Instructions for Adding a JRE containing JavaFX to Eclipse
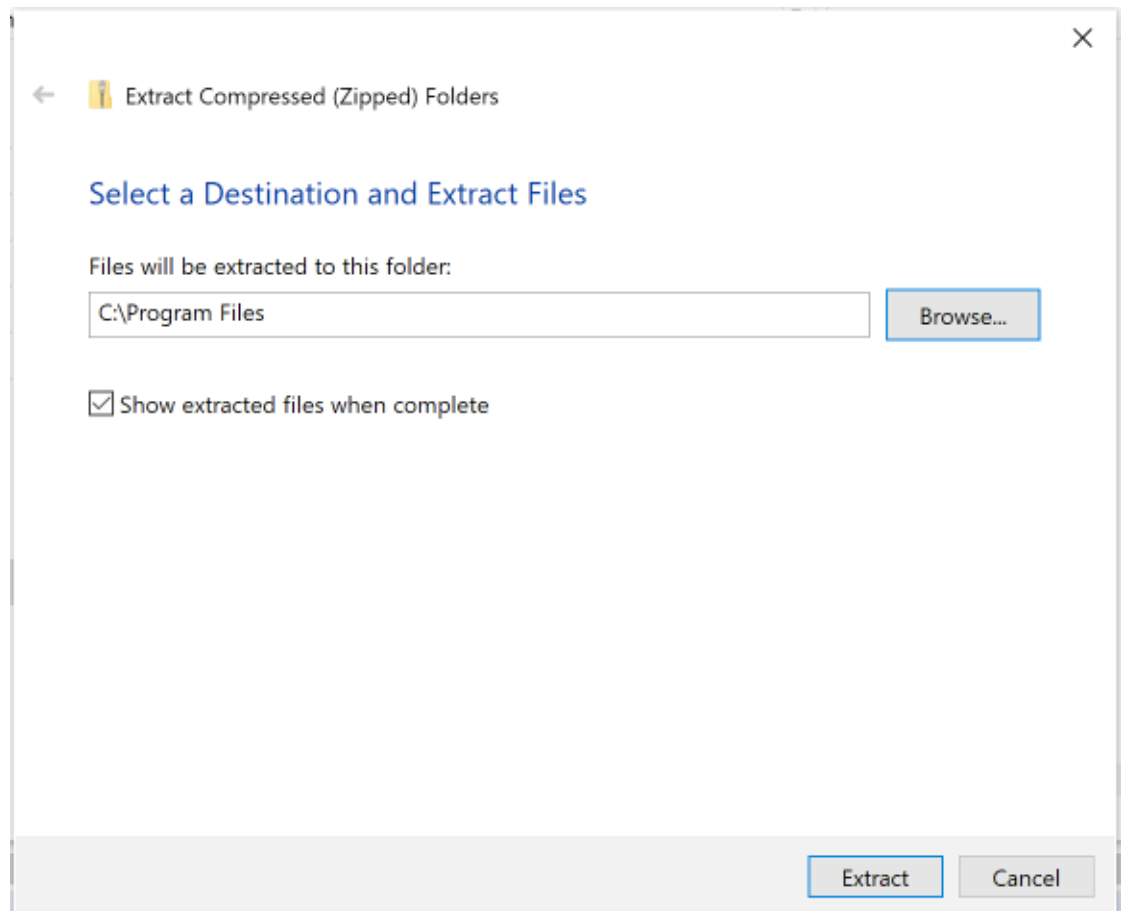
Oracle's version of the Java8 JDK contains JavaFX by default. However in 2019 Oracle changed their licencing agreements so that their version of the JDKs are only free for **Personal Use** and **Product Development** (*but not product distribution*).

As a result, many java-based software products now distribute OpenJDK with their products.

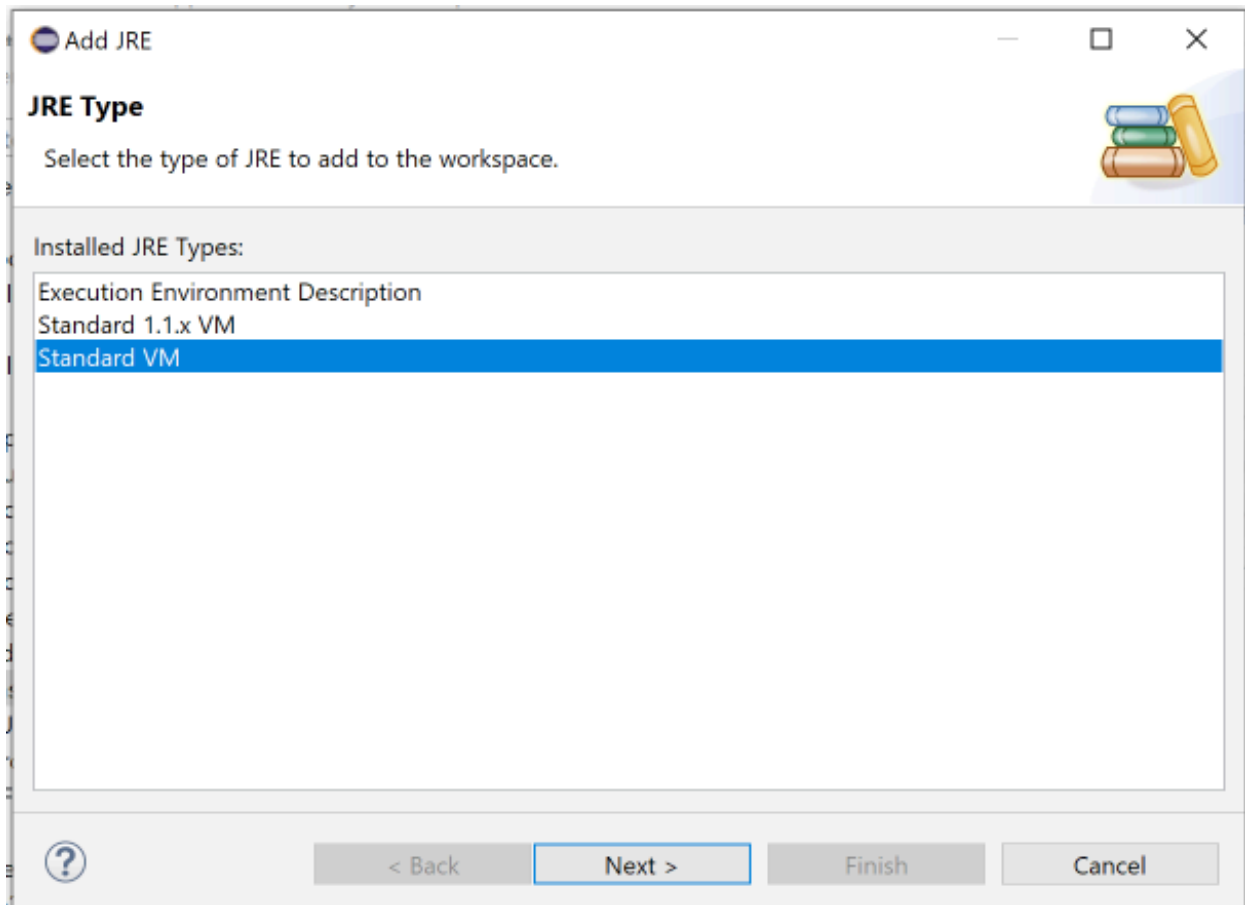**Most versions of OpenJDK do not contain JavaFX.**

If you already had Java8 on your system from products distributing it prior to 2019..the instructions in the Lab should work fine. However, if you only have a version of OpenJDK installed, creating a JavaFX project will result in a lot of "___ *cannot be resolved*" errors. To solve your problem you're going to need to start your JavaFX project with a JDK that contains JavaFX. The following instuctions should help you accomplish this:

1. Download a version of the Java8 JDK that contains JavaFX. You have two options:

    - You can download the offical Oracle version (since what your're doing is essentially Personal Use, you're fine with the new licence agreement) from here: https://www.oracle.com/java/technologies/javase-jdk8-downloads.html *Note: this does require you to create an Oracle account (it's free, mostly just a pain)*. Run the .exe provided and remember where it installs.

    - I've added a zipped version of OpenJDK for Windows (If you're on a Mac and having this problem let me know) that contains JavaFX to Blackboard. Download and unzip it.

        - When you unzip: On the **Select Destination** screen, navigate to your *Program Files* directory extract it there.

2. Once you've got a JDK available, you'll need to configure Eclipse to allow you to choose it when creating a project. In Eclipse go to *Window > Preferences > Java > Installed JREs*

3. Press *Add*, then choose *Standard VM*

4. In the *JRE Home*, press the *Directory* button and navigate to the folder that contains your JDK.

    ○ If you went with the Oracle option, it's probably *Program Files / Java / jdk..*
    ○ If you went with the OpenJDK I provided and extracted correctly, it should be *Program Files / zulu_jdk...

5. Select the entire folder and choose *Finish*, then press *Apply and Close*

Add JRE

JRE Definition

Specify attributes for a JRE

JRE home:	C:\Program Files\zulu_jdk_fx_1.8	Directory...
JRE name:	zulu_jdk_fx_1.8	
Default VM arguments:		Variables...

JRE system libraries:

> C:\Program Files\zulu_jdk_fx_1.8\jre\lib\resources.jar
> C:\Program Files\zulu_jdk_fx_1.8\jre\lib\rt.jar
> C:\Program Files\zulu_jdk_fx_1.8\jre\lib\jsse.jar
> C:\Program Files\zulu_jdk_fx_1.8\jre\lib\jce.jar
> C:\Program Files\zulu_jdk_fx_1.8\jre\lib\charsets.jar
> C:\Program Files\zulu_jdk_fx_1.8\jre\lib\jfr.jar
> C:\Program Files\zulu_jdk_fx_1.8\jre\lib\ext\access-bridge-64.jar
> C:\Program Files\zulu_jdk_fx_1.8\jre\lib\ext\cldrdata.jar
> C:\Program Files\zulu_jdk_fx_1.8\jre\lib\ext\dnsns.jar
> C:\Program Files\zulu_jdk_fx_1.8\jre\lib\ext\jaccess.jar
> C:\Program Files\zulu_jdk_fx_1.8\jre\lib\ext\jfxr.jar

Add External JARs...
Javadoc Location...
Source Attachment...
External annotations...
Remove
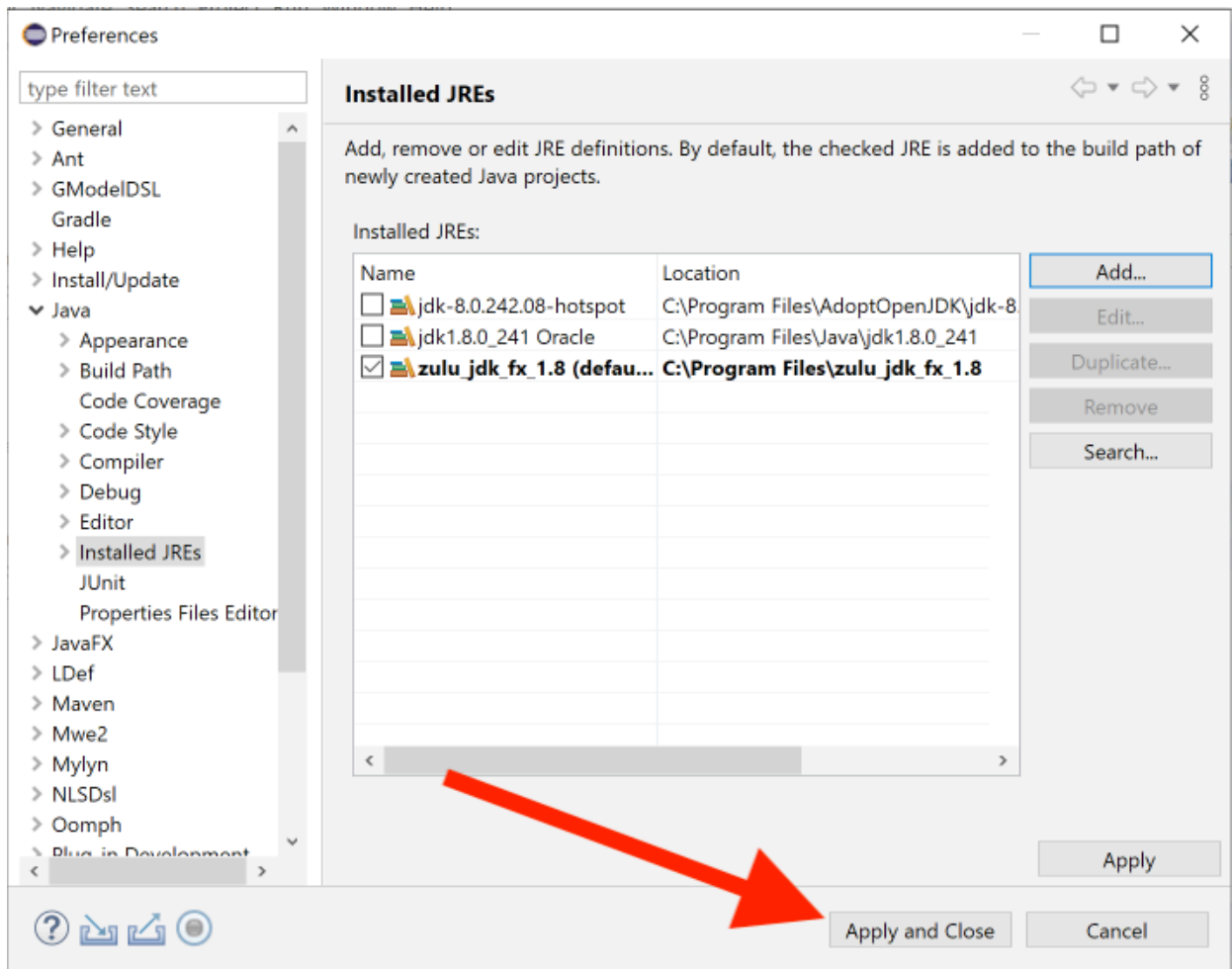Up
Down
Restore Default

< Back   Next >   Finish   Cancel

6. Now create your JavaFX project following the instructions in the Lab. But on the project screen, you need to choose the JDK you just added.

7. Now you should be able to complete the Lab as instructed, any other JavaFX projects you need to create.