Supplemental Instructions for JavaFX on JDK 17 in Eclipse
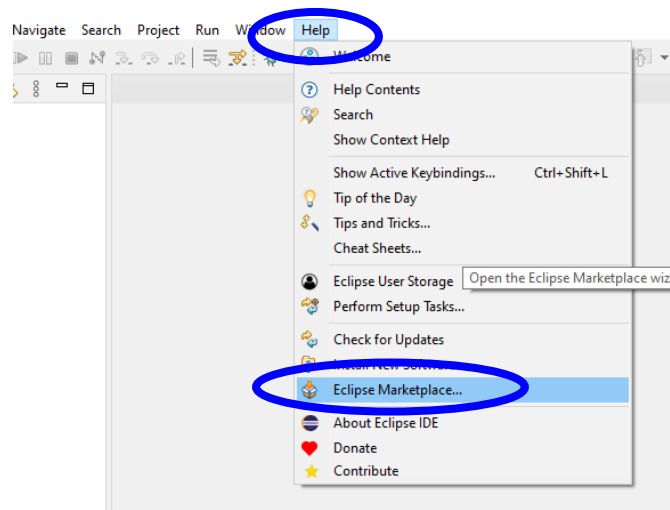
Part 1: Download and Setup JavaFX Libraries for JDK 17
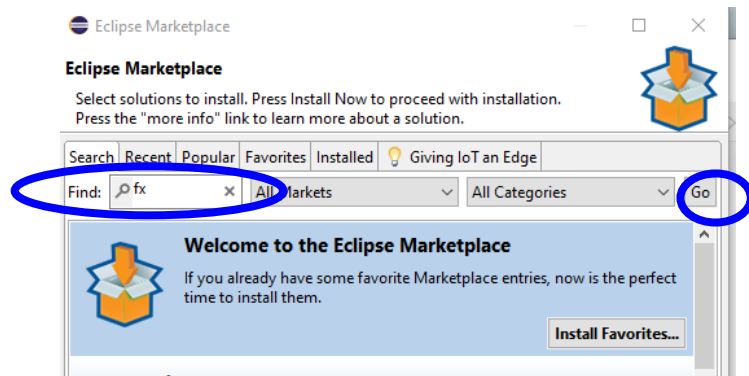
Overview

The JavaFX platform lets you create graphical user interface (GUI) driven applications for desktops, web browsers, and mobile devices. JavaFX libraries were originally included within the core Java SE platform until Java SE 11. The decision to decouple JavaFX from the core JDK allows the JavaFX libraries to be maintained by its passionate open source community. If you are using Oracle JDK11 or later, you must manually download and setup the JavaFX libraries to write JavaFX applications. Instructions to do so in this document are based on material from https://openjfx.io/openjfx-docs/.
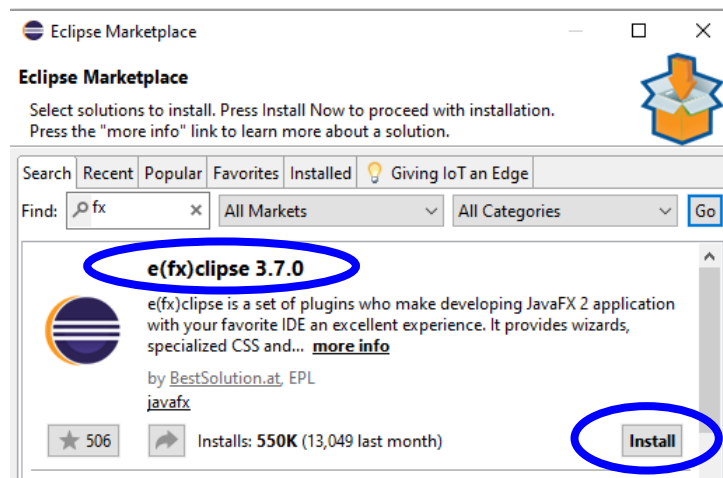
Tasks

1. Navigate to the JavaFX Product page: https://gluonhq.com/products/javafx/

2. Scroll down the page and download the software development kit (SDK) zip for your operating system

3. Inside this zip you'll see a folder such as **javafx-sdk-17.0.1** Extract and add this folder to a known location

4. Your system will refer to this location each time it needs JavaFX
   a. Example for Windows:
      i. Extract to the folder **C:\Program Files\Java**

5. Install the JavaFX plugins to Eclipse:
   a. Start Eclipse and select Help from the menu
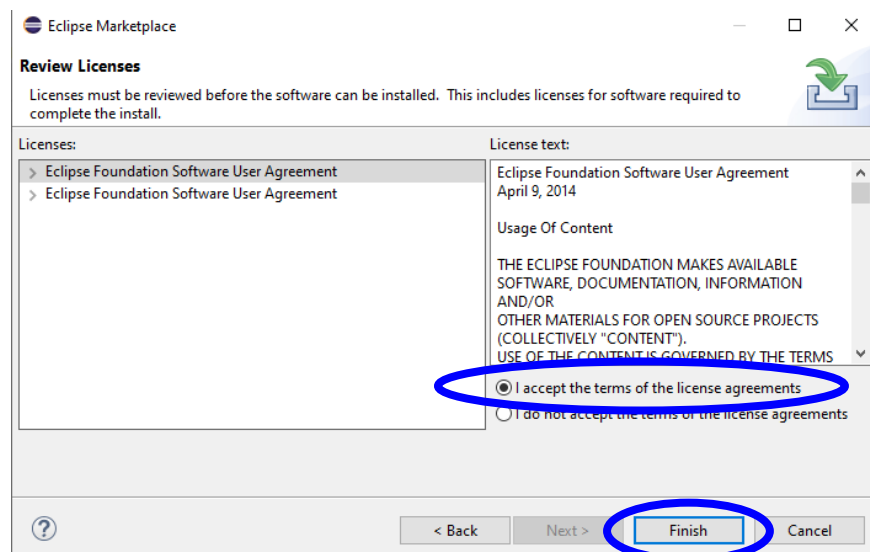   b. Select Eclipse Marketplace from the drop-down menu

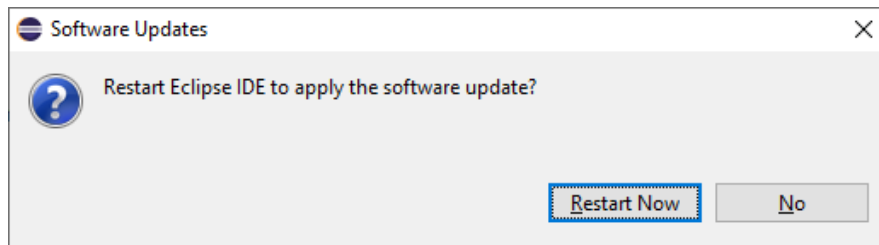   c. Type fx in the search bar, and then click the Go button

d. Select the latest version of e(fx)clipse and click the Install button

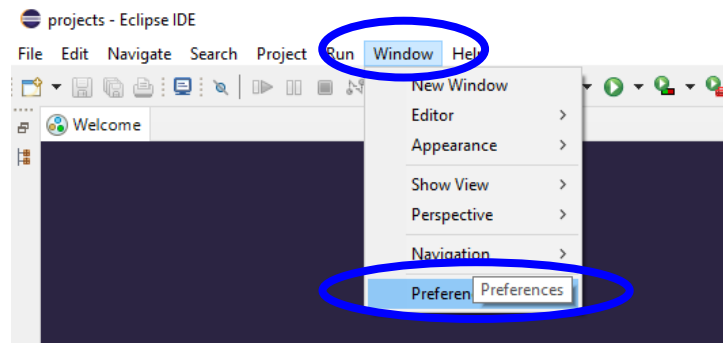e. Agree to the terms of the license agreement and click Finish

f. Eclipse will install the selected software (this may take a few minutes)

g. When the software is installed you will be asked to restart Eclipse, click Restart now to complete the installation
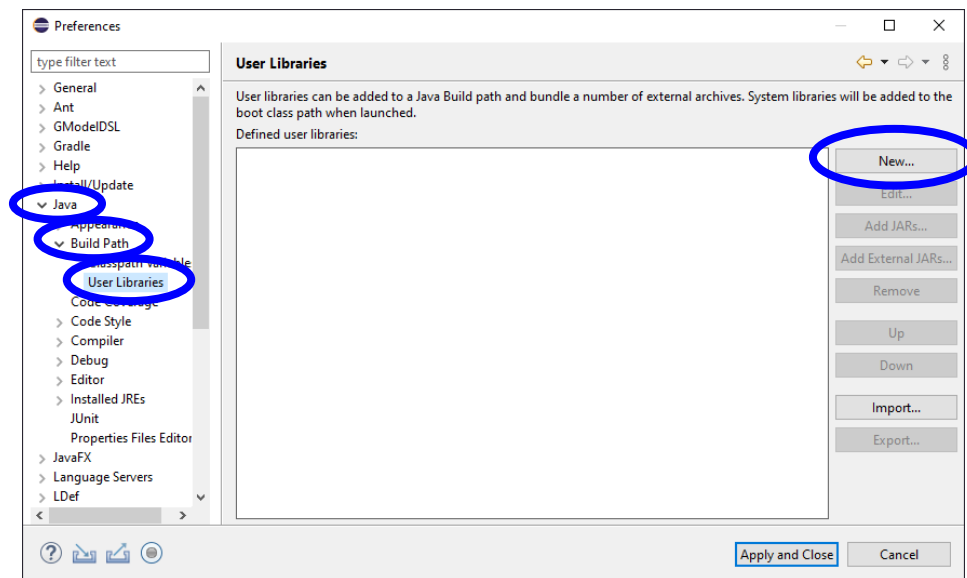
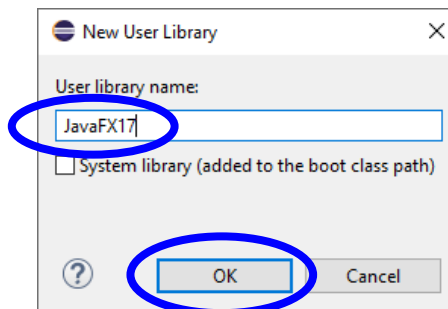6. Add the libraries to Eclipse

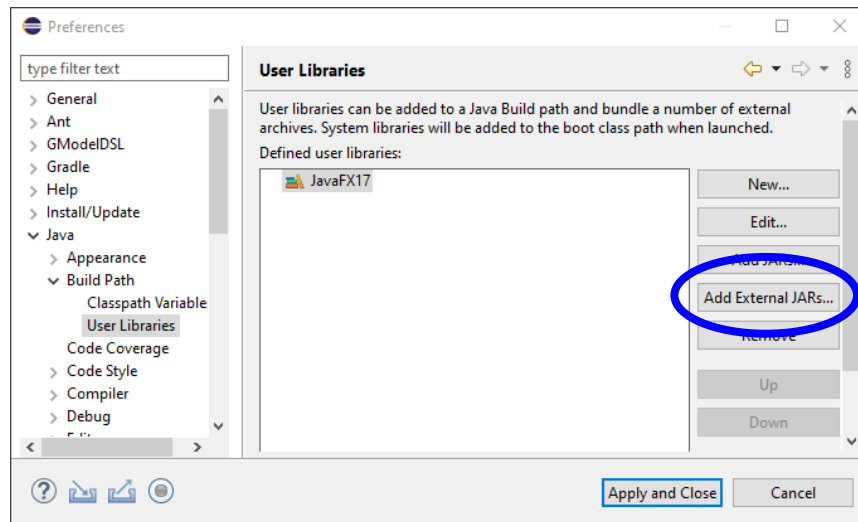   a. In Eclipse, navigate to **Window > Preferences**

    b. From the options expand: **Java>Build Path>User Libraries**. Click **New**
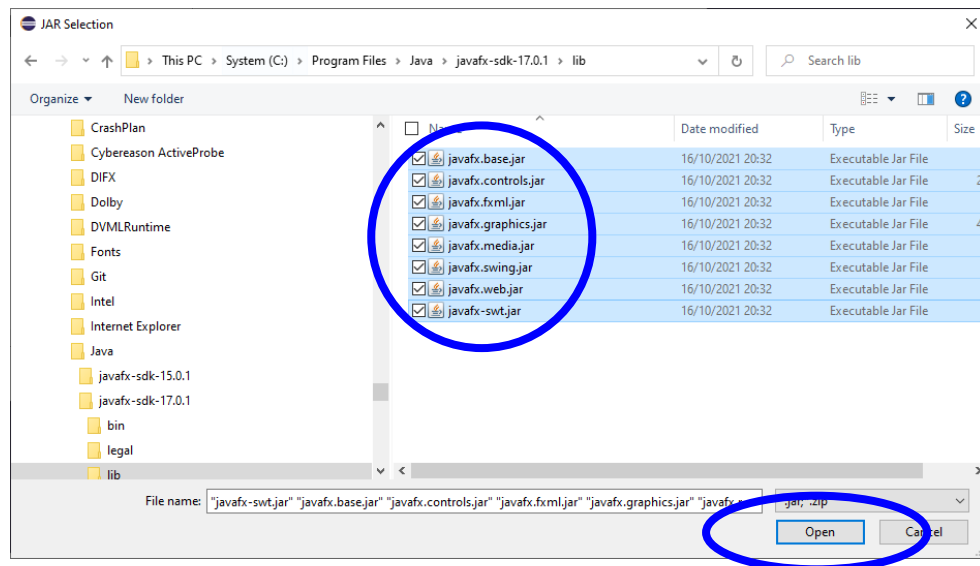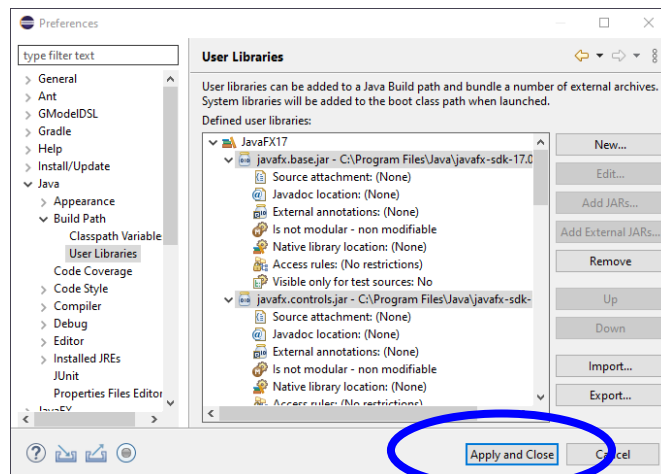

    c. Name the library **JavaFX17** and click **OK**

    d. Click **Add External JARs…**

e. Navigate to the `lib` folder, select the eight JAR files, and click **Open**

f. Click **Apply and Close**

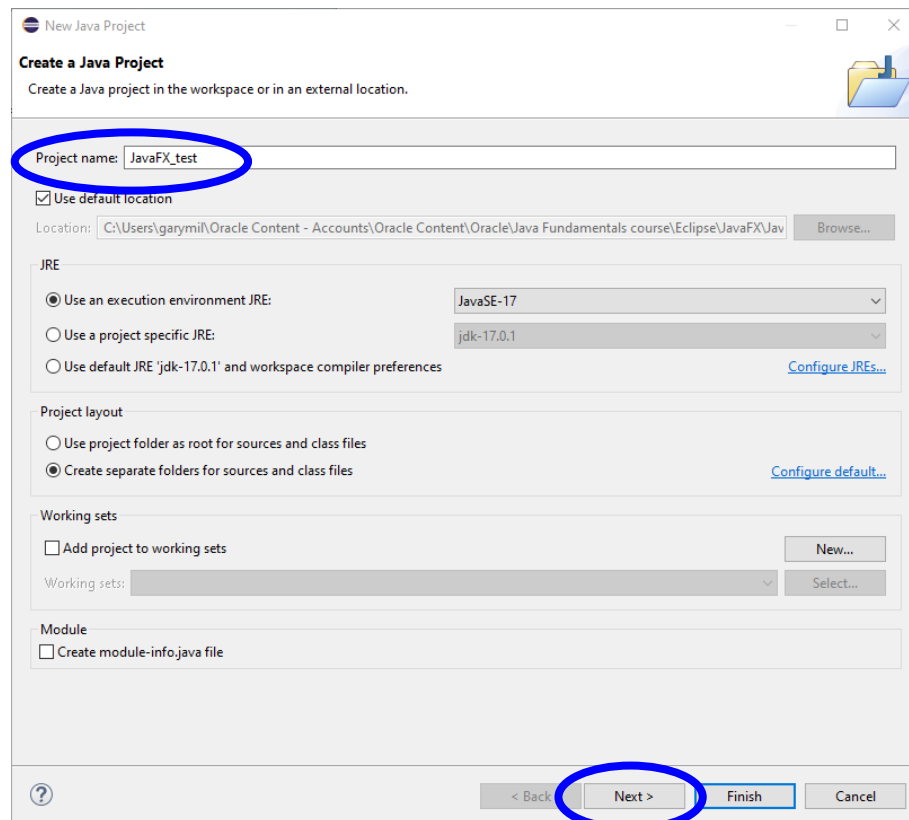## Part 2: Create a New JavaFX Project in Eclipse

Overview

Leading up to JDK 9, Java underwent a process of modularization. Modularization is what allowed for the decoupling of JavaFX libraries from the core JDK. Your programs can also be constructed as modules. Modularization however is not covered in this course. These instructions will show you how to create a non-modular Java FX project.
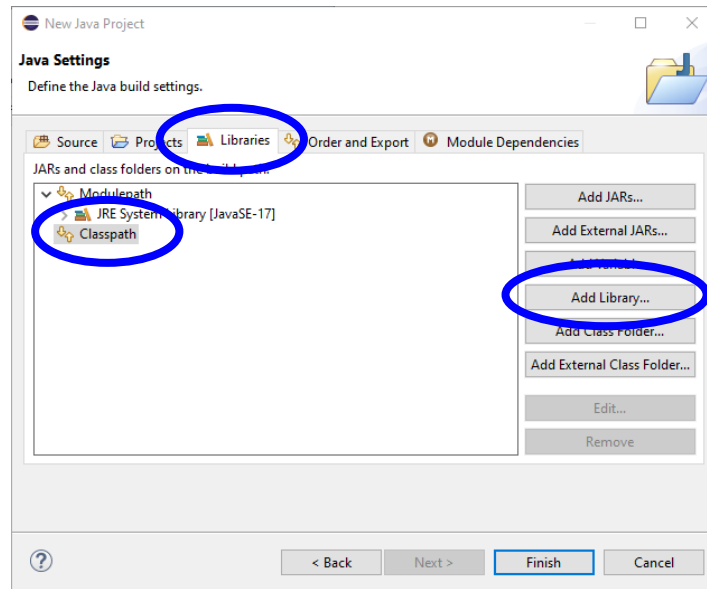
The following tasks are required for every new JavaFX project.
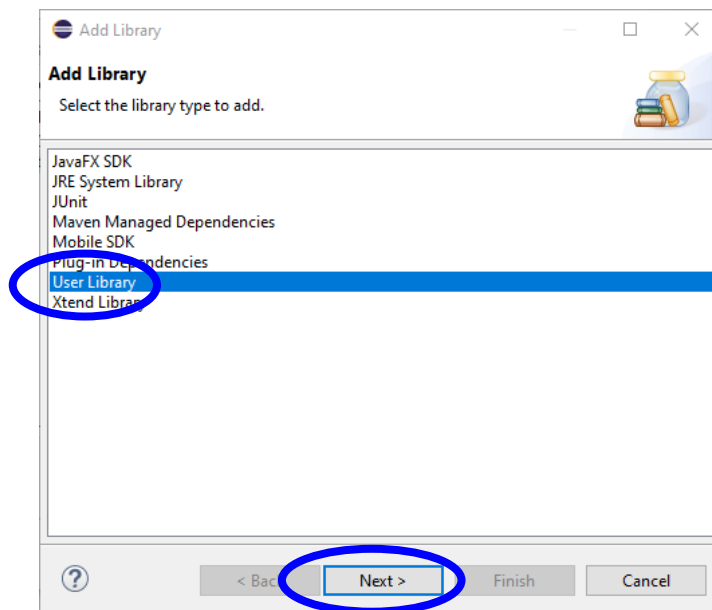
Tasks

1. Ensure that you've completed Part 1

2. Launch Eclipse

3. Go to **File>New>Java Project**
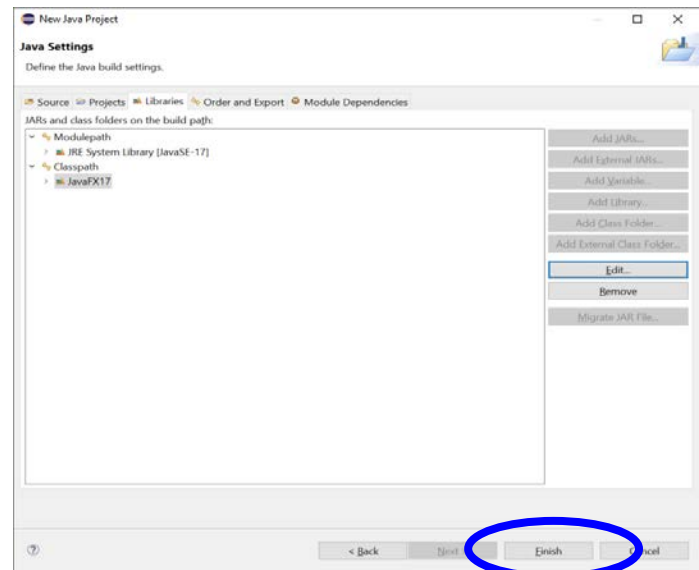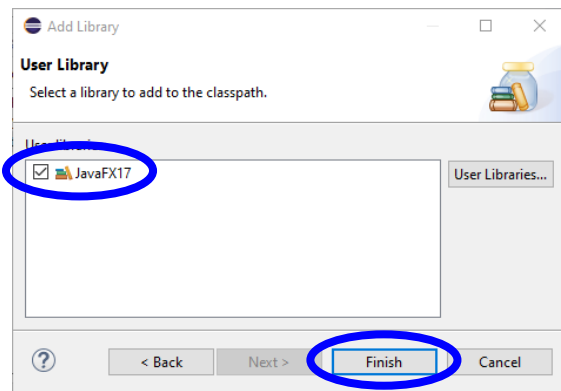
4. **Name the project**, and then click **Next**

5. Click the **Libraries tab**, click **Classpath**, and then **Add Library**
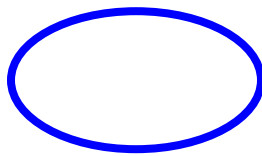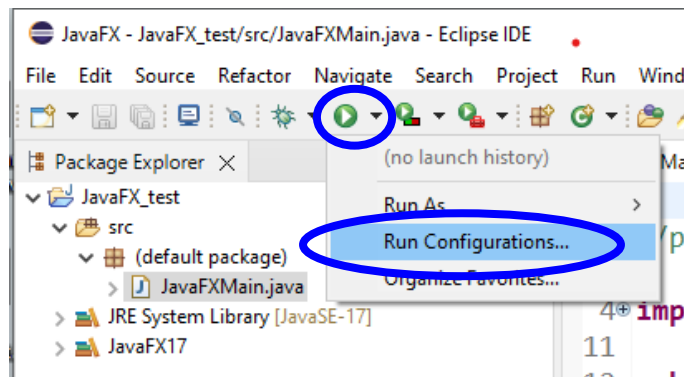

6. Select **User Library> Next**

7. Select **JavaFX17**, which you created in Part 1, and click **Finish** to close the User Libraries tab, then **Finish** to create the project
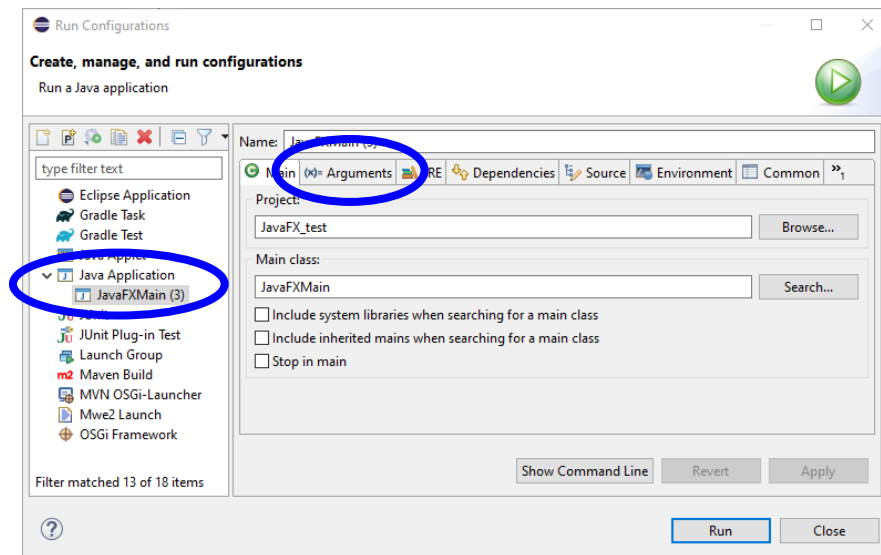

8. Copy the supplied **JavaFXMain.java** file to the project package (ignore any errors at this point)

9. Click the **down arrow** next to the **Run** button, and select **Run Configurations**

10. Double click **Java Application** in the filter pane and click **JavaFXMain**. Click the **Arguments** tab
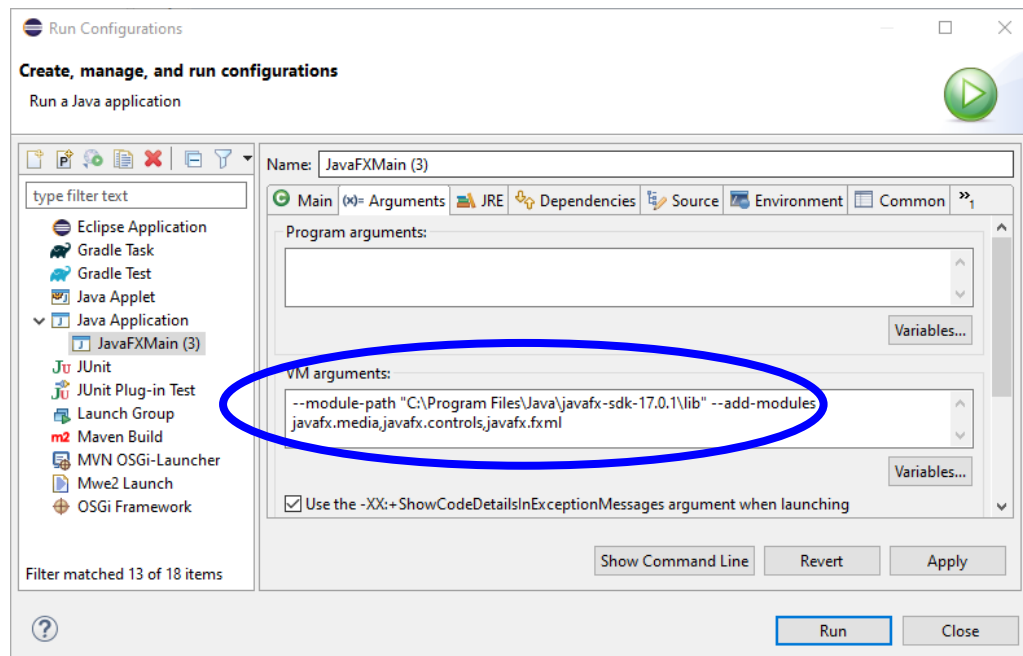

11. Enter the **text below** into the VM Arguments window:

   (Assuming you've added the JavaFX SDK to the location specified in Part 1)

   a.   This helps provide access to the relevant JavaFX modules
   b.   You may need to add more if your projects grow more complex and make use of other aspects of JavaFX
   c.   Update highlighted text below as necessary to match your sdk version

```
--module-path "C:\Program Files\Java\javafx-sdk-17.0.1\lib" --add-modules
                javafx.media,javafx.controls,javafx.fxml
```

12. Click **Run**

As your JavaFX programs grow more complicated, you may need to include other modules for the additional functionality. JavaFX modules include:

- javafx.base
- javafx.control
- javafx.fxml
- javafx.graphics
- javafx.media
- javafx.swing
- javafx.web