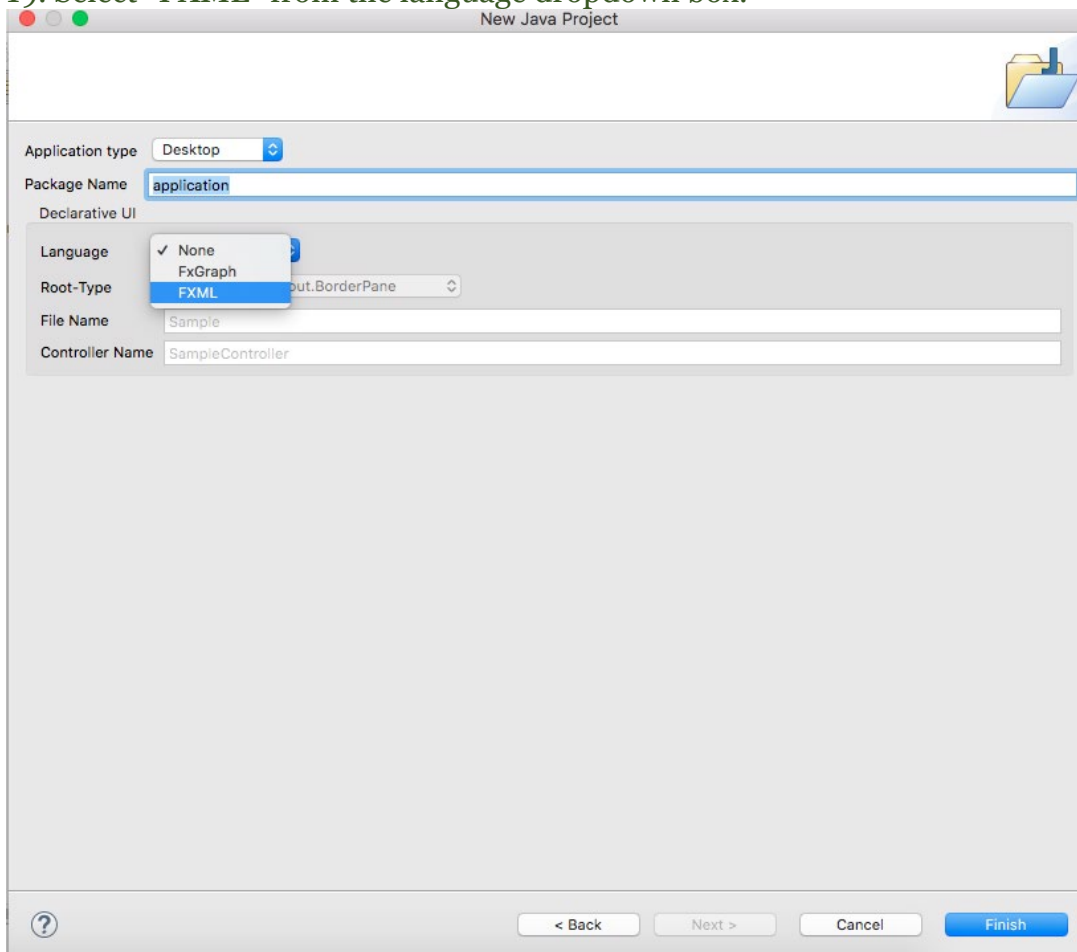
20. You can select root pane type and click on "Finish" button.
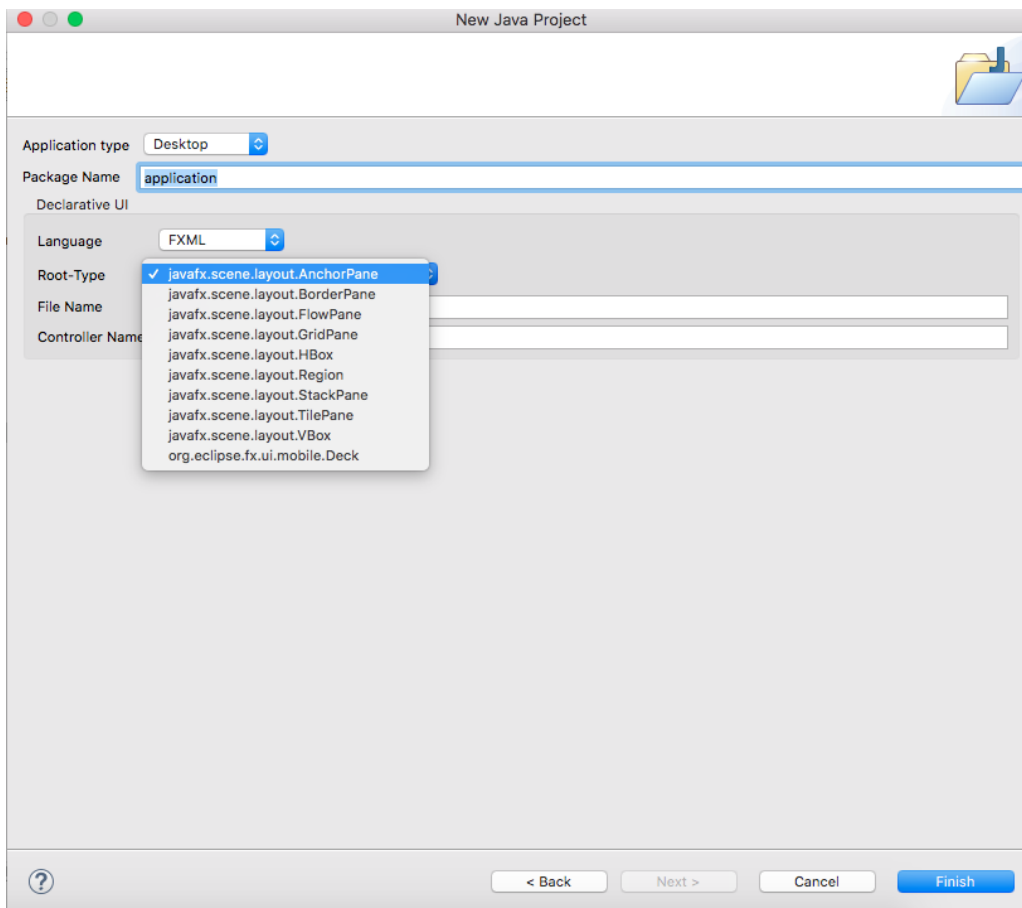
21. Right click on your project and select "Run As" -> "Run Configurations.." menu.
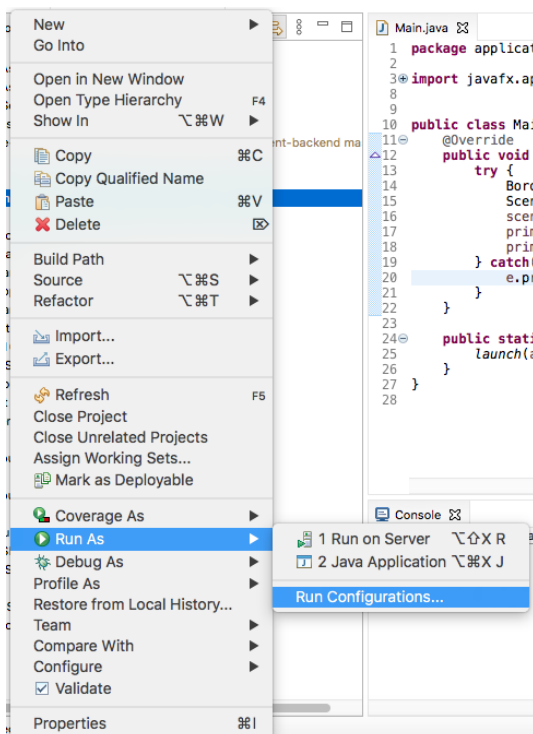
22. In Run Configurations window select "Arguments" tab and uncheck the "*Use the -XstartOnFirstThread argument when launching with SWT*" checkbox.
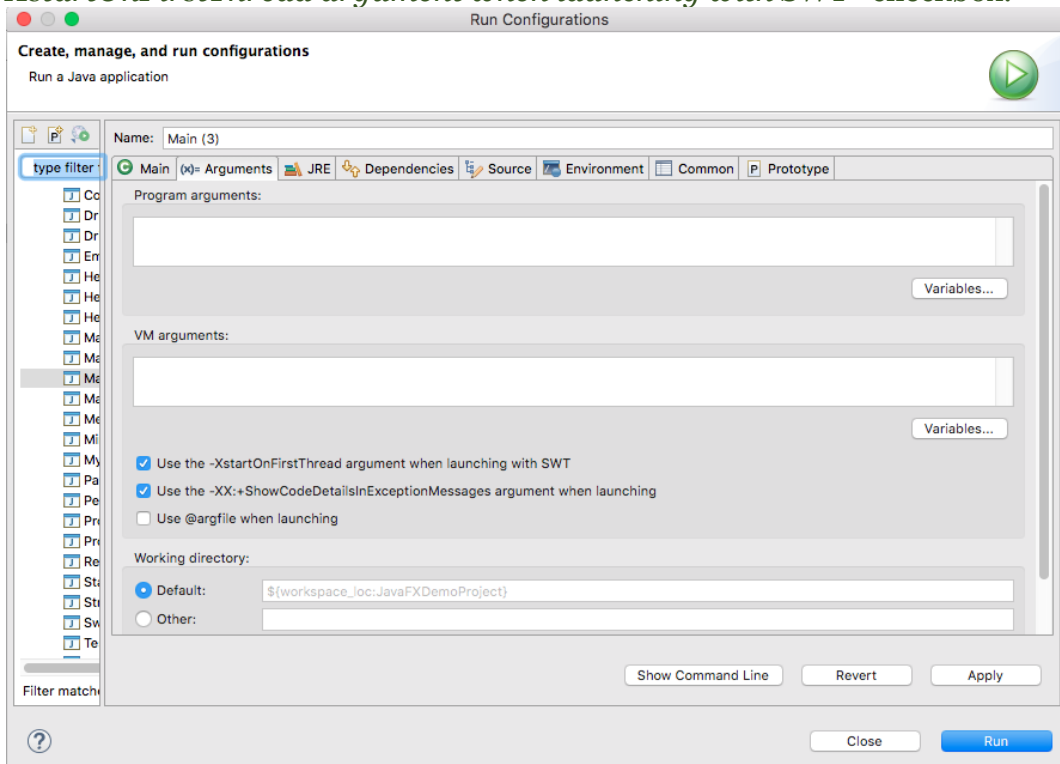
23. Paste the following line in the VM arguments text area and replace the YOUR_SYSTEM_PATH with the path where you stored the JavaFX SDK then click "Run".

--module-path /YOUR_SYSTEM_PATH /javafx-sdk-16/lib --add-modules javafx.controls,javafx.fxml
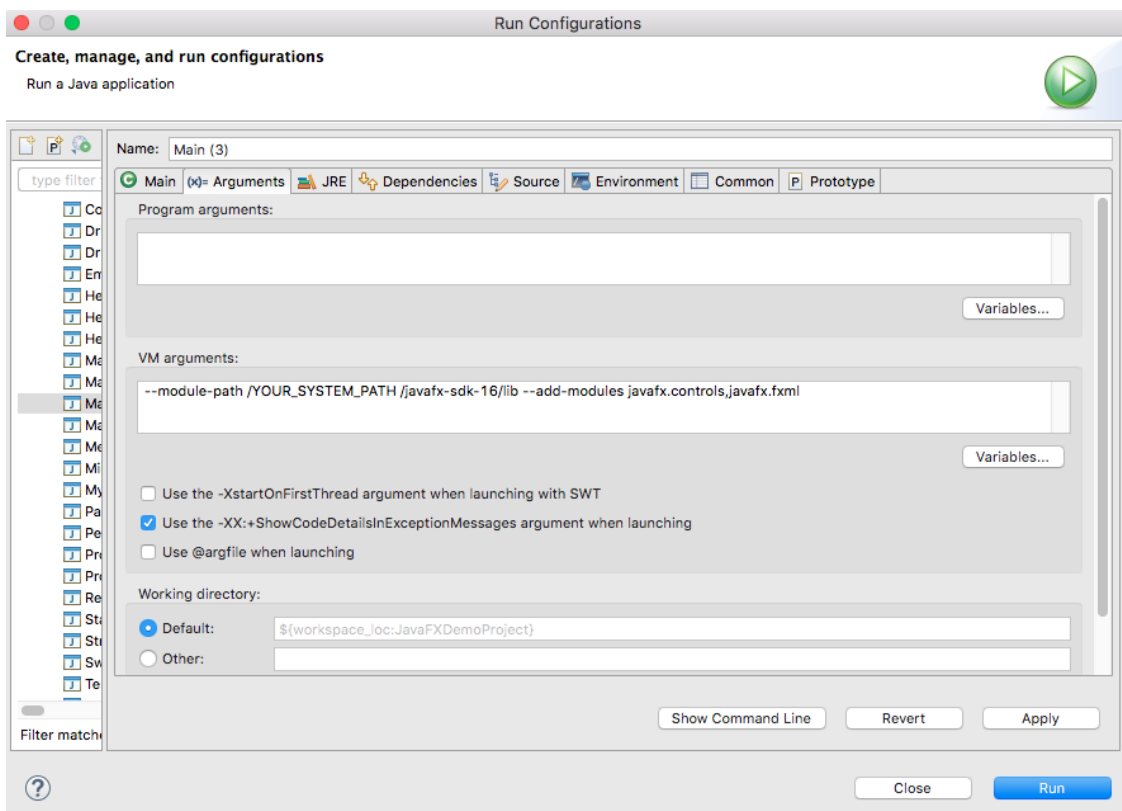
23B. In Dependencies tab you should see the JavaFX SDK you added. If not click on "Advanced" button.
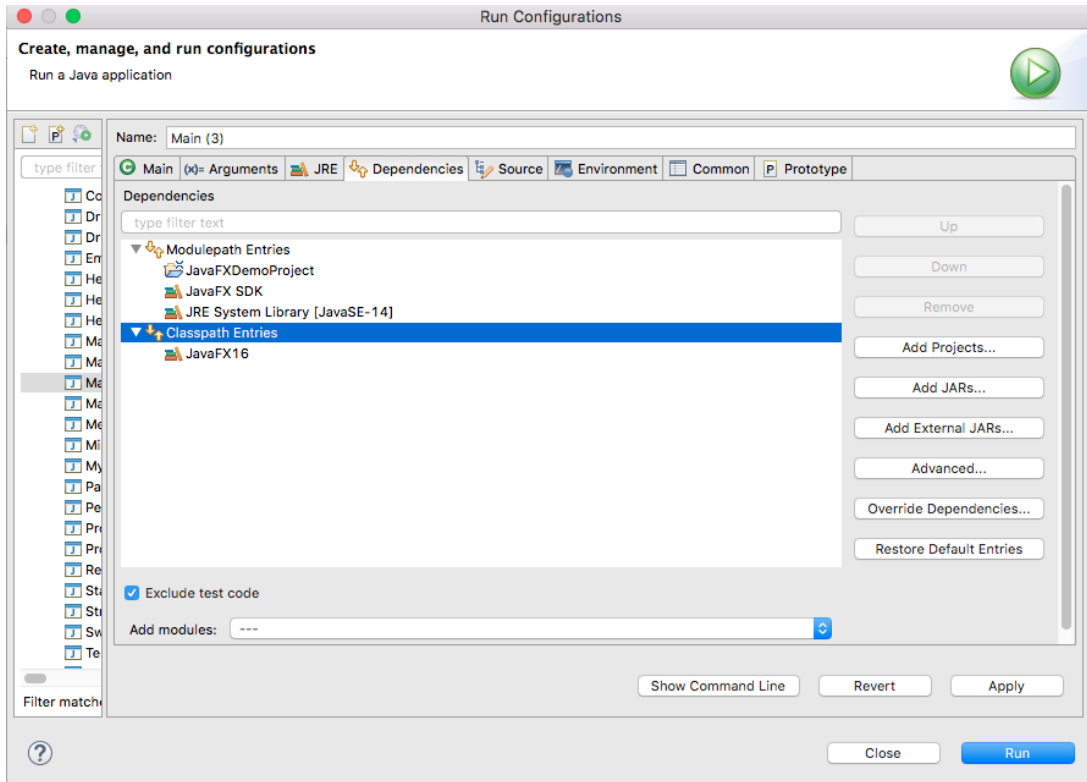
23B1. Check "Add Library" then click "OK" button.
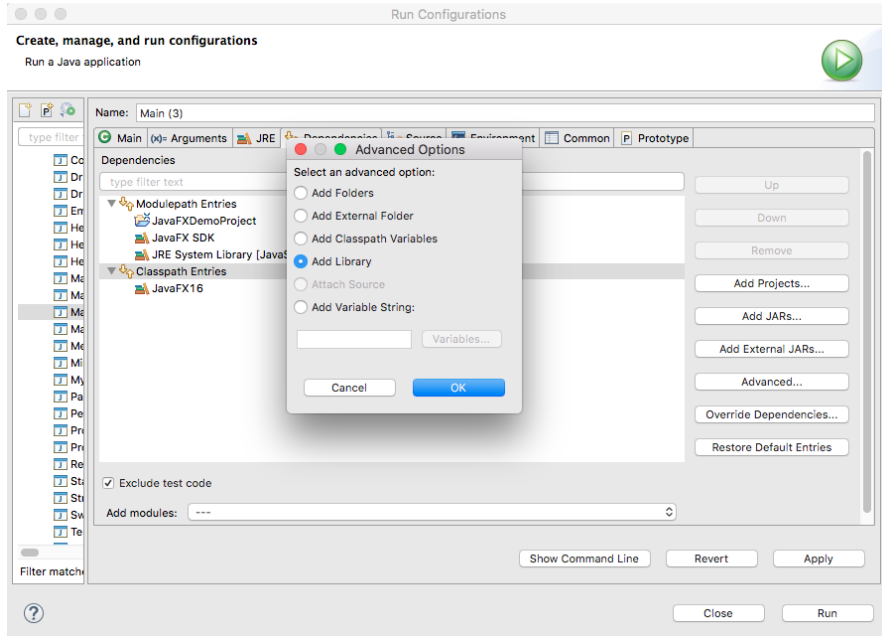
23B2. Choose "User Library" then click "Next".
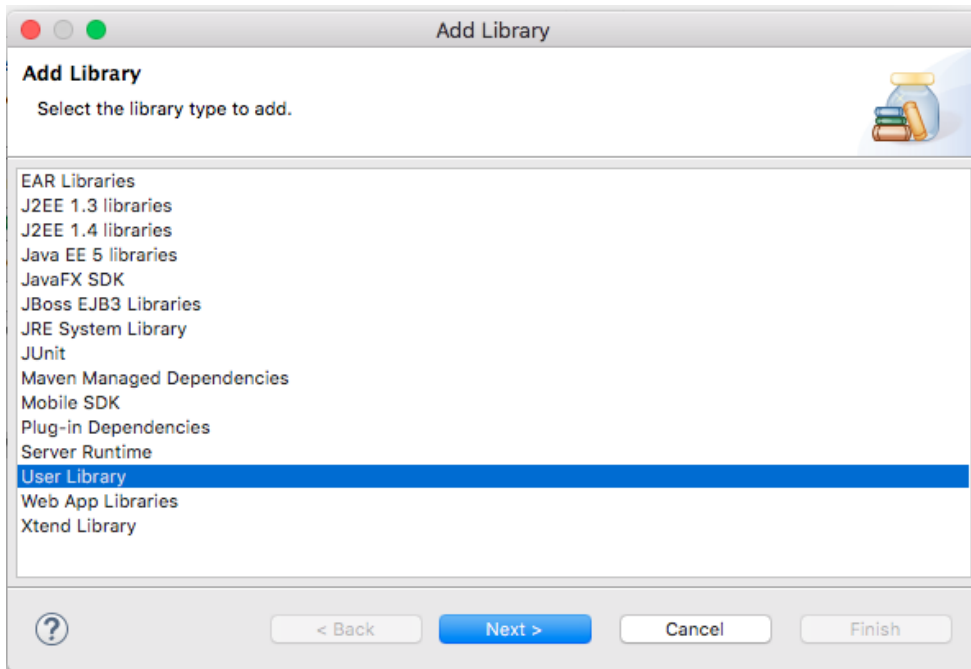
23B3. Check the JavaFX SDK library you added and click "Finish".

24. Right click on your Main.java class and select Run As -> Java Application.

```java
package application;

import javafx.application.Application;

public class Main extends Application {
    @Override
    public void start(Stage primaryStage) {
        try {
            BorderPane root = (BorderPane)FXMLLoader.load(getClass
            Scene scene = new Scene(root,400,400);
            scene.getStylesheets().add(getClass().getResource("app
            primaryStage.setScene(scene);
            primaryStage.show();
        } catch(Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

Menu items:
- Save ⌘S
- Open Declaration F3
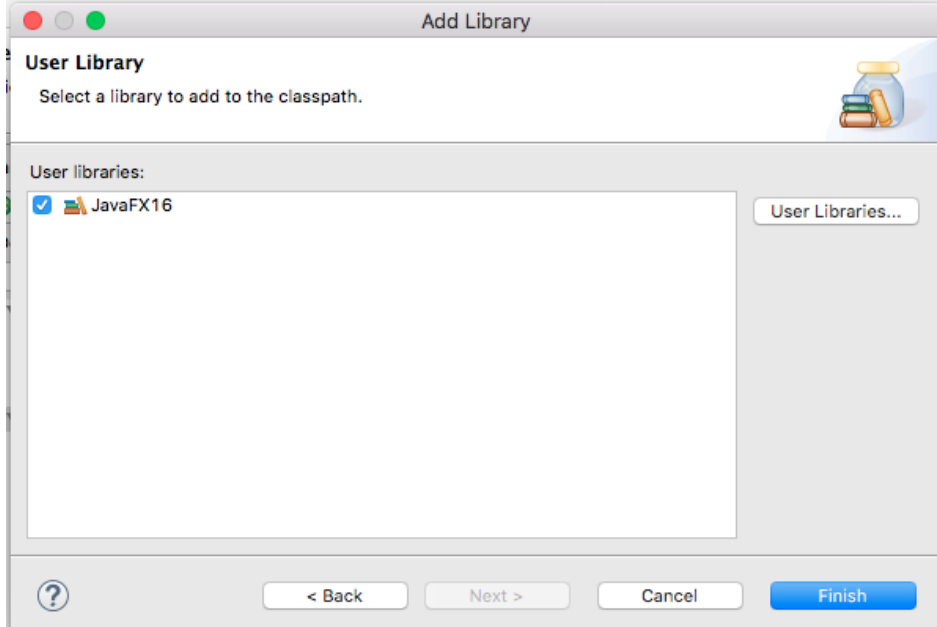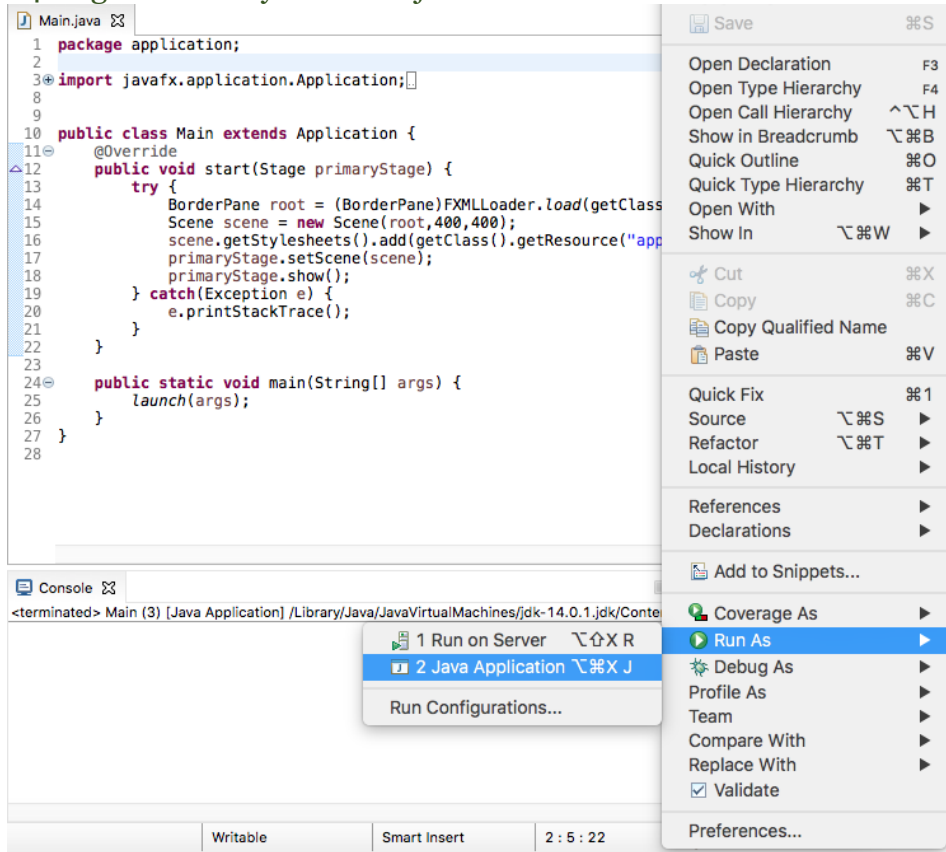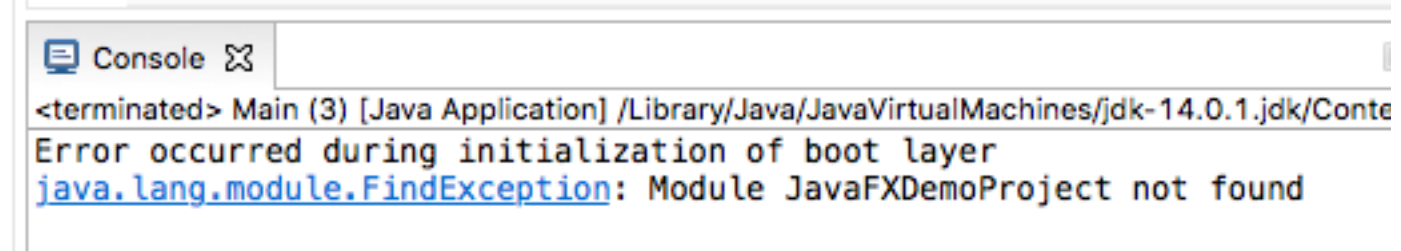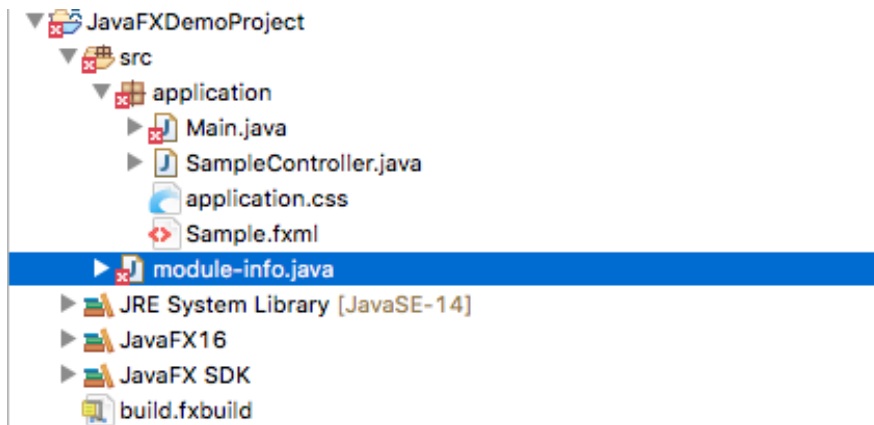- Open Type Hierarchy F4
- Open Call Hierarchy ^⌥H
- Show in Breadcrumb ⌥⌘B
- Quick Outline ⌘O
- Quick Type Hierarchy ⌘T
- Open With ▶
- Show In ⌥⌘W ▶
- Cut ⌘X
- Copy ⌘C
- Copy Qualified Name
- Paste ⌘V
- Quick Fix ⌘1
- Source ⌥⌘S ▶
- Refactor ⌥⌘T ▶
- Local History ▶
- References ▶
- Declarations ▶
- Add to Snippets...
- Coverage As ▶
- Run As ▶
- Debug As ▶
- Profile As ▶
- Team ▶
- Compare With ▶
- Replace With ▶
- ☑ Validate
- Preferences...

Console:
`<terminated> Main (3) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Conte`

- 1 Run on Server ⌥⇧X R
- 2 Java Application ⌥⌘X J
- Run Configurations...

Writable | Smart Insert | 2 : 5 : 22

24B. If you see the following error.

Console

`<terminated> Main (3) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Conte`
`Error occurred during initialization of boot layer`
`java.lang.module.FindException: Module JavaFXDemoProject not found`

24B1. Delete the module-info.java class.

- JavaFXDemoProject
  - src
    - application
      - Main.java
      - SampleController.java
      - application.css
      - Sample.fxml
    - **module-info.java**
  - JRE System Library [JavaSE-14]
  - JavaFX16
  - JavaFX SDK
  - build.fxbuild

25. Try restarting your Eclipse after these steps if you're not seeing window. If you're getting the following error still after restarting then re-do the steps from the beginning.

```
Error occurred during initialization of boot layer
java.lang.module.FindException: Module javafx.controls not found
```
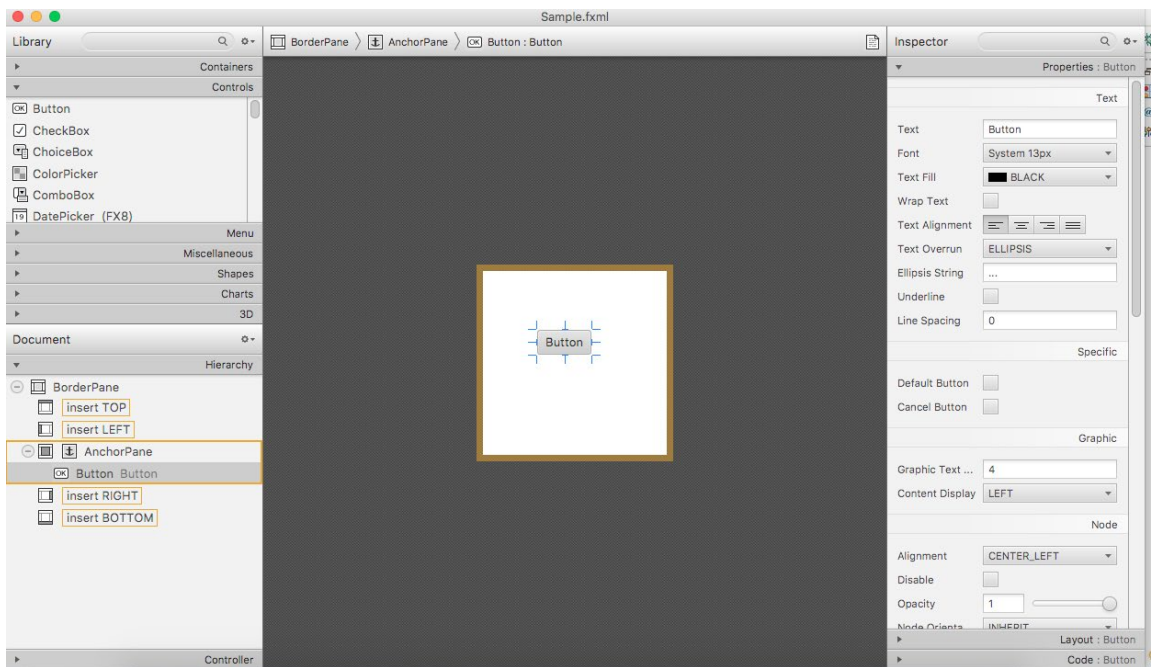
26. After running the project you should see the empty window.

Main (3) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Co
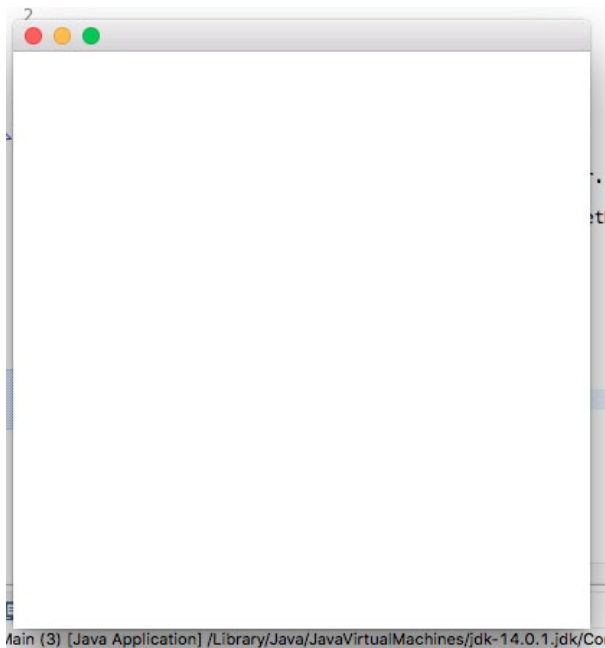
27. Right click on the "Sample.fxml" and choose "Open with Scene Builder".
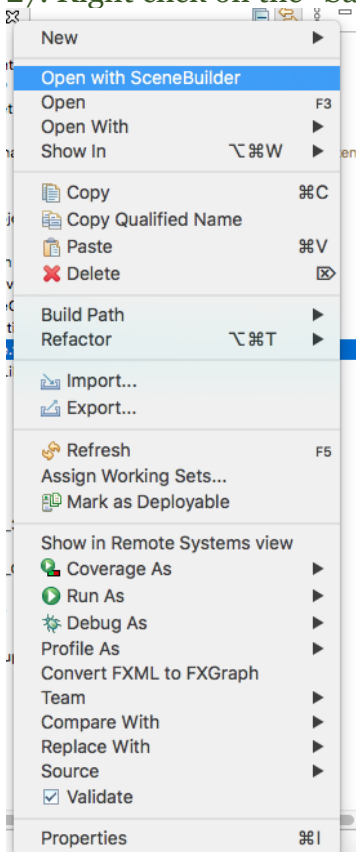
28. Drag and drop some controllers to the window. Don't forget to save the "Sample.fxml" file.

29. Close the current window and run the Main.java class again. You should see the updated window.
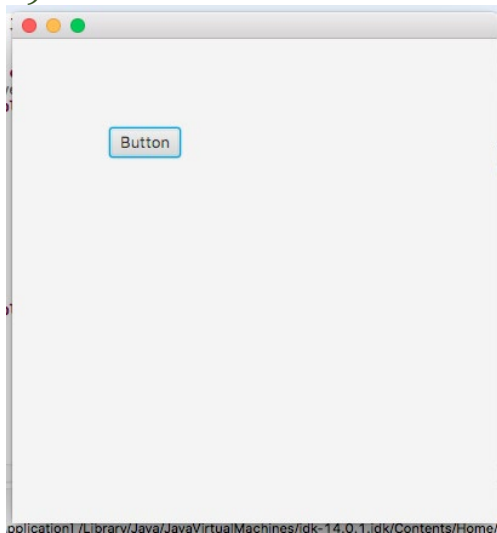
30. Enjoy developing GUI with JavaFX ☺