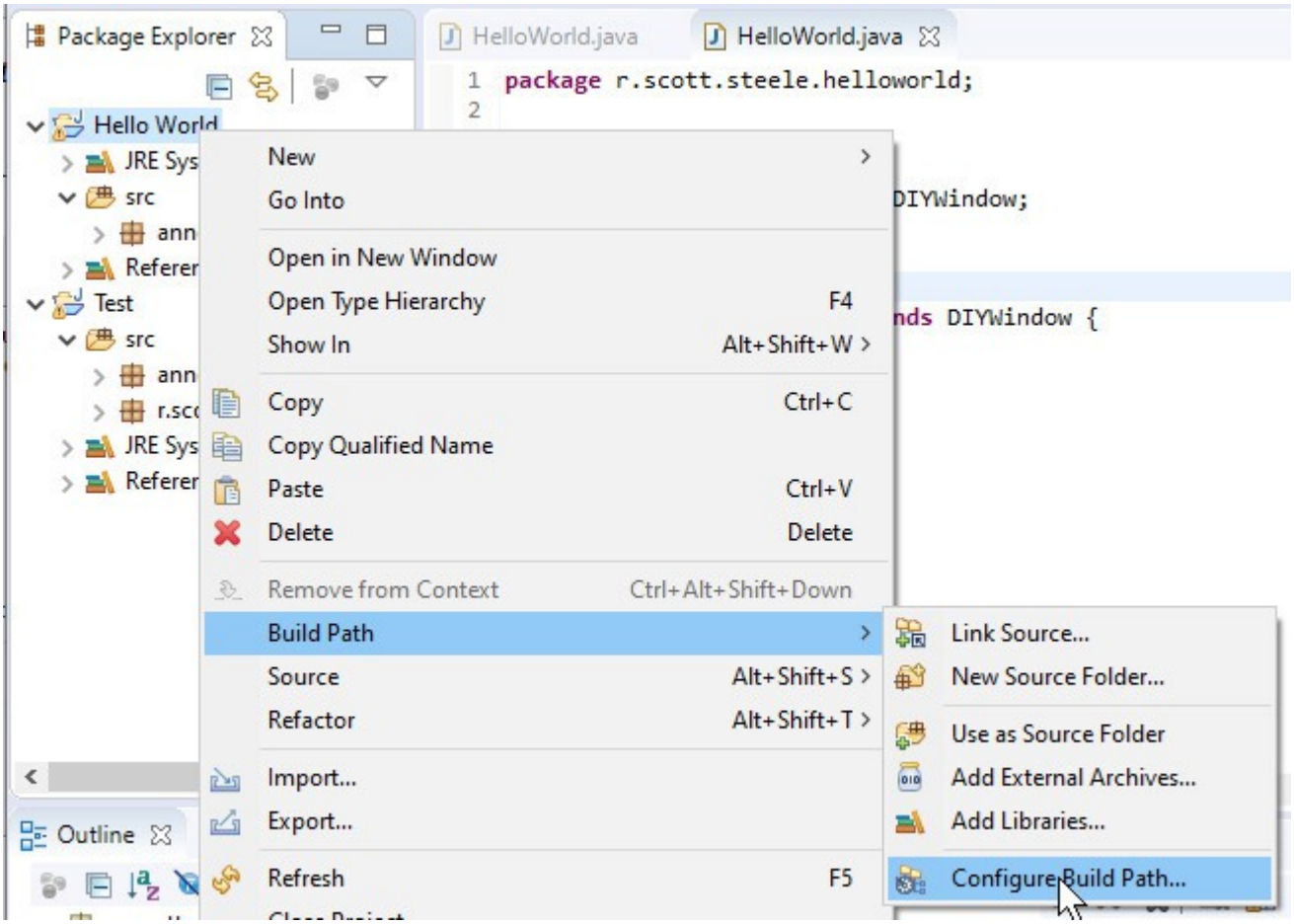
If You Got *NoClassDefFoundError* Error

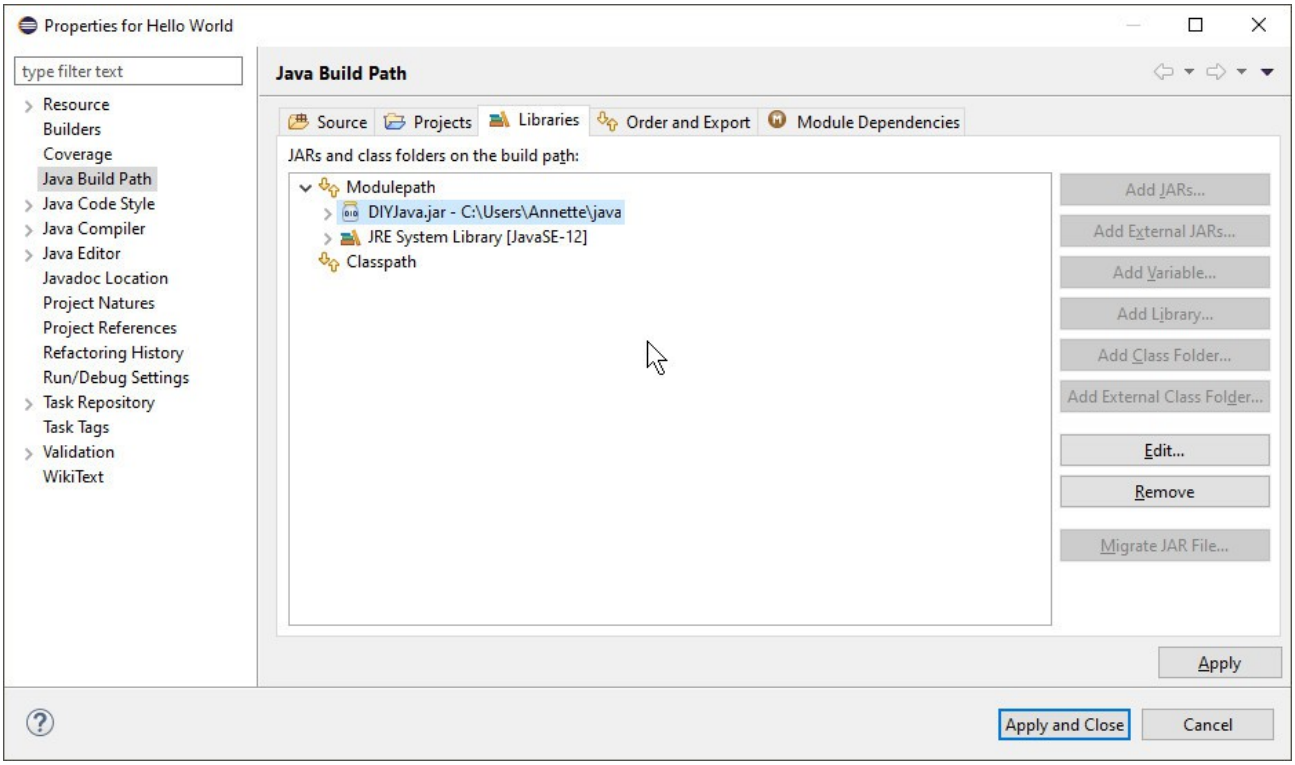
Error: Could not find or load main class xxxxx.helloworld.HelloWorld
Caused by: java.lang.NoClassDefFoundError: com/godtsoft/diyjava/DIYWindow

This message happens if the DIYWindow was added to the Modulepath instead of the Classpath. Eclipse then cannot find DIYWindow. To fix this:

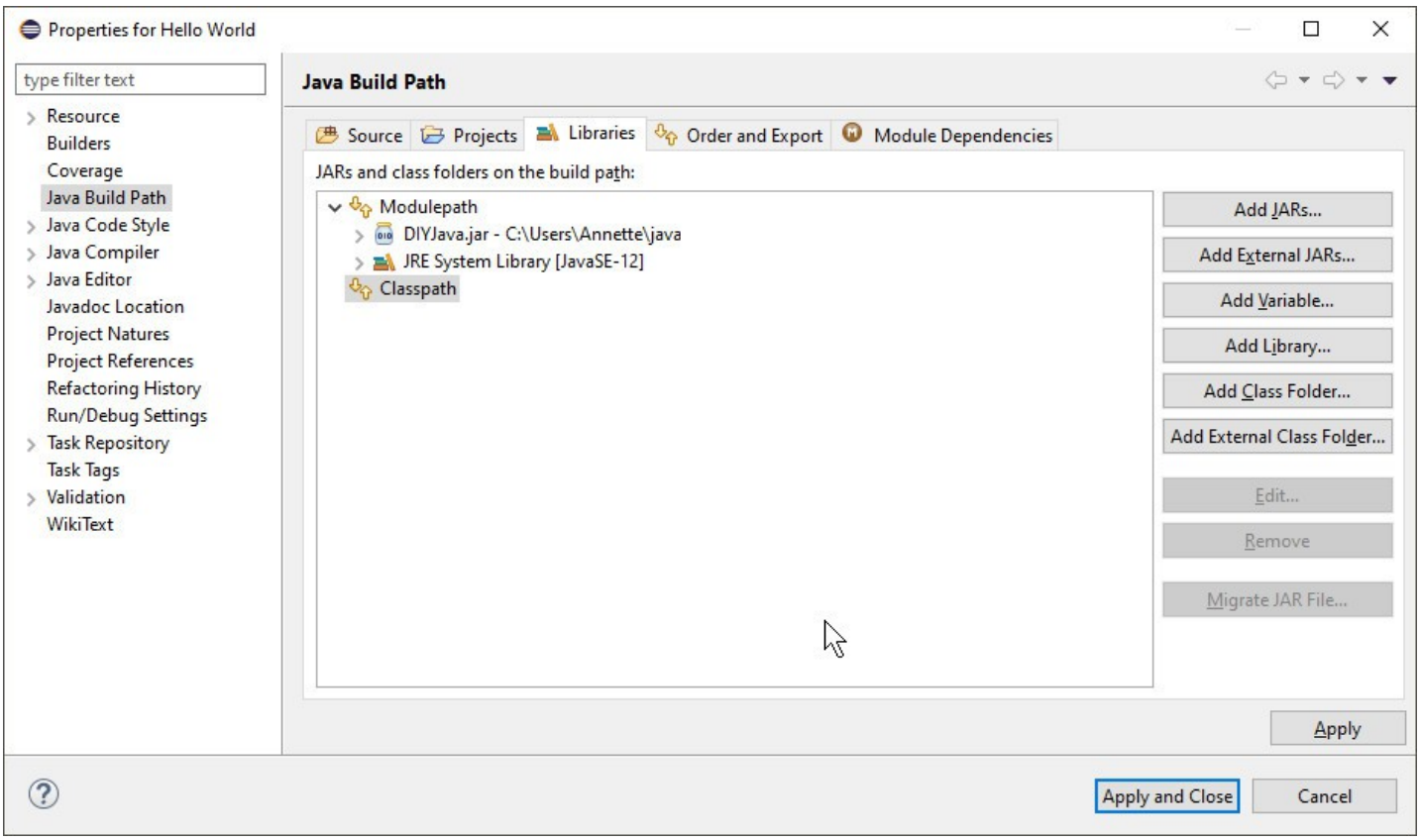
Right click on the Hello World Project in the Package Explorer of Eclipse and choose *Build Path / Configure Build Path...*



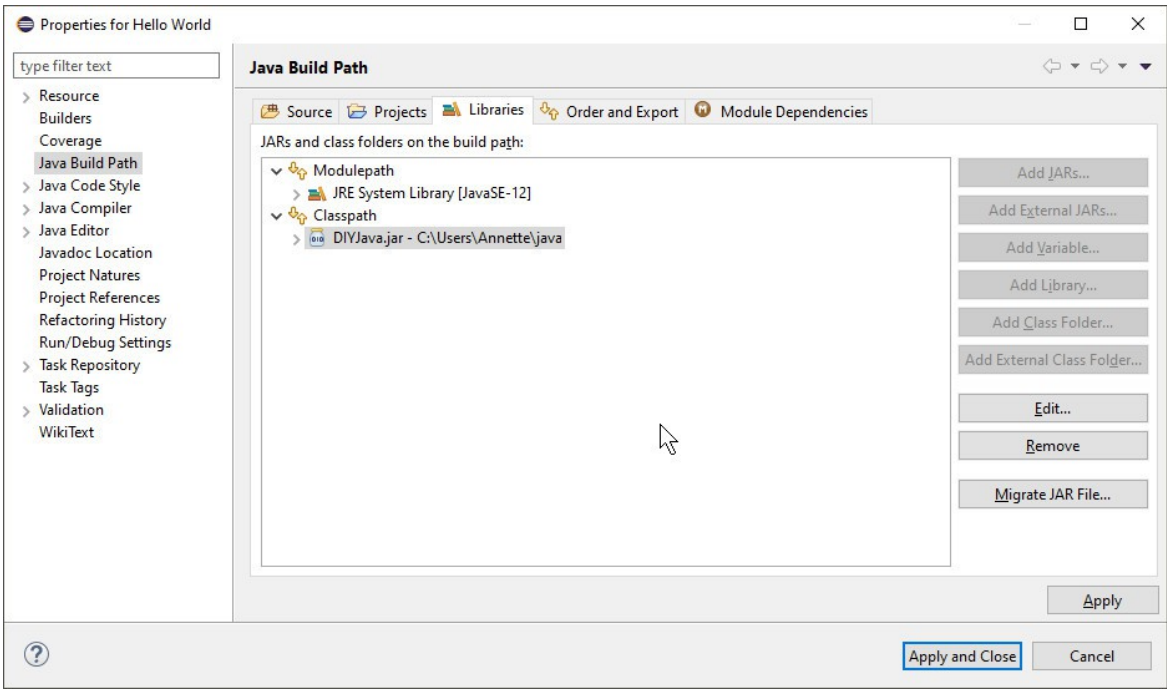
If DIYJava.jar is in the Modulepath, select it and click *Remove*.



Now, click *Classpath* and click *Add External JARs...*



Browse to where you saved the DIYJava.jar file and select *DIYJava.jar*. DIYJava.jar should now be in your classpath.

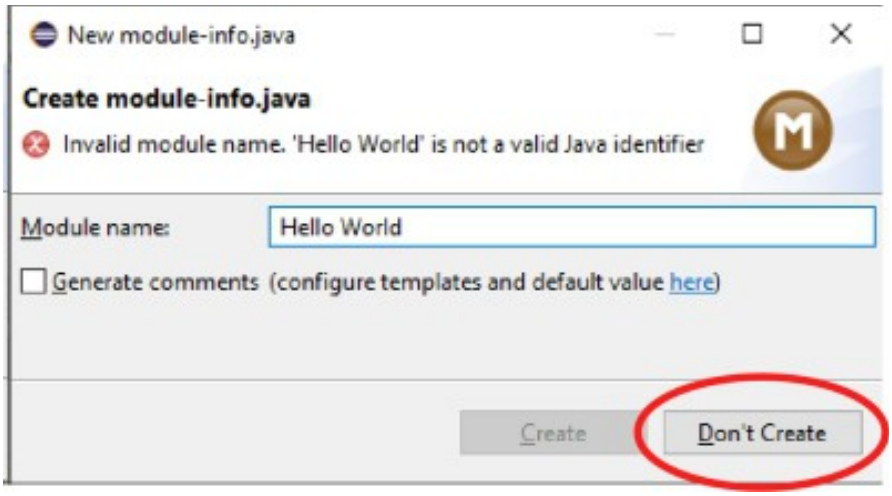


Click *Apply and Close*.

Now when you run your HelloWorld program it should be able to find the HelloWorld class and you should no longer get that error.

You Don't Need to Create module-info.java for Your Project...

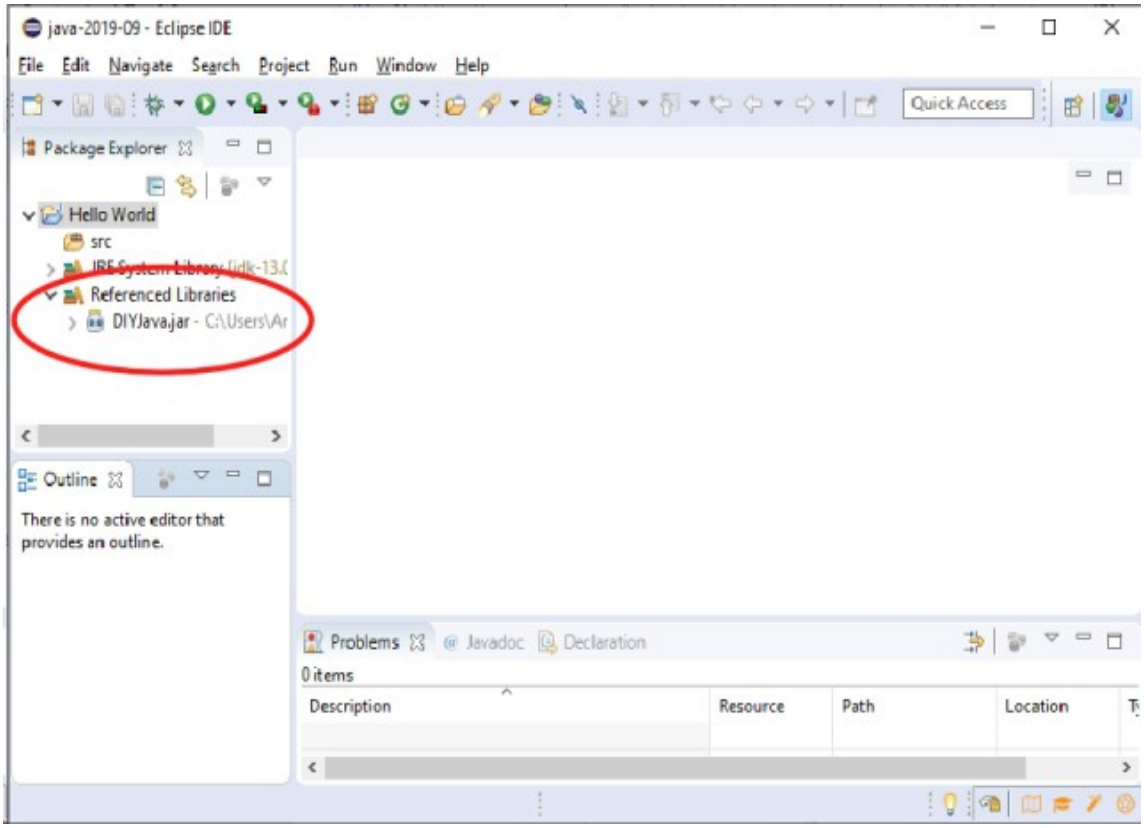
Modules are a new packaging option introduced after these books were written. None of the projects in these books use modules. If you see a window to create a module-info.java file, click *Don't create*.



1. Click *Don't Create*.

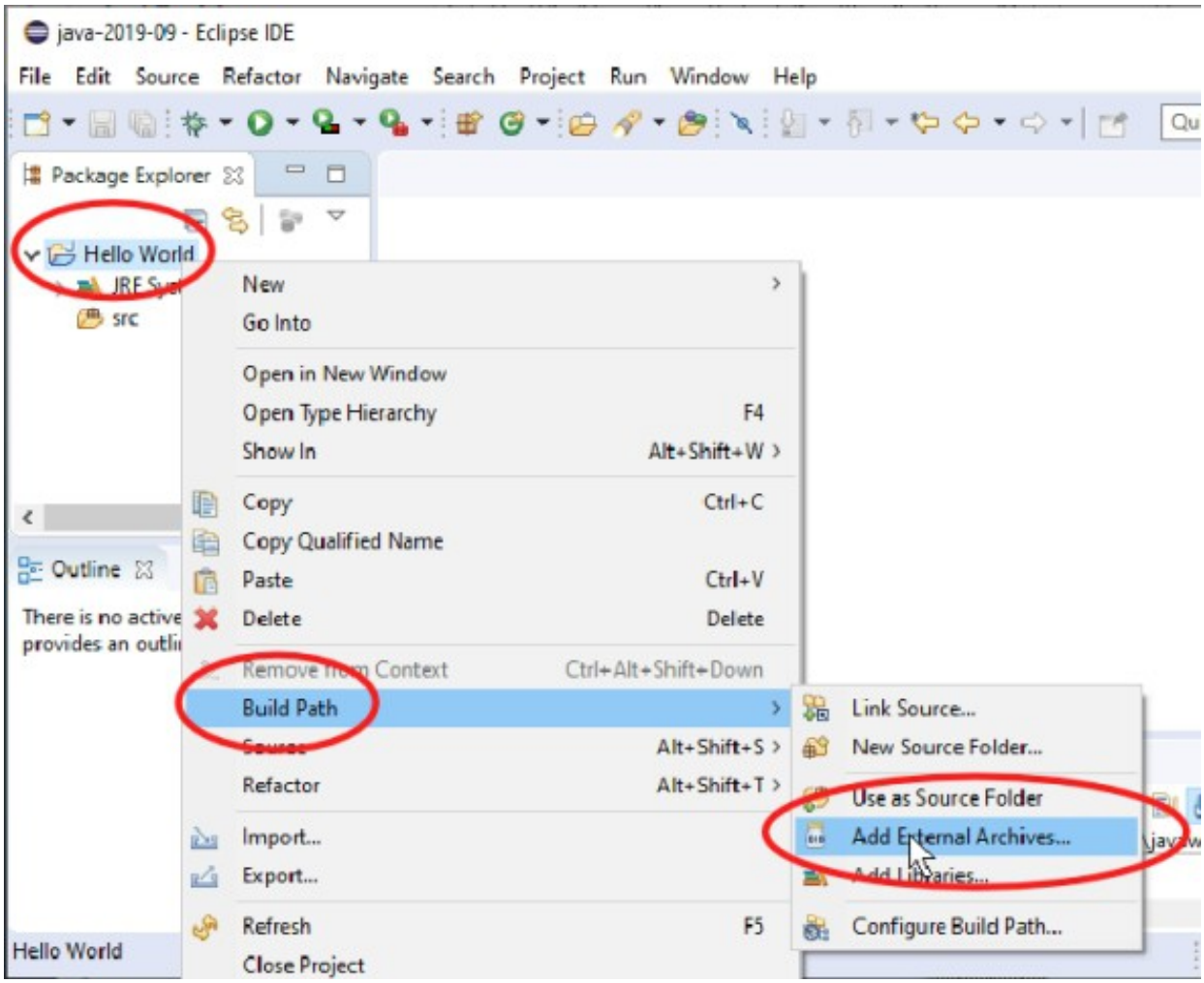
If You Were Unable to Add DIYJava.jar as the External JAR File...

There is a problem with some versions of Eclipse. After completing Lesson 1.1, if you do not see the DIYJava.jar file in the Package Explorer window as shown here, follow the steps below to add DIYJava.jar to the project.



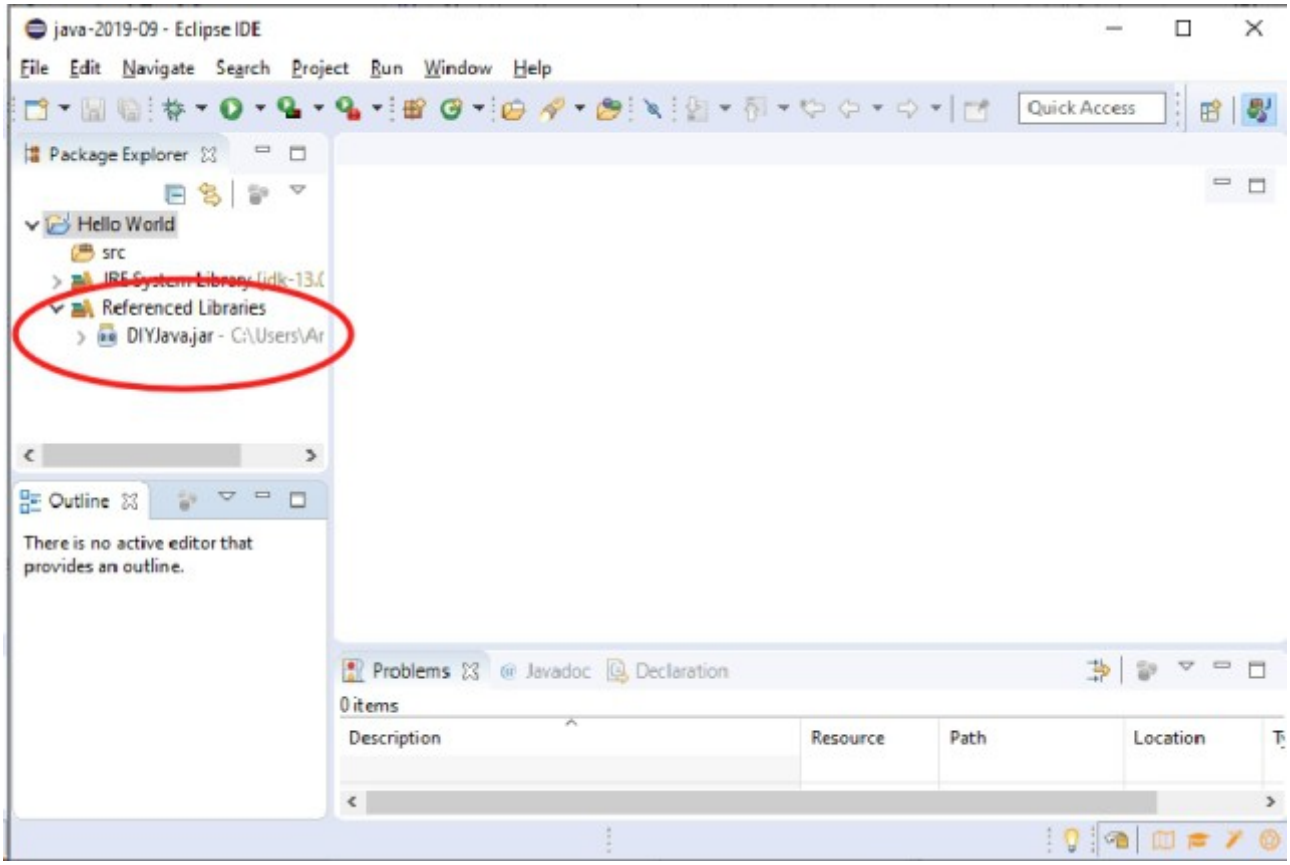
Add DIYJava.jar as an External JAR file to an existing project:

1. Right click on the existing project in Package Explorer.
2. Select *Build Path*.
3. Select *Add External Archives...*



4. Browse to and select *DIYJava.jar*, which you installed in your Java work folder, and click *Open*.
5. Click *Finish*.

The Package Explorer pane should now show the added JAR file (DIYJava.jar).



If Eclipse Can't Find the Latest Installed JDK.

This error occurs if the JAVA_HOME System Environment Variable is not set up correctly. To set the JAVA_HOME and PATH System Environment Variables, do the following:

- ✔ Go to *Start / Windows System / Control Panel*, click *View by:*, and select *Large icons*.
- ✔ Click *System / About*.
- ✔ Scroll down and click the *Advanced system settings* link.
- ✔ Click *Advanced / Environment Variables...*.
- ✔ Under *System variables*, if *JAVA_HOME* exists, select it and click *Edit*. If *JAVA_HOME* does not exist, click *New*.
  - ✔ Set *JAVA_HOME* to the directory of the newly installed JDK and click *OK*. For example, *C:\Program Files\Java\jdk-18.0.1.1*.
- ✔ Under *System variables*, select *PATH* and click *Edit / New*. Add *%JAVA_HOME%\bin* **to the beginning** of the list and click *OK*.
- ✔ Under *User variables*, if *Path* exists, select *Path* and click *Edit*. Add *%JAVA_HOME%\bin* **to the beginning** of the list and click *OK*. If *Path* doesn't exist, don't add it. It should automatically default for your user to use the system variable *PATH*.
- ✔ Click *OK* to close the Environment Variables window.
- ✔ Click *OK* to close the System Properties window.

- ✔ Close the Control Panel window.

Next, make sure Eclipse uses the latest JDK.

- ✔ Restart Eclipse so it uses the latest JAVA_HOME System Environment Variable.
- ✔ Click *Window / Preferences /Java / Installed JREs*.
- ✔ Select the newest JDK in the list.
- ✔ Click *Apply and Close*.

If the newest JDK is not in the list of Installed JREs, add the new JDK to the list.

- ✔ In Eclipse, click *Window / Preferences / Java / Installed JREs*.
- ✔ Click *Add* to get to the *Add JRE* window.
- ✔ Choose to add a *Standard VM* and click *Next*.
- ✔ Click *Directory...* to get to the JDK folder that contains the bin folder. It may be something like *C:\Program Files\Java\jdk-18.0.1.1*. Click *Select Folder* after you select the folder.
- ✔ Click *Finish*.
- ✔ Select the JDK you just added as the new default JRE.
- ✔ Click *Apply and Close*.

The Do-It-Yourself Java Games Series of Books

The books currently available in the series:

1. Do-It-Yourself Java Games: An Introduction to Java Computer Programming - learn the fundamentals of Java programming as you create fourteen different text-based games. No previous programming experience is required.
2. More Do-It-Yourself Java Games: An Introduction to Java Graphics and Event-Driven Programming - learn how to program windows, icons, and menus as you create ten more colorful, more interactive games.
3. Advanced Do-It-Yourself Java Games: An Introduction to Java Threads and Animated Video Games - learn to control multiple, simultaneous activities as you create eight more lively, audible games.
4. Do-It-Yourself Multiplayer Java Games: An Introduction to Java Sockets and Internet-Based Games - Learn to communicate across a network as you create seven games that you can play against friends on other computers.

Questions?

Email further questions to Annette Godtland at diyjava@godtlandsoftware.com