

Installing and Using Eclipse

1. Installing Eclipse

Before downloading and installing Eclipse, ensure that you have downloaded and installed the Java SE Development Kit (referred to as the JDK or SDK) from the Oracle site.

The latest version can be downloaded from here:

<https://www.oracle.com/uk/java/technologies/downloads/>

Once you have downloaded and installed the JDK you can then download and install Eclipse from the following site:

<https://www.eclipse.org/downloads/>

Once you have installed both the JDK and Eclipse you will be in a position to run all the programs in this book, except those that involve JavaFX. Running JavaFX programs requires some additional steps - this is dealt with in section 3.

2. Using Eclipse

When you first launch Eclipse a screen will appear asking you to select a directory as your workspace, as shown in Figure 1. Once you have made your selection, you can press the *Launch* button.

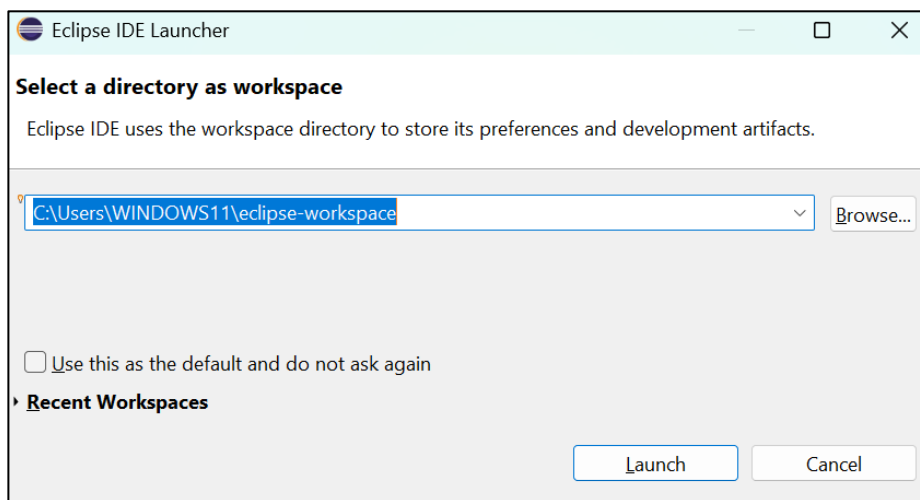


Figure 1

2.1 Starting a new project

Each project in Eclipse consists of a number of classes and sometimes additional files - for example image files. While you are getting started, it might be a good idea to start a new project for each chapter of the book. When you move on to larger applications, such as the case study in chapters 11 and 12, or applications that you are creating yourself (such as a class assignment), you should create a single project for your application.

To create a project, choose **File > New > Java Project** as shown in Figure 2.

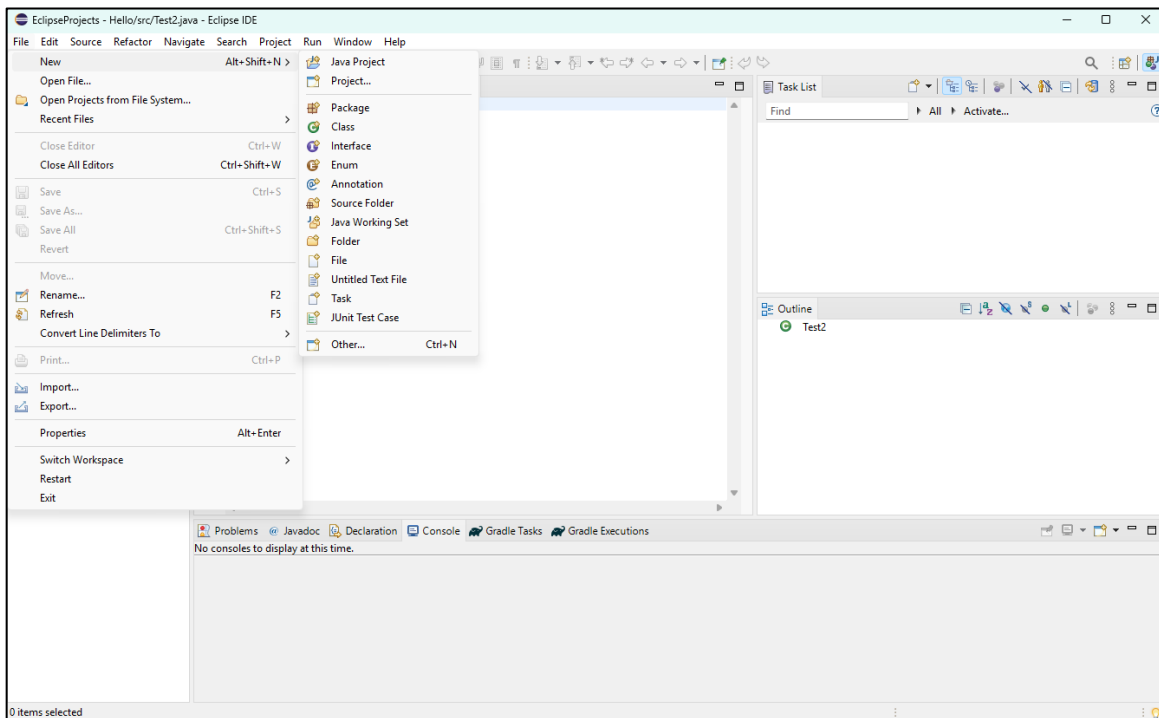


Figure 2

You will now see the screen shown in Figure 3. Choose a name for your project, such as *FirstProject*, then select the execution environment from the drop down list on the right - the one corresponding to the JDK you downloaded should appear in the list.

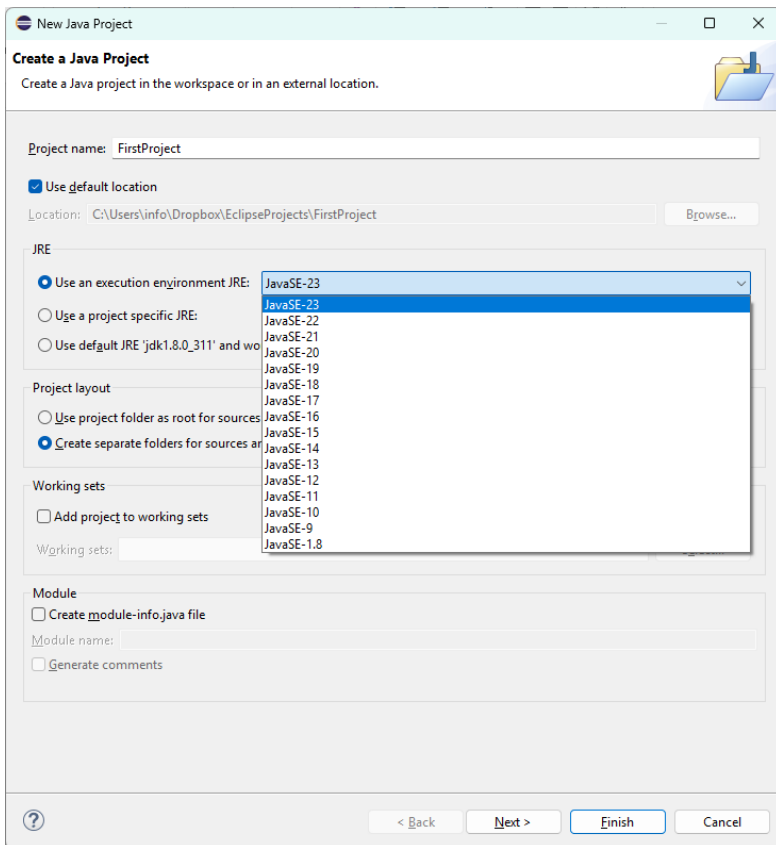


Figure 3

You can now press **Finish**.

You will see the screen shown in Figure 4

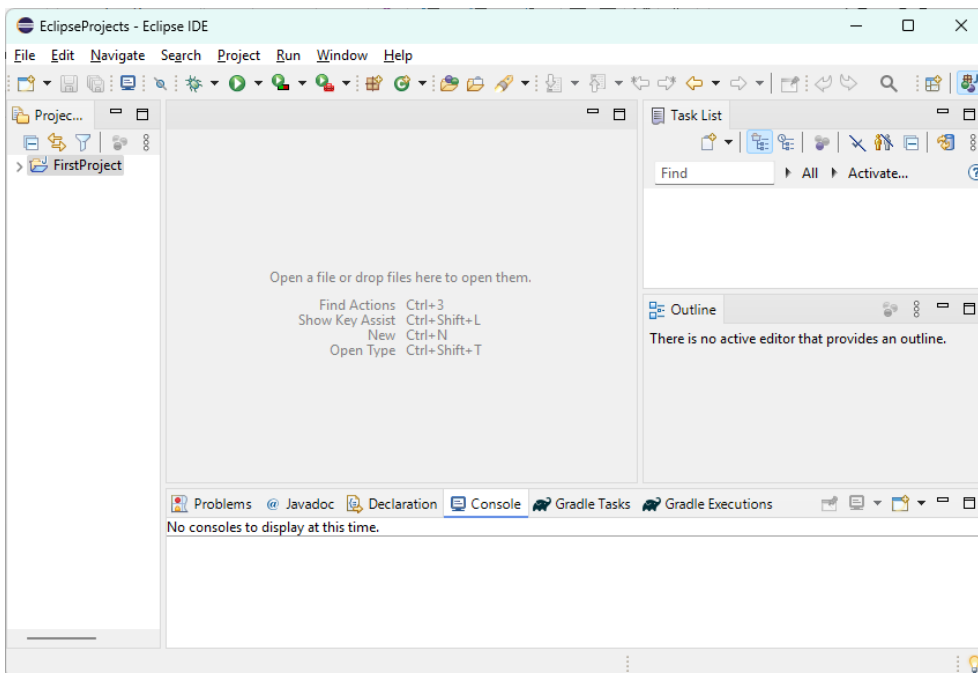


Figure 4

Open projects are listed on the left (there is only one at the moment). If you expand this entry in the list, you will see that it contains a folder called `src`, which is short for source. This is where your source code files will be kept (see Figure 5).

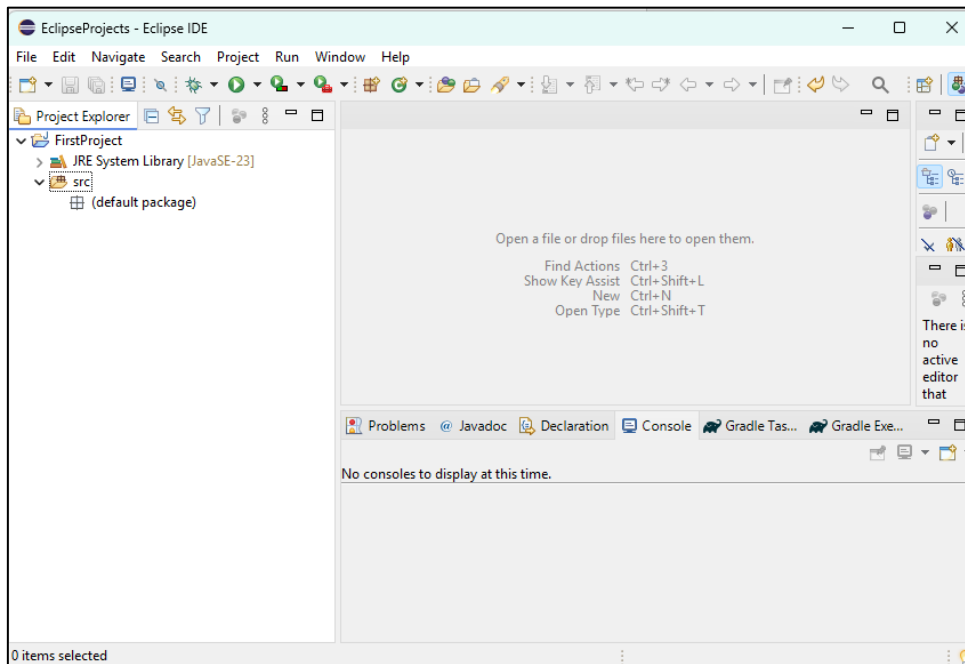


Figure 5

2.2 Adding new classes to the project

Choose **File > New > Class** from the top menu - or alternatively right click on the `src` folder and choose **New > Class**. The screen shown in Figure 6 will appear.

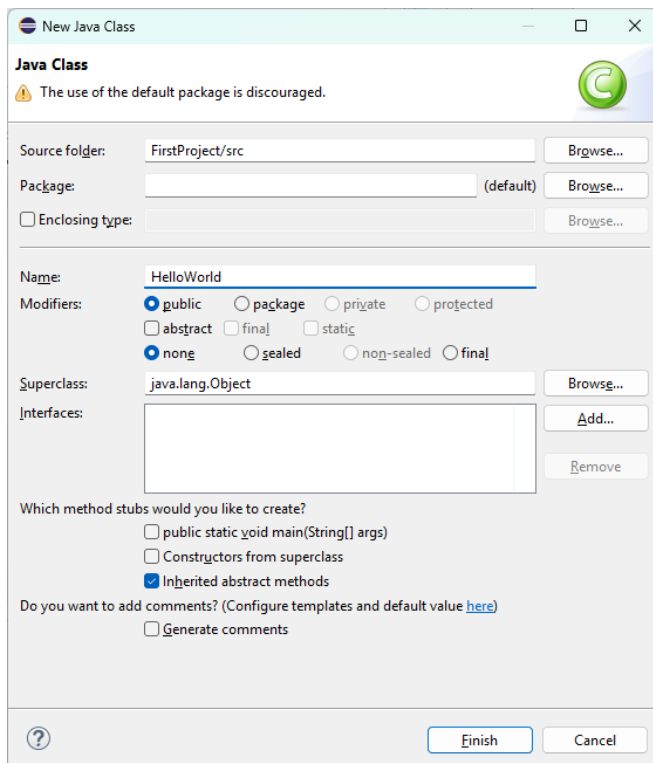


Figure 6

Choose a name for your class - we have chosen **HelloWorld**. Leave the Package field blank - you may see a message discouraging use of the default package. You can safely ignore this for now (this is a message for professional Java developers - we are not yet ready to organise our applications into packages).

As you can see from Figure 7, your Java class is now listed on the right, and the class header is provided for you in the code window.

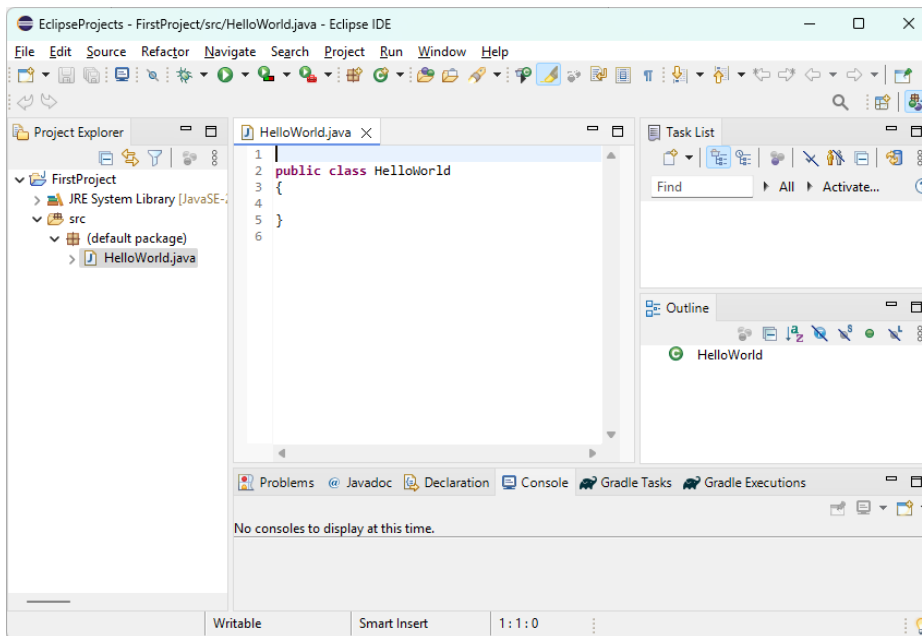


Figure 7

You can now complete the code for your class as shown in Figure 8.

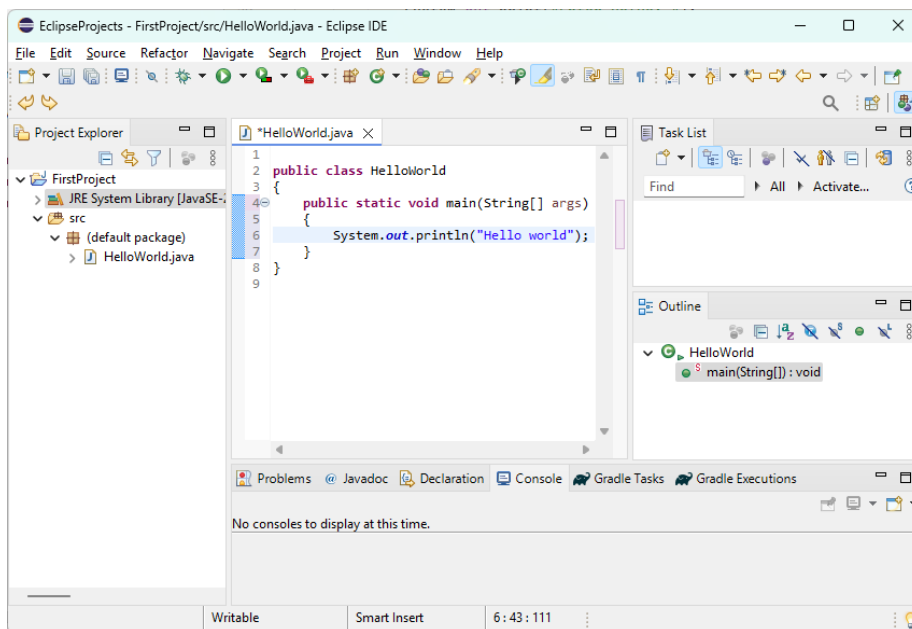


Figure 8

2.3 Compiling and running applications

There are various ways in which you can run an application:

1. Choose **Run > Run** from the top menu.
2. Use a keyboard shortcut: **control+F11** on Windows or **command+Fn +F11** on a Mac.
3. Press the **Run** icon (🟢)

The output will appear in the bottom window as shown in Figure 9.

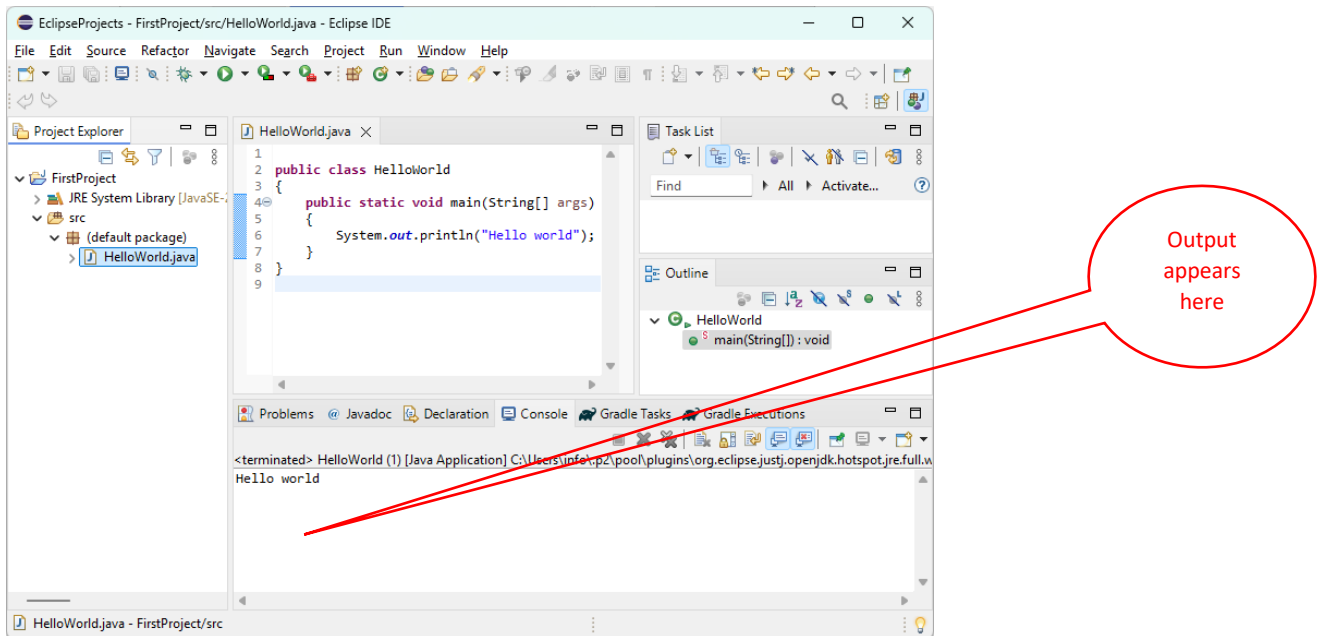


Figure 9

2.4 Correcting syntax errors

Any syntax errors that you make when typing your code will be highlighted as shown in Figure 10.

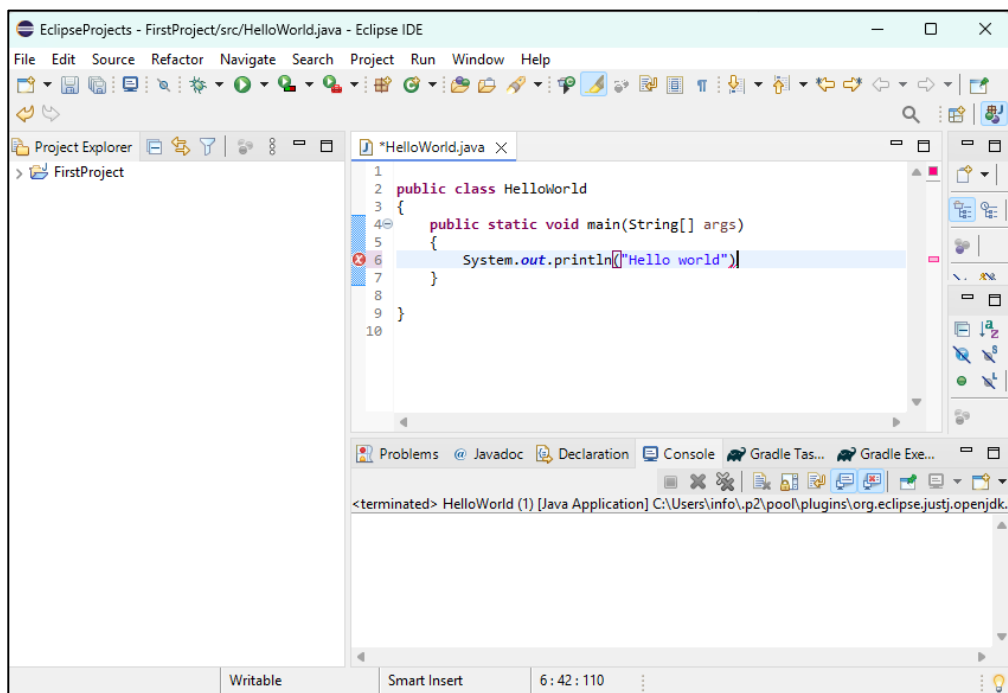


Figure 10

Hovering over the 'x' will reveal the nature of the particular error (Figure 11).

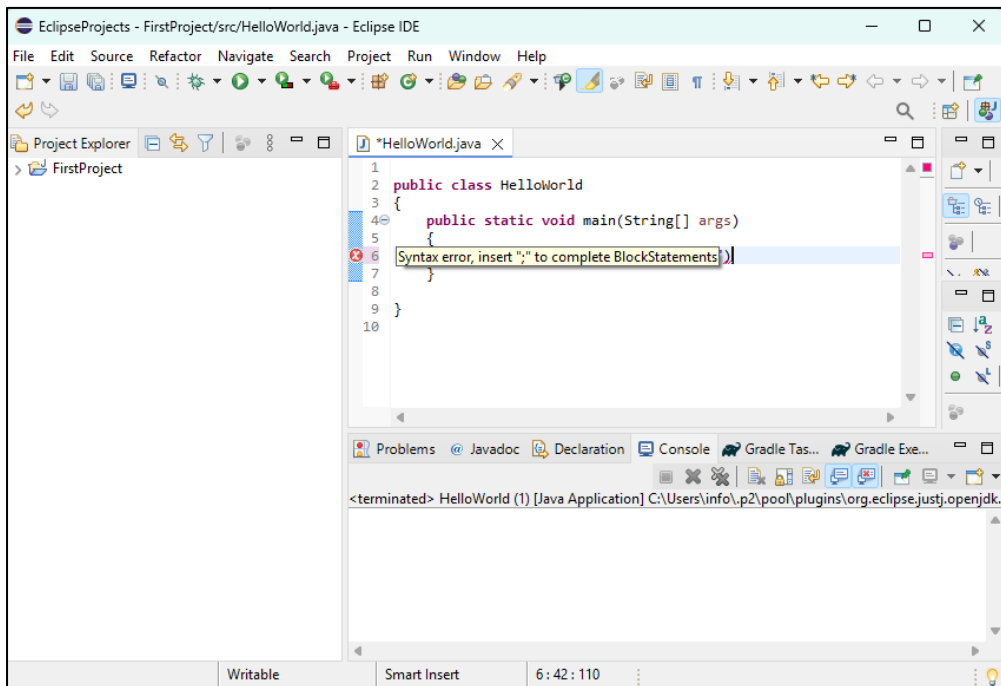


Figure 11

If you have existing source code files (.java) you can simply copy them into the *src* folder while still in the Eclipse environment.

3. Creating JavaFX projects

Download the latest JavaFX SDK (which comes in the form of a zipped file) from the link below.

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

Extract the files to a location of your choice. Here we will assume they are in a folder called:

C:\JavaFX

The above assumes the use of a Windows operating system. If you are using another operating system such as Linux or MacOS, note that such systems use the forward slash instead of the backslash, and use the **mount point (/)** instead of a drive letter.

Within the folder where you have unzipped your files, there will be another folder called *lib* which contains the required .jar files. There will also be a folder called *bin* and a folder called *legal*.

The above step only has to be done once. However the following steps have to be followed for each new project.

Start a new project - do not choose a JavaFX project, just choose a regular Java project as before.

Highlight the project (we have called it *FirstJavaFXProject*) and right click. You will see the screen shown in Figure 12.

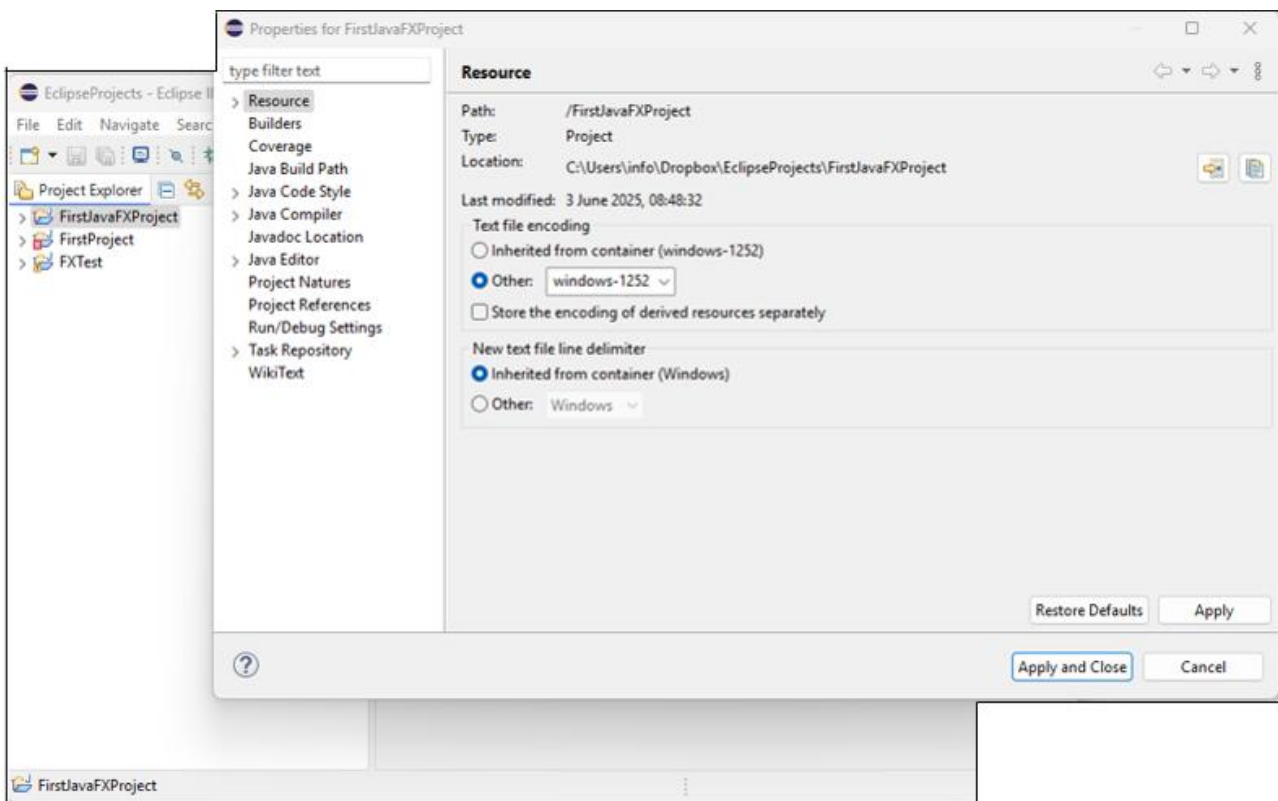


Figure 12

Click on *Java Build Path* and open the *Libraries* tab. Then click on *Add External JARs...* as shown in Figure 13.

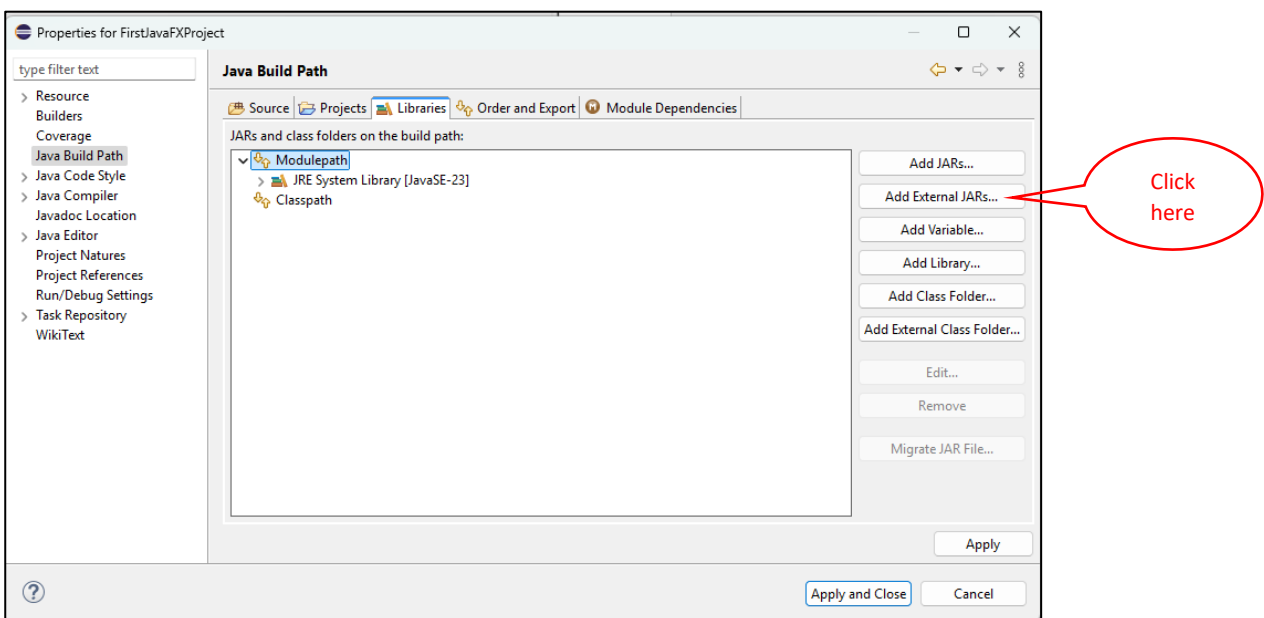


Figure 13

You can now add the JavaFX files that you downloaded. Browse to the location of your JavaFX\lib folder that you created earlier (in our case C:\javafx\lib) and select all the files from that folder as shown in Figure 14

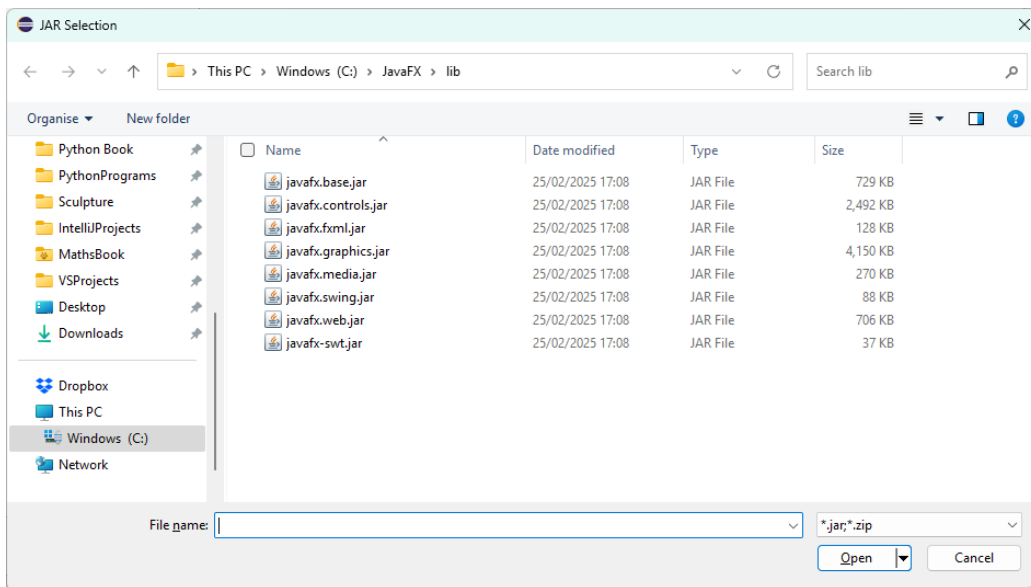


Figure 14

Click *Open*. As shown in Figure 15, the files will be added to the build path.

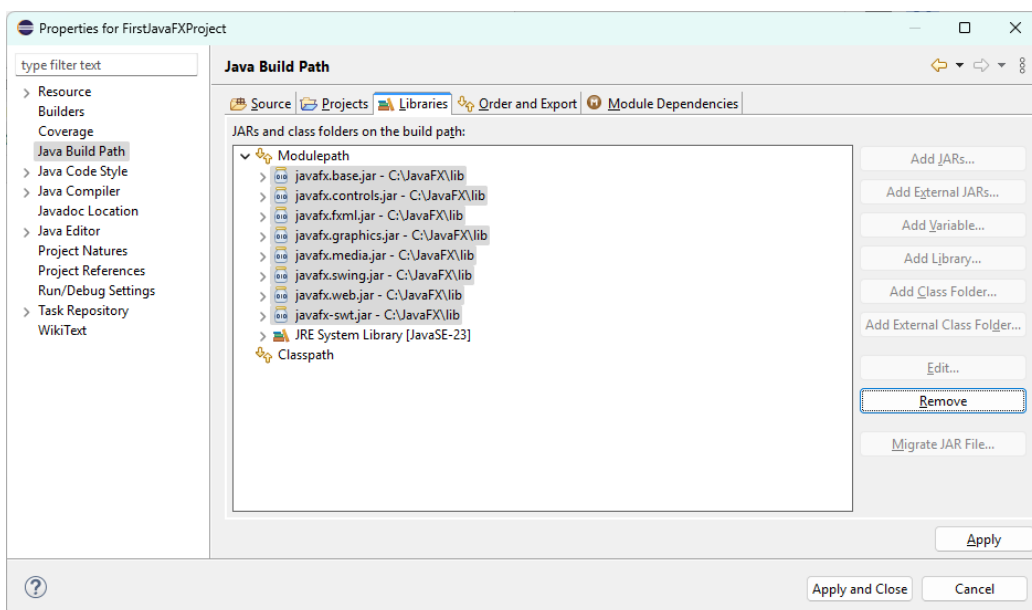


Figure 15

Click *Apply and Close*.

You can now add your JavaFX application classes to your project, and you should find that they compile successfully.

However if you try running the program you will get a runtime error. This happens because Eclipse does not add all the modules to the module path.

To fix this choose **Run > Run Configurations...** from the top menu

Under *Java Application* double click the main class, and then select the *Arguments* tab (Figure 16).

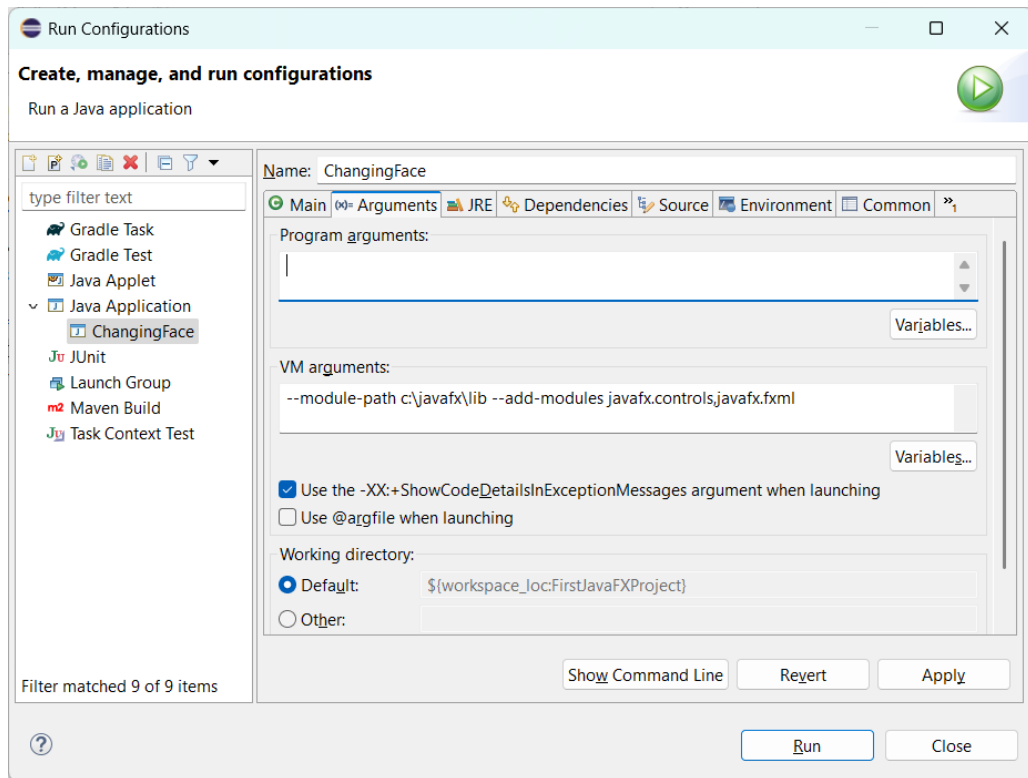


Figure 16

In this box enter the following, if necessary replacing the directory (in red) with your own location for the JavaFX files - don't forget that the path name should be enclosed in quotes if it contains spaces.

```
--module-path c:\javafx\lib --add-modules javafx.controls,javafx.fxml
```

You can choose **Run** from here, or choose close and run the program as normal. Your application should now run successfully.

4. Configuring Eclipse so that the braces line up

In order to be consistent with the examples in the test, you should configure Eclipse so that the braces line up. Choose *Window* from the top menu, then *preferences*. Choose *Java, Code Style, Formatter* (Figure 17).

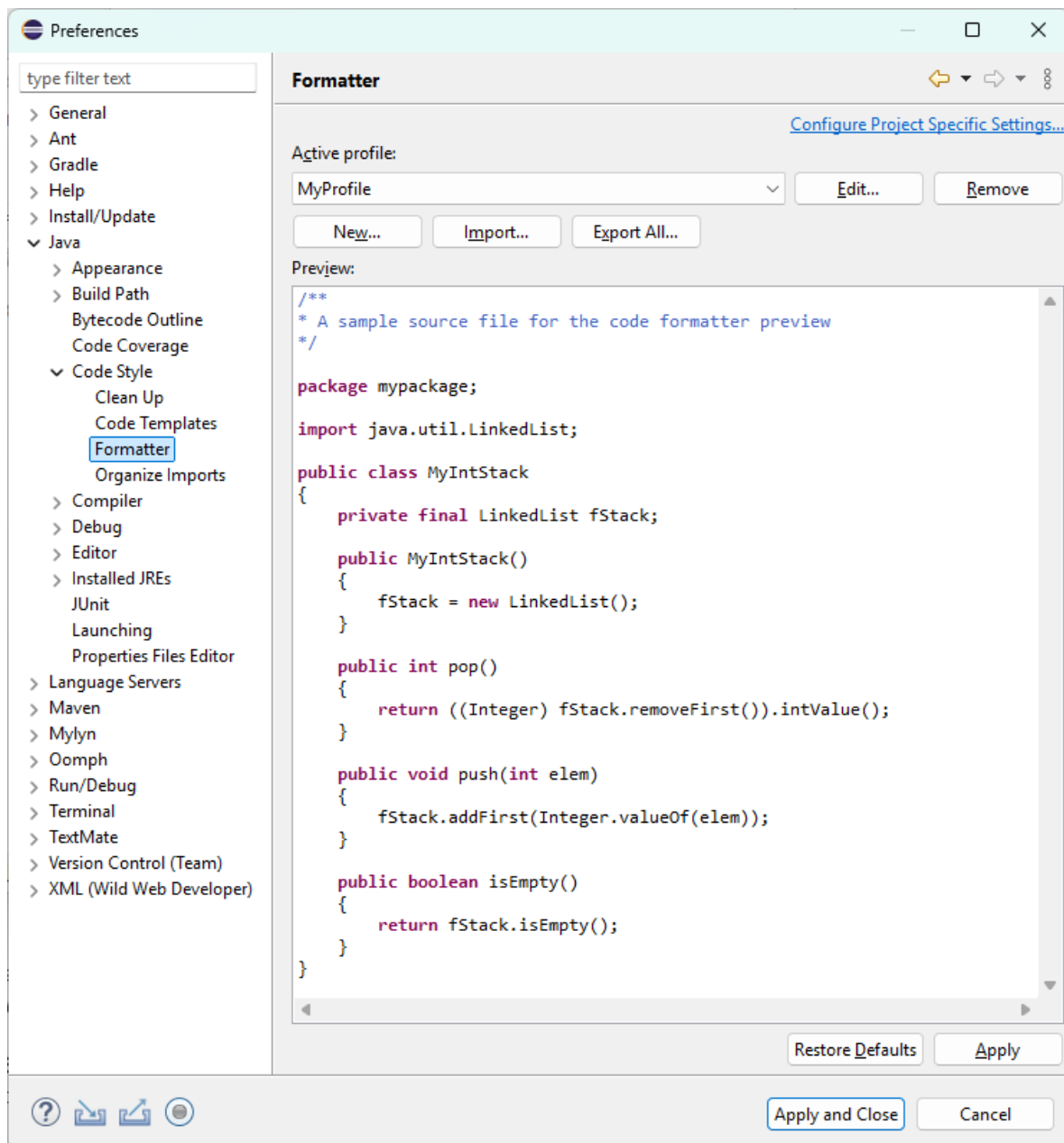


Figure 17

Click on *Edit*. The dialogue shown in Figure 18 will appear.

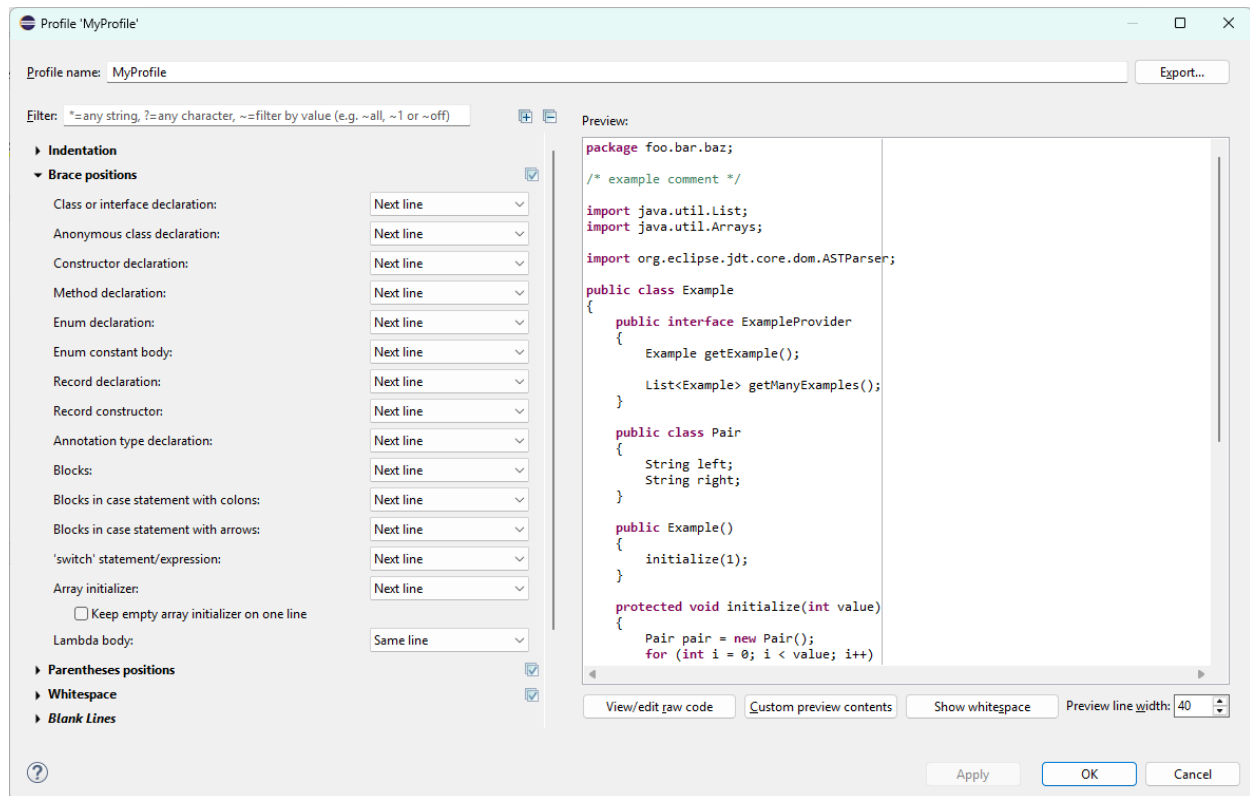


Figure 18

Under *Brace Positions* on the left, select *Next line* for each option under *Brace positions*. Finish by pressing OK.