Exercise 1 – Hello World

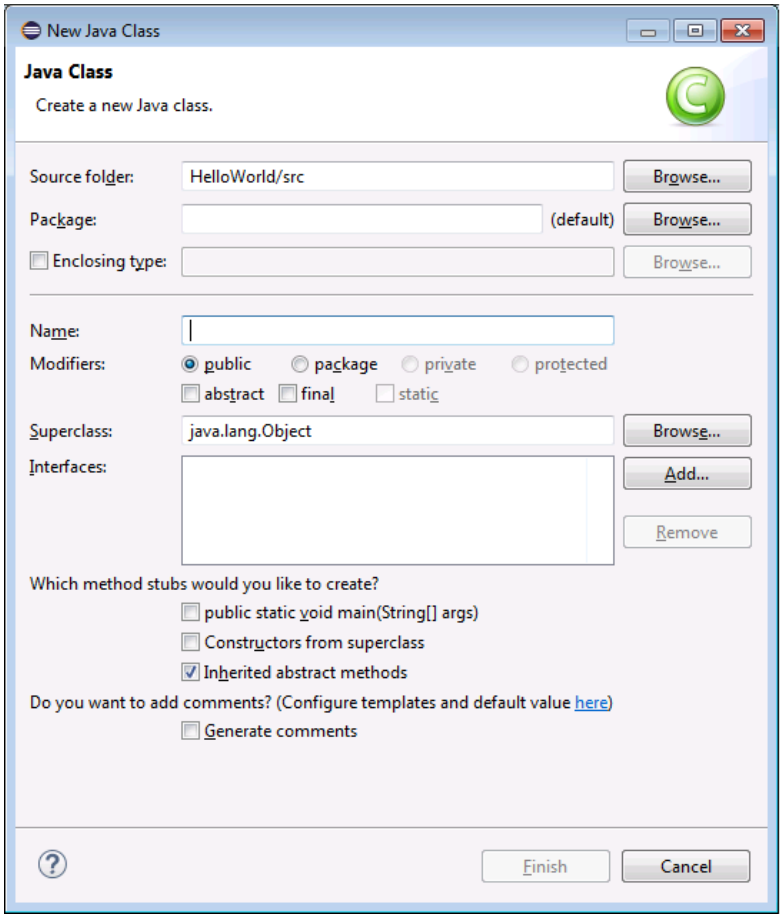
Objective

Write a hello world application. Run it using both Eclipse and on the command line.

Overview

Use the project we created in Exercise 0 for this!

We want to create a new Java file to run our Hello World application. To do this, right click on the src folder on the left hand side of the screen and go to new -> class



In the new Java class screen we have several options. The source folder is set to where we right clicked. The package we're going to leave blank for now. This isn't recommended as Eclipse will tell you, but for something this simple we don't want to add extra levels of complexity.

The name you can set to anything; I used "HelloWorld". Convention is to have classes start with capital letters and capitalise each word in the name.

We want to leave the modifiers as they are for now. This needs to be a public class so we can run it. The only bit we want to change is in "method stubs" we want to add a public static void main method to the class automatically. We can write it ourselves later, but this will save some time.

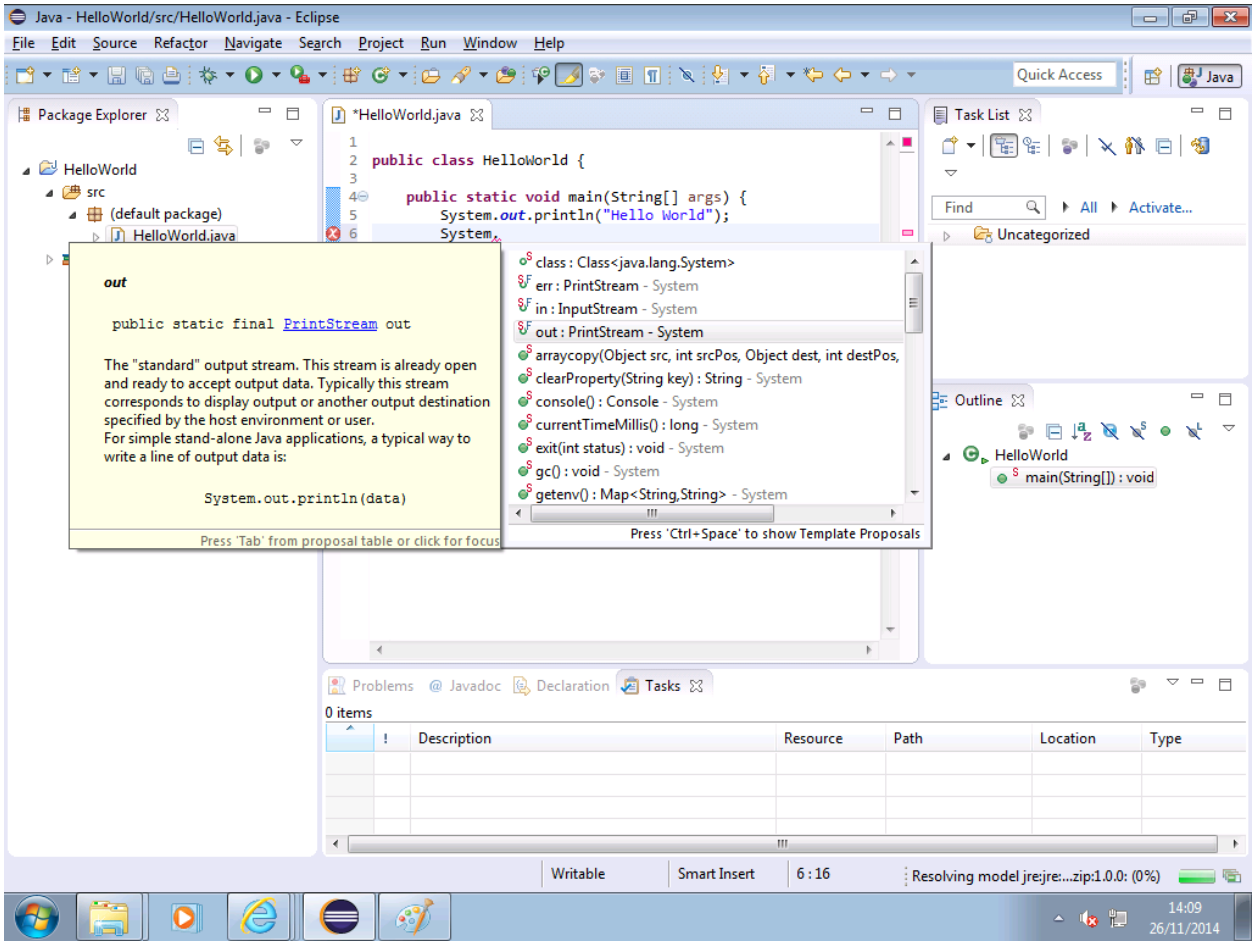
Select that option and click Finish. We now have our first class.

```
1
2 public class main {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6     }
7
8 }
9
```

The //TODO line is automatically generated for us, and lets us know that we still have stuff 'to do' in this class. If you go to the window menu, then show view, and select tasks then a tab will appear at the bottom with all the tasks to do. This is automatically generated from the TODO comments in the code.

Delete the comment and start writing `"System.out.println("Hello World");"` As you type you will see that a code recommender option pops up on the screen. We like having recommendations – it's why we're using an IDE! So press enter to say yes to this option.

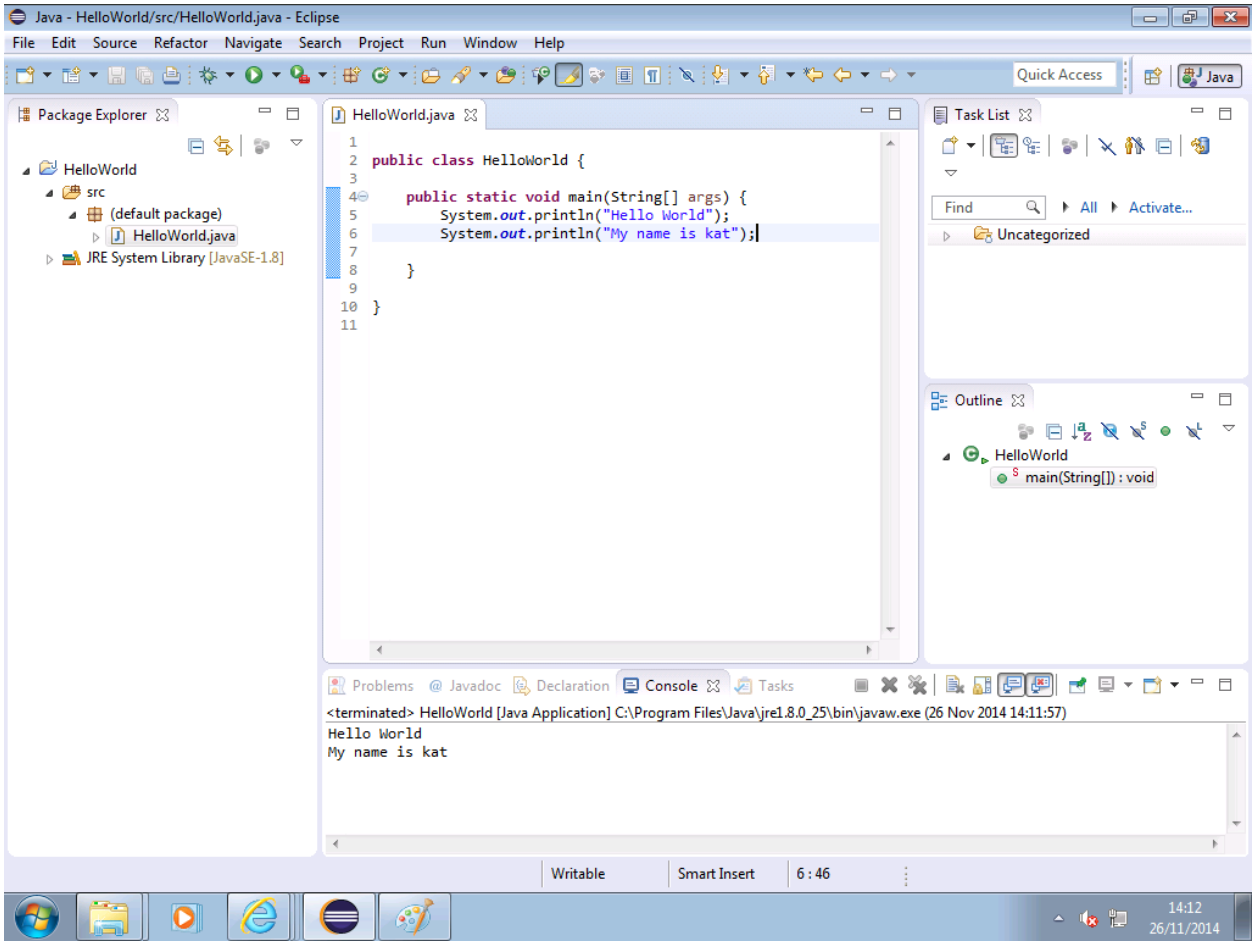
Now if you start on a new line and just type `"System."` and wait for a moment, a list will populate itself with all the options you can use on the System object. Find "out" and have a read of the text that is available for it. This is Eclipse presenting you with the documentation for this method. Don't worry if you don't understand what it is saying at the moment!



Write a print statement to print out your name, and make sure you include the semi-colon at the end of the line. Save your file with Ctrl+S. Then we are ready to run the program.

If there are any red lines then we have a syntax error. Hover over the red mark on the right hand side of the edit box, where the scroll bar is to see what the message is. If there are none then we can either run the program by going to the run menu and selecting run, pressing Ctrl+F11 or by pressing the green 'play' button on the top of the screen.

The output for your program should appear at the bottom of the screen.



Congratulations! You've just written your very first Java program!

If you have time:

Try compiling and running the program from the command line. To do this you will need to navigate to your project directory. Eclipse puts all the class files by default in a "bin" directory.