

# Installing JavaFX in Eclipse

This tutorial describes how to install and configure JavaFX as a library in Eclipse and set it up to work with a new eclipse project.

Before going through the steps in this tutorial, make sure you have Java JDK 11 and Eclipse 2021-03 installed and working for Java programs that do not use JavaFX.

## Download JavaFX

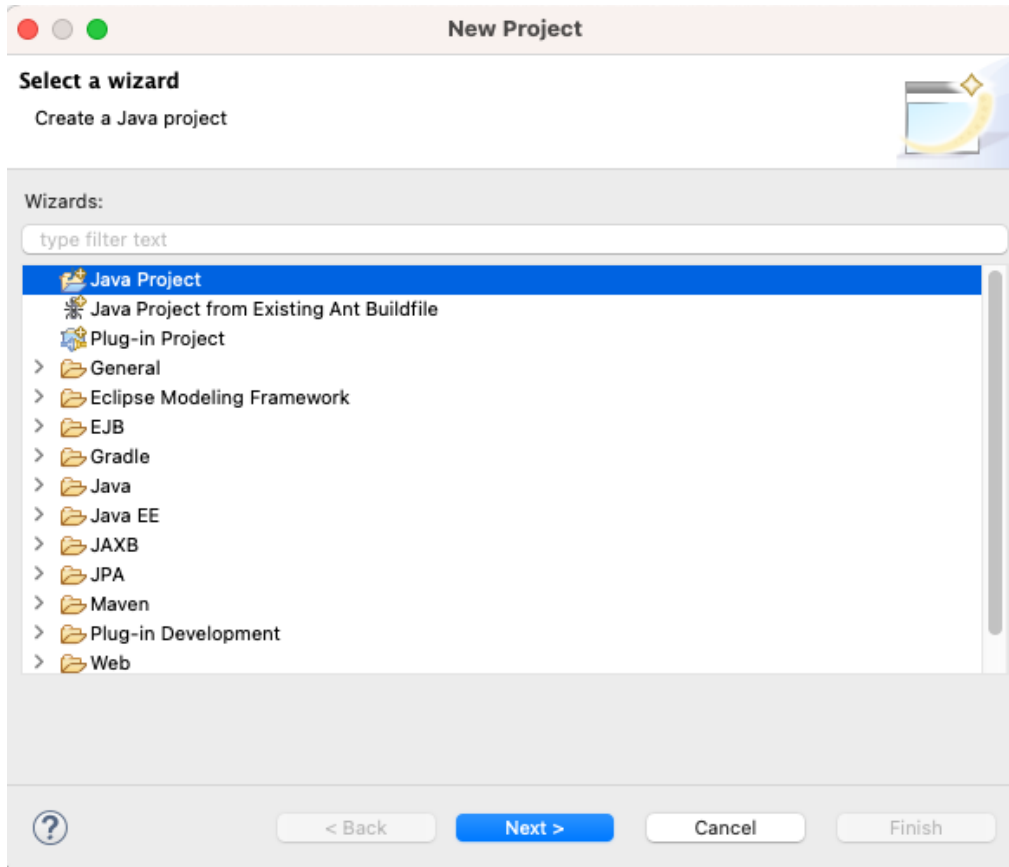
1. Go to <https://gluonhq.com/products/javafx/> [\(https://gluonhq.com/products/javafx/\)](https://gluonhq.com/products/javafx/) and download the SDK version for your platform (you'll only need the SDK, not the jmods).

Product	Public version	LTS version	Platform
JavaFX Windows SDK	11.0.2	11.0.10 <a href="#">More info</a>	Windows
JavaFX Windows jmods	11.0.2	11.0.10 <a href="#">More info</a>	Windows
JavaFX Mac OS X SDK	11.0.2	11.0.10 <a href="#">More info</a>	Mac
JavaFX Mac OS X jmods	11.0.2	11.0.10 <a href="#">More info</a>	Mac
JavaFX Linux SDK	11.0.2	11.0.10 <a href="#">More info</a>	Linux
JavaFX Linux jmods	11.0.2	11.0.10 <a href="#">More info</a>	Linux
JavaFX armv6hf SDK	11.0.2	11.0.10 <a href="#">More info</a>	Embedded armv6hf
JavaFX Documentation	11.0.2	11.0.10 <a href="#">More info</a>	Javadoc

2. Extract the zip archive to a folder in your home directory. On my OSX machine, I extracted it into **/Users/florian/Applications/javafx-sdk-11.0.2/**. You will need this path later to set up JavaFX in Eclipse.

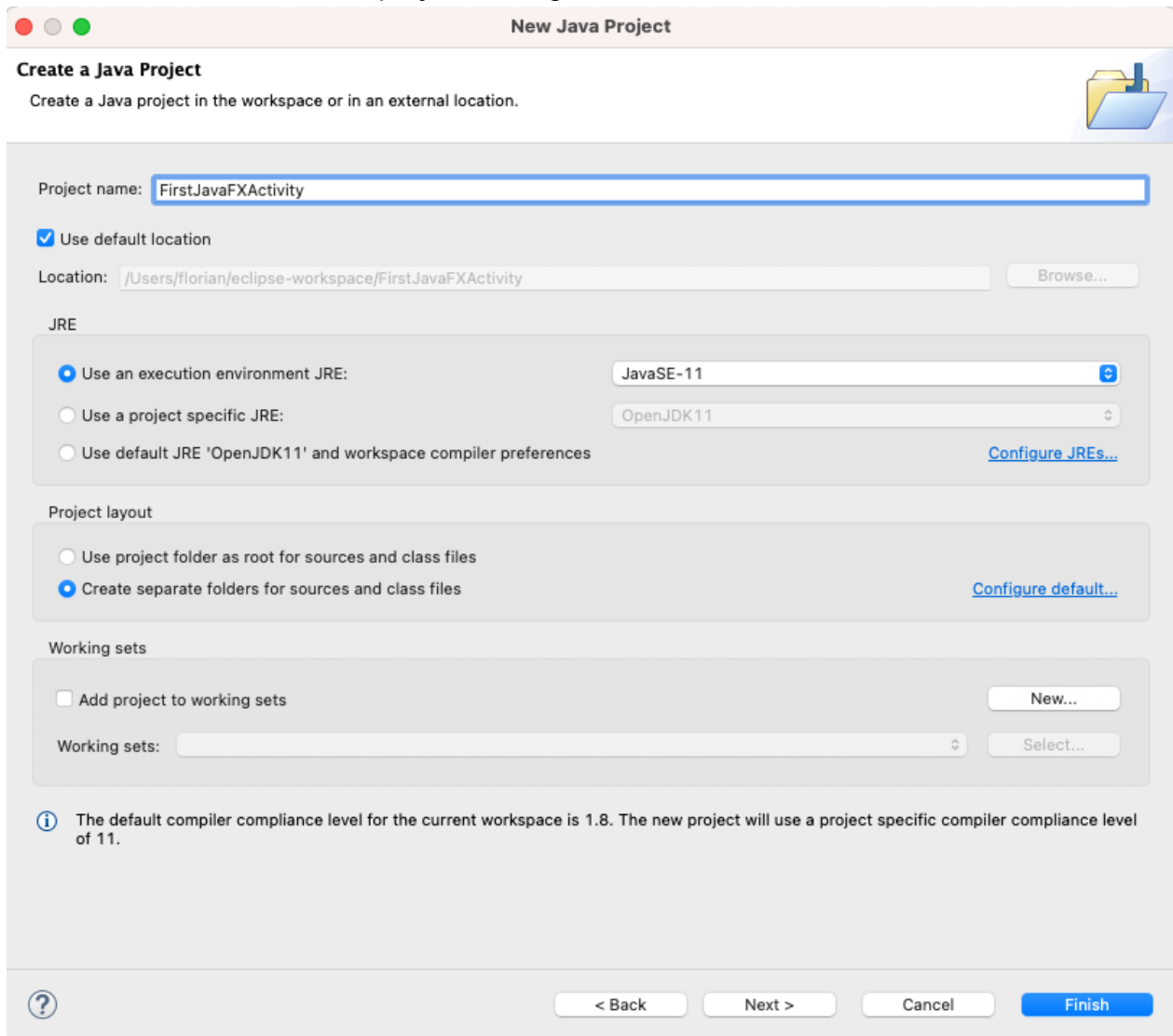
## Create a New Java Project

3. Navigate to **File -> New -> Project**. Then select **Java Project** and click on **Next**.



4. Provide a name for the project, such as **FirstJavaFXActivity**.

5. Click **Next** to edit additional project settings.



The screenshot shows the 'New Java Project' dialog box in the Eclipse IDE. The title bar reads 'New Java Project'. The main heading is 'Create a Java Project', followed by the instruction 'Create a Java project in the workspace or in an external location.' and a folder icon. The 'Project name' field contains 'FirstJavaFXActivity'. The 'Use default location' checkbox is checked, and the 'Location' field shows the path '/Users/florian/eclipse-workspace/FirstJavaFXActivity' with a 'Browse...' button. The 'JRE' section has three radio buttons: 'Use an execution environment JRE:' (selected), 'Use a project specific JRE:', and 'Use default JRE 'OpenJDK11' and workspace compiler preferences'. The selected option has a dropdown menu showing 'JavaSE-11'. The 'Project layout' section has two radio buttons: 'Use project folder as root for sources and class files' and 'Create separate folders for sources and class files' (selected). The 'Working sets' section has a checkbox 'Add project to working sets' and a 'New...' button. Below it, the 'Working sets:' field is empty, and there is a 'Select...' button. A note at the bottom states: 'The default compiler compliance level for the current workspace is 1.8. The new project will use a project specific compiler compliance level of 11.' The bottom of the dialog features a help icon, '< Back' button, 'Next >' button, 'Cancel' button, and 'Finish' button.

**New Java Project**

**Create a Java Project**  
Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location:  [Browse...](#)

**JRE**

☒ Use an execution environment JRE:  [Configure JREs...](#)

☐ Use a project specific JRE:

☐ Use default JRE 'OpenJDK11' and workspace compiler preferences

**Project layout**

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

**Working sets**

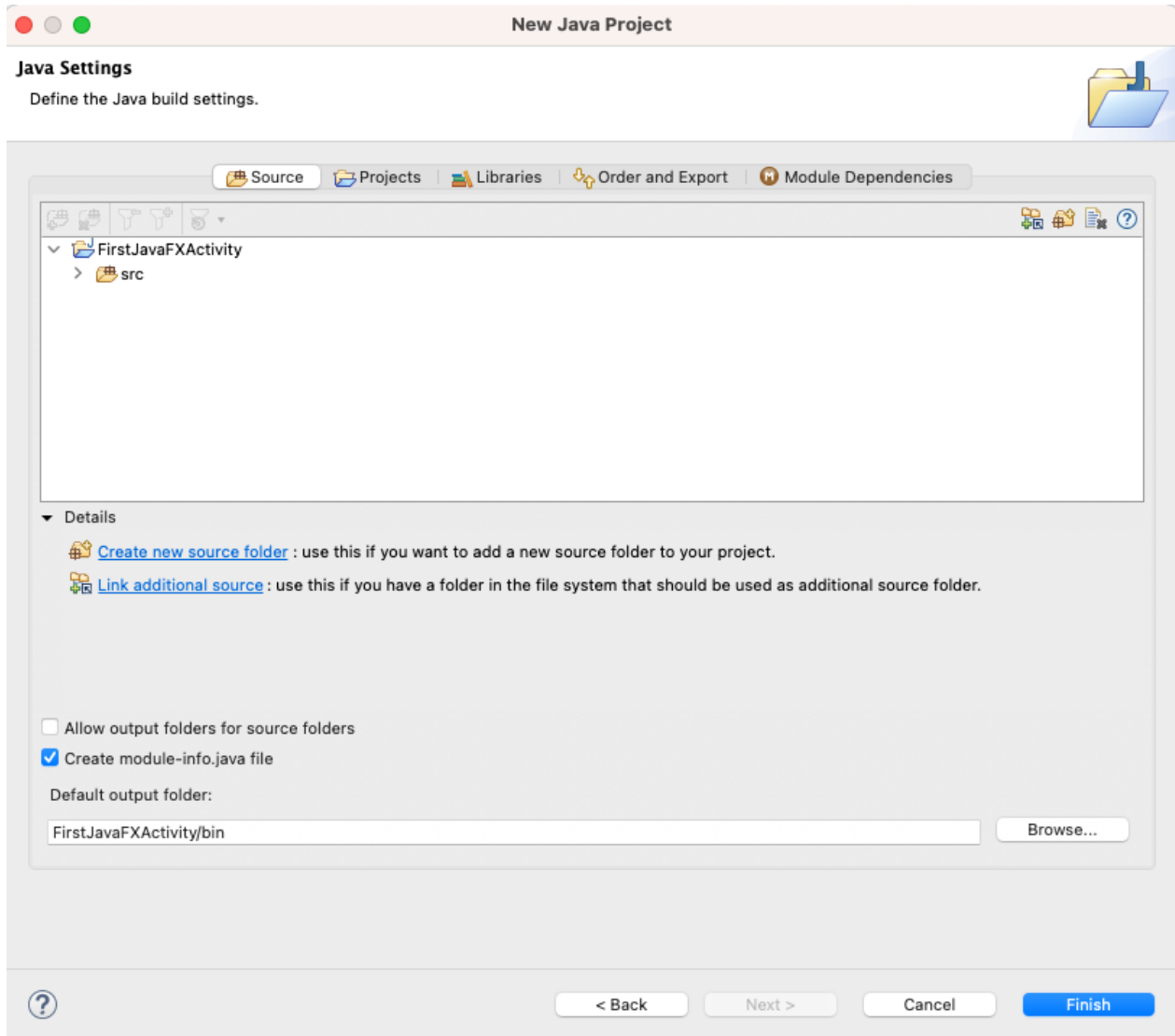
☐ Add project to working sets [New...](#)

Working sets:  [Select...](#)

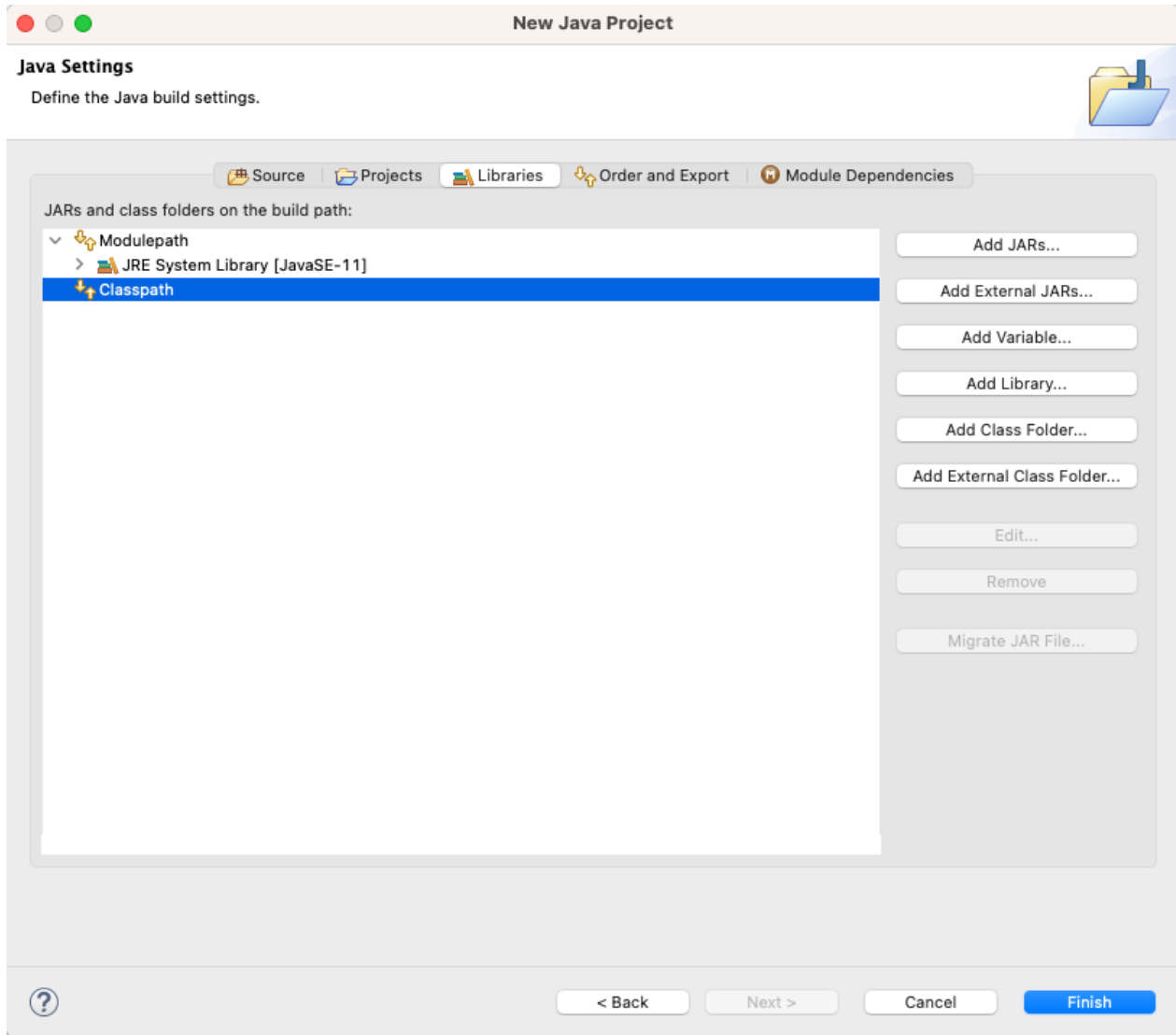
**Information:** The default compiler compliance level for the current workspace is 1.8. The new project will use a project specific compiler compliance level of 11.

[?](#) [< Back](#) [Next >](#) [Cancel](#) [Finish](#)

## 6. Select the **Libraries** tab.

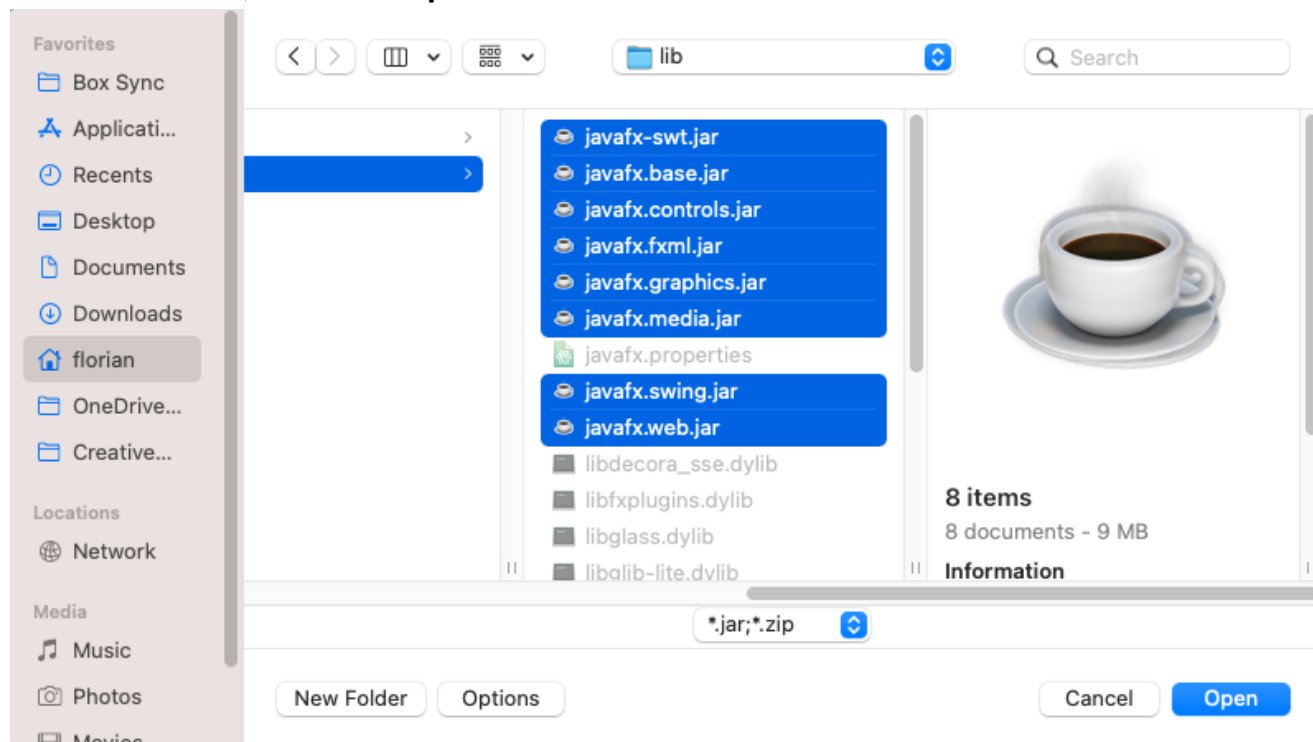


7. Select **Classpath** and click on **Add External JARs...**



8. Go to the folder that you extracted the JavaFX archive into and navigate to the **lib** subdirectory. On my machine, the full path is **/Users/florian/Applications/javafx-sdk-11.0.2/lib/**. Select all jar

files in the folder, then click **Open**.



9. The jar files will now be listed under **Classpath**. Click on **Finish**. You'll be asked to create a module-info.java, which you can skip by clicking **Don't Create**.

## New Java Project

### Java Settings

Define the Java build settings.



Source Projects Libraries Order and Export Module Dependencies

JARs and class folders on the build path:

- ▼ Modulepath
  - > JRE System Library [JavaSE-11]
- ▼ Classpath
  - > javafx.base.jar - /Users/florian/Applications/javafx-sdk-11.0.2/lib
  - > javafx.controls.jar - /Users/florian/Applications/javafx-sdk-11.0.2/lib
  - > javafx.fxml.jar - /Users/florian/Applications/javafx-sdk-11.0.2/lib
  - > javafx.graphics.jar - /Users/florian/Applications/javafx-sdk-11.0.2/lib
  - > javafx.media.jar - /Users/florian/Applications/javafx-sdk-11.0.2/lib
  - > javafx.swing.jar - /Users/florian/Applications/javafx-sdk-11.0.2/lib
  - > javafx.web.jar - /Users/florian/Applications/javafx-sdk-11.0.2/lib
  - > javafx-swt.jar - /Users/florian/Applications/javafx-sdk-11.0.2/lib

Add JARs...

Add External JARs...

Add Variable...

Add Library...

Add Class Folder...

Add External Class Folder...

Edit...

Remove

Migrate JAR File...



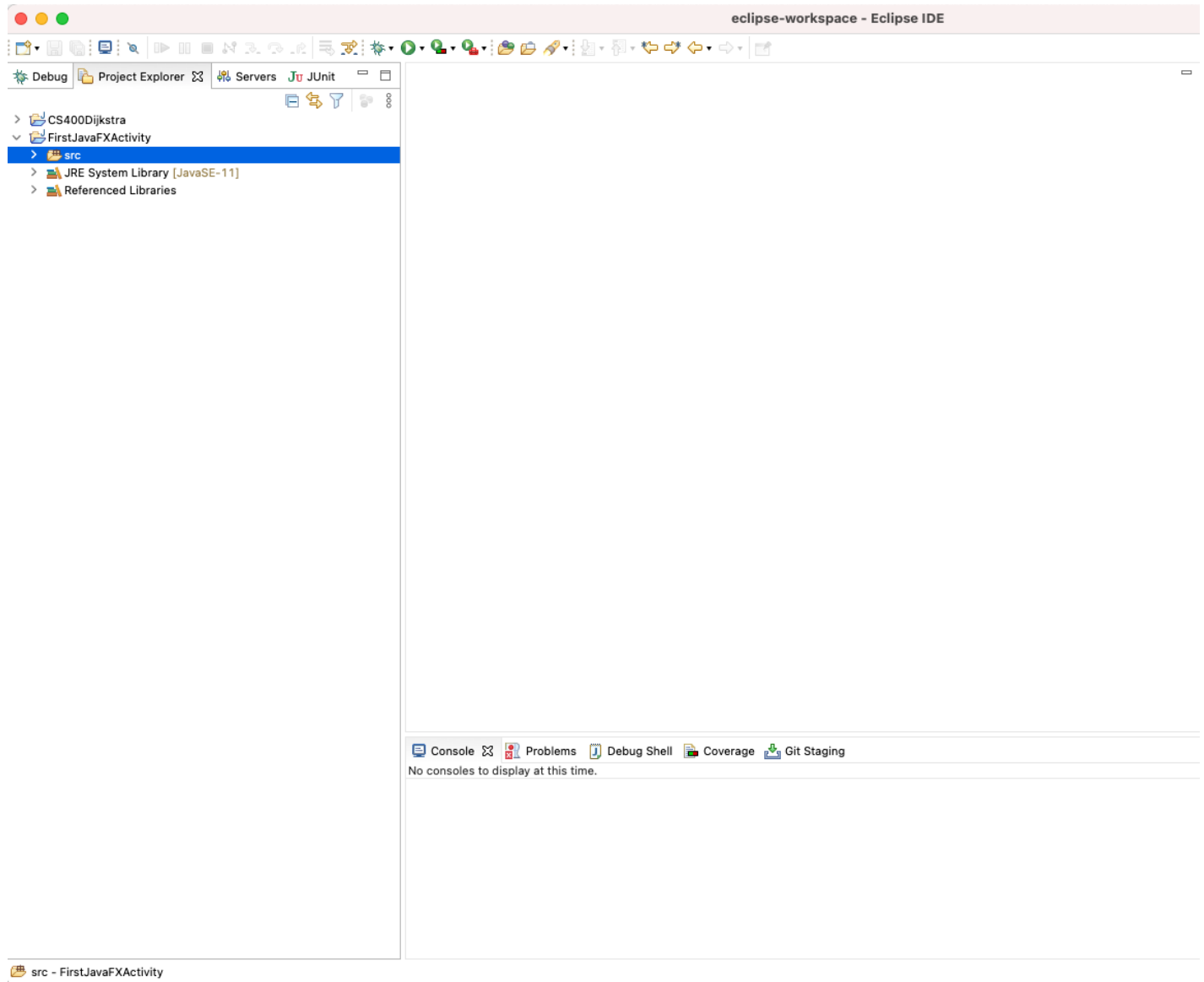
< Back

Next >

Cancel

Finish

10. The project **FirstJavaFXActivity** will now show up in your **Projects** list in Eclipse.



## Creating the Program Starter File

11. Right click on the **src** folder and navigate to **New -> Class** in the context menu. Type in the name **JavaFXActivity** and click on **Finish**.

**New Java Class**

**Java Class**

⚠ The use of the default package is discouraged.

Source folder:

Package:

☐ Enclosing type:

---

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

---

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

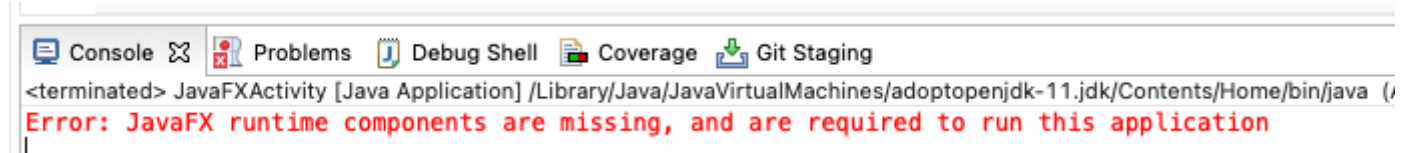
12. Open the file **JavaFXActivity.java** and copy the following starter code for the [First JavaFX Activity](https://canvas.wisc.edu/courses/244796/pages/12-first-javafx-activity) (<https://canvas.wisc.edu/courses/244796/pages/12-first-javafx-activity>) into it:

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.stage.Stage;
// these imports are used for the First JavaFX Activity
import javafx.scene.control.Label;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Polygon;

public class JavaFXActivity extends Application {
    @Override
    public void start(final Stage stage) {
        // update this method definition to complete the First JavaFX Activity
        Group group = new Group();
        Scene scene = new Scene(group, 320, 240);
        stage.setScene(scene);
        stage.show();
    }

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

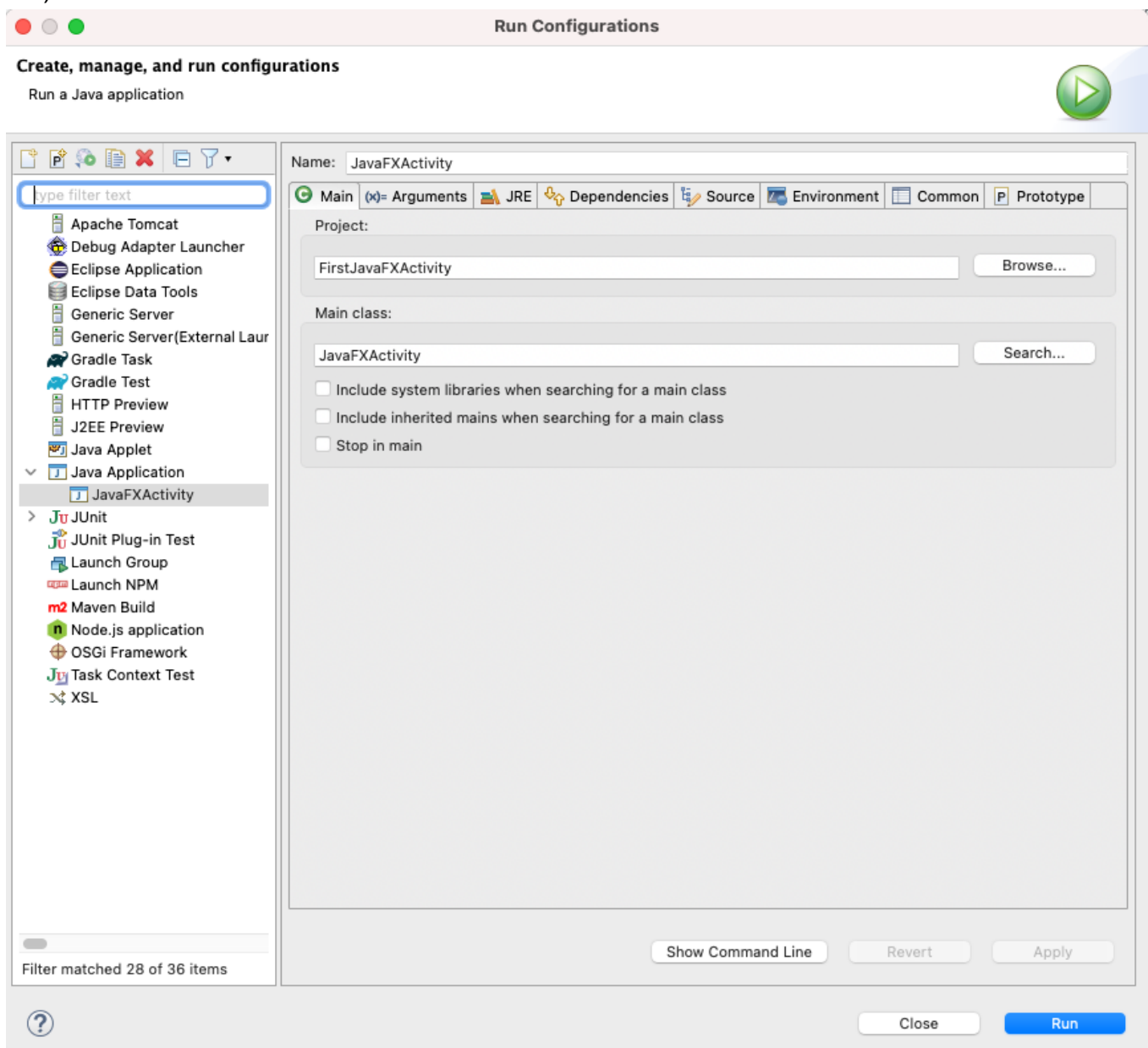
13. When you now try to run the class (right click on **JavaFXActivity.java** and navigate to **Run as -> Java Application**) you will see the following error:



The screenshot shows the Eclipse IDE's console window. At the top, there are tabs for Console, Problems, Debug Shell, Coverage, and Git Staging. The Console tab is active, displaying the following text: `<terminated> JavaFXActivity [Java Application] /Library/Java/JavaVirtualMachines/adoptopenjdk-11.jdk/Contents/Home/bin/java (`. Below this, a red error message is displayed: **Error: JavaFX runtime components are missing, and are required to run this application**.

## Set Run Configurations

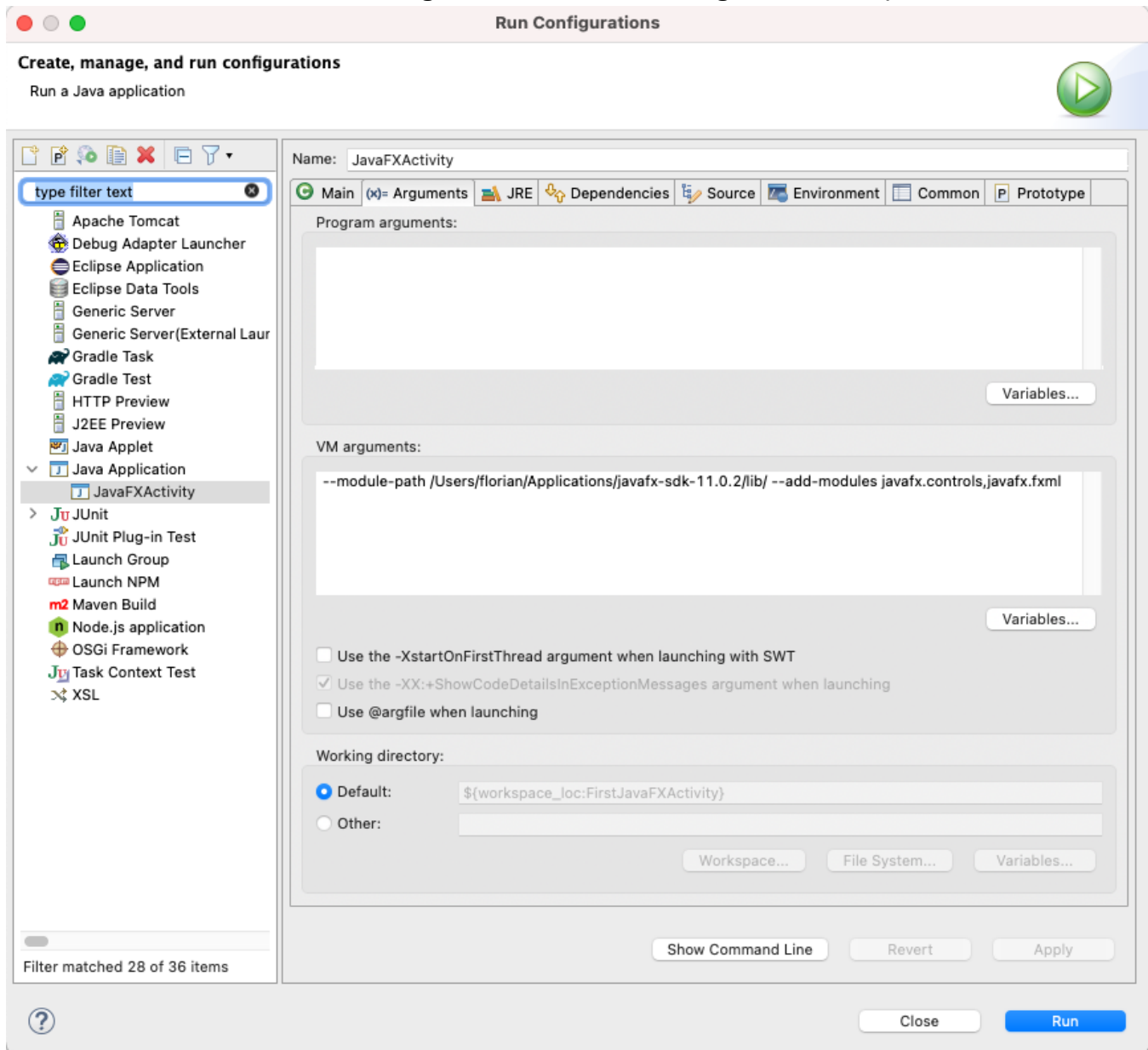
14. Right click on the file **JavaFXActivity.java** in the **Project Explorer** and navigate to **Run as -> Run Configurations**. Make sure **JavaFXActivity** is selected under **Java Application** (on the left).



15. Select the **Arguments** tab and add the following VM arguments (replace **/Users/florian/Applications/javafx-sdk-11.0.2/lib/** with the path on your machine):

```
--module-path /Users/florian/Applications/javafx-sdk-11.0.2/lib/ --add-modules javafx.controls,javafx.fxml
```

16. (This is an important step on OSX, the option might not exist on Windows or Linux) Uncheck the **Use the -XstartOnFirstThread argument when launching with SWT** option.



17. Then click on **Run**. You should see an empty window pop up. Continue filling the window as part of the [First JavaFX Activity \(https://canvas.wisc.edu/courses/244796/pages/12-first-javafx-activity\)](https://canvas.wisc.edu/courses/244796/pages/12-first-javafx-activity)