TP01: First ”Hello World!” program

Dominique Blouin, Télécom Paris, Institut Polytechnique de Paris
dominique.blouin@telecom-paris.fr

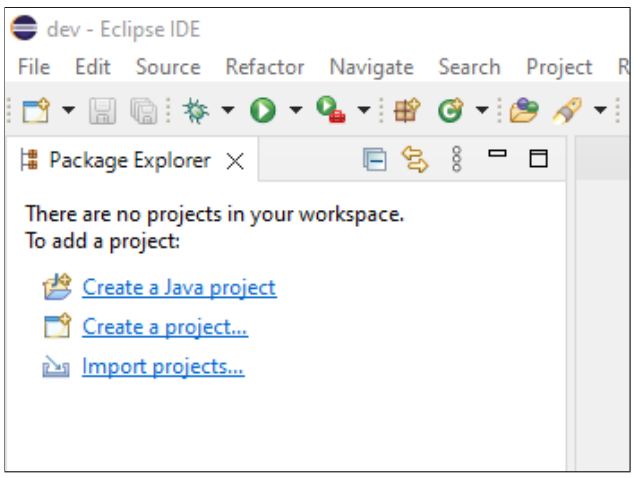
During this lab session, we will introduce the Integrated Development Environment (IDE) named Eclipse. We will learn to create and execute a simple Java program of the type `"Hello World"` using Eclipse. Then, we will explore how to achieve the same result using the terminal to understand how to compile and execute a Java program without relying on an IDE.

1 First contact with Eclipse

Launch Eclipse from the menus of your Linux or Windows environment.

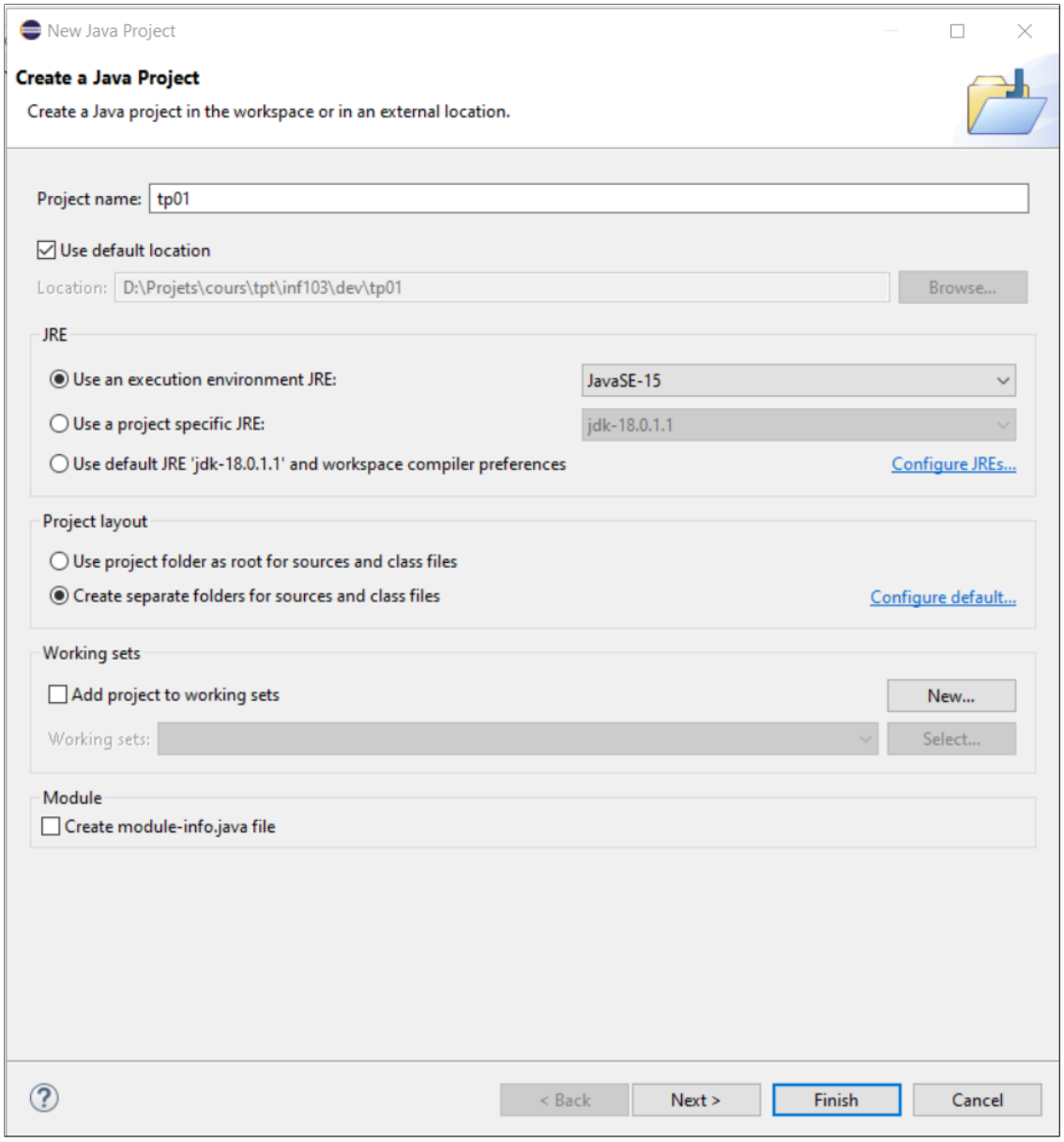
Explorer and project

1. In the main Eclipse window that appears, locate a panel named `Package Explorer` as shown below:



2. Inside the `Package Explorer`, click on `Create a Java project`.

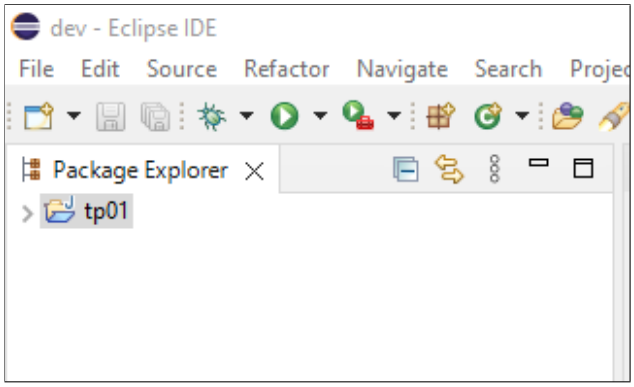
3. A new window appears (see below) asking you to enter the project name.



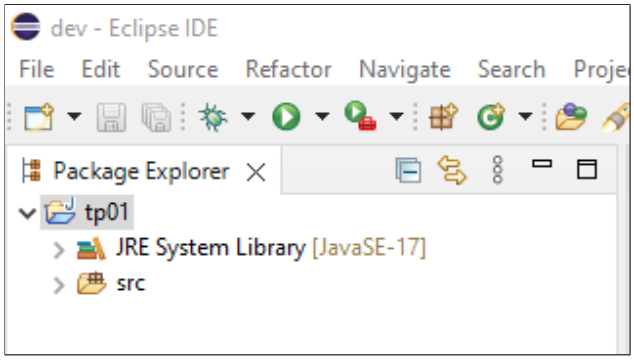
4. In the `Project name` field, enter `tp01`.
5. At the bottom of the window, locate the `Create module-info.java file` option (see `Note 1`). Ensure that this option is unchecked. If it is checked, uncheck it.
6. Leave all other options in the window unchanged.
7. Click the `Finish` button.

Note 1: The concept of *modules*, which we will not use for this course, is designed to improve the modularity of Java applications, particularly those of very large size.

8. The created project (`tp01`) appears in the `Package Explorer` panel (see below).

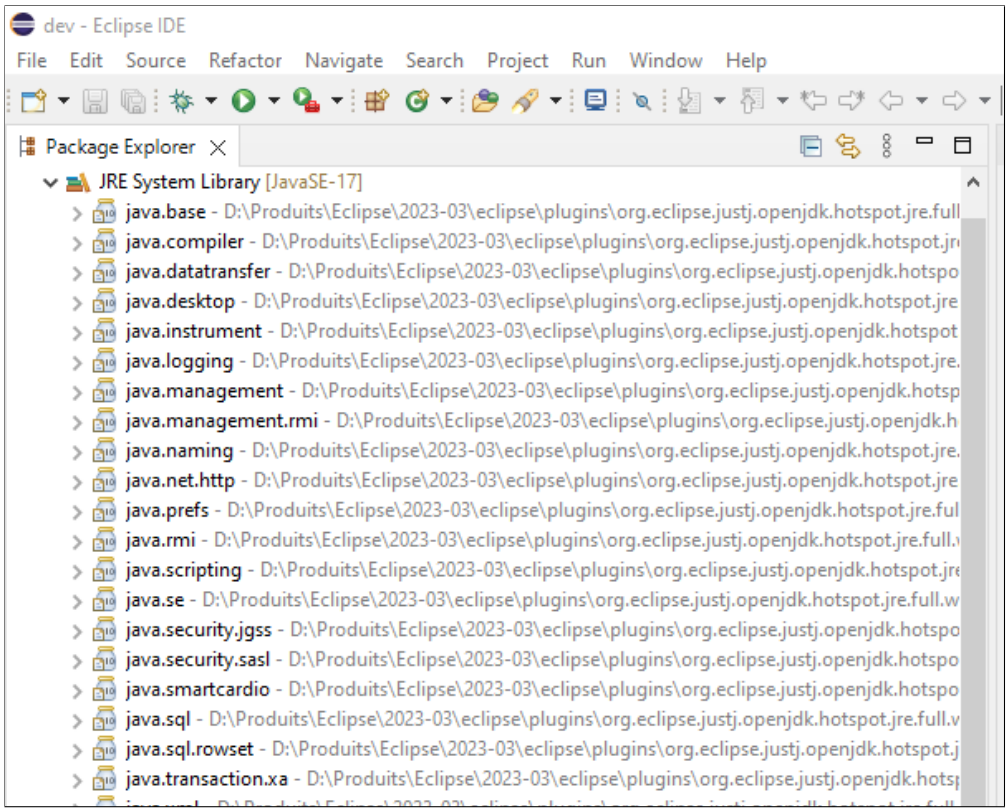


9. Click on the small triangle (`>`) to the left of the project name `tp01` . The triangle turns downward (`V`) and the project contents are revealed (see below).



10. The `src` folder (abbreviation for **source**) appears, intended to contain the Java source code of your programs, along with the `JRE System Library` folder, which contains all the libraries that Java makes available to you. By clicking on the triangle (`>`) to the left of the `JRE System Library` folder, you can see the list of these libraries (see below).

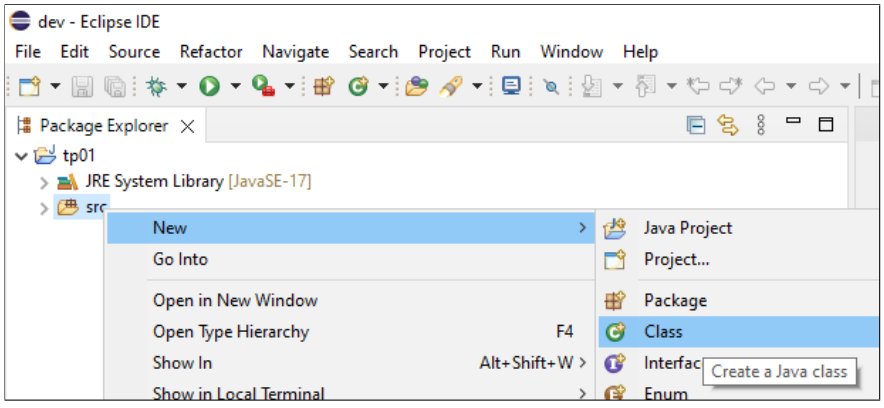
11. These libraries are sets of predefined classes made available by Java. We will study some of them during the course. By clicking again on the triangle (`V`) to the left of the `JRE System Library` folder, you can close this folder.



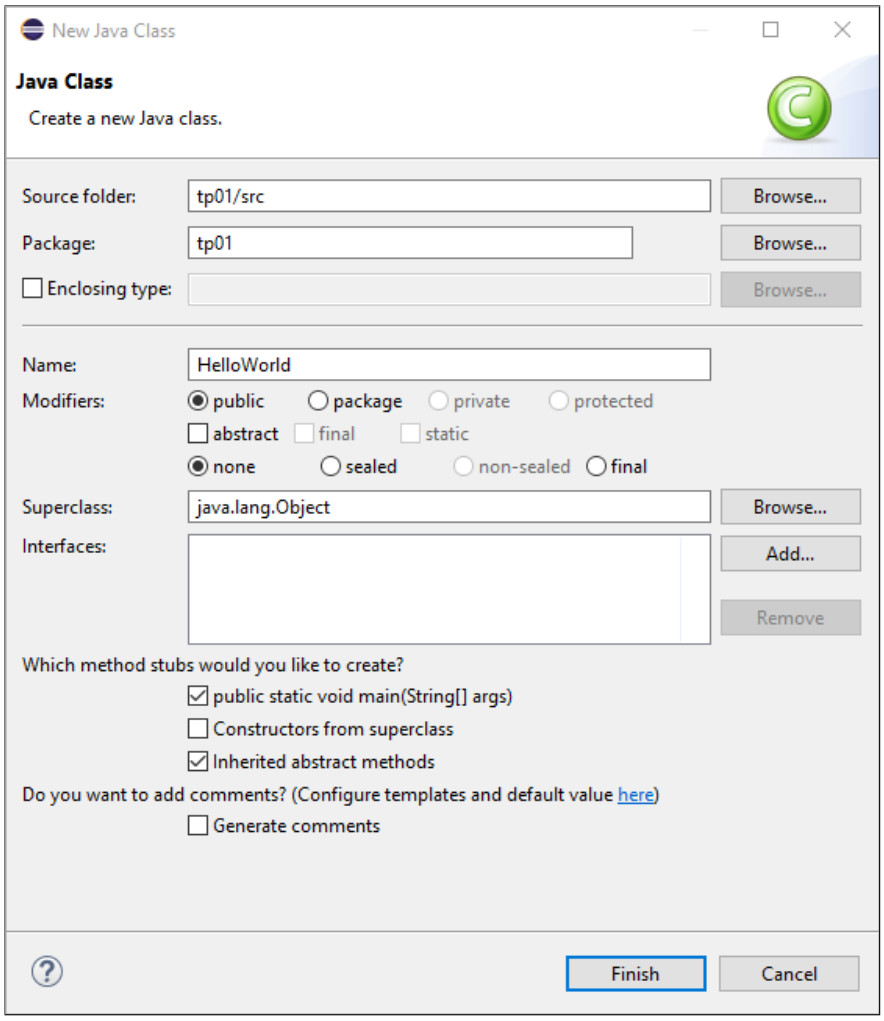
Class

We are going to create a class.

1. To do this, right-click on the `src` folder. In the menu that appears, hover the mouse over `New`. A submenu will appear, hover the mouse over `Class` and click (see below).



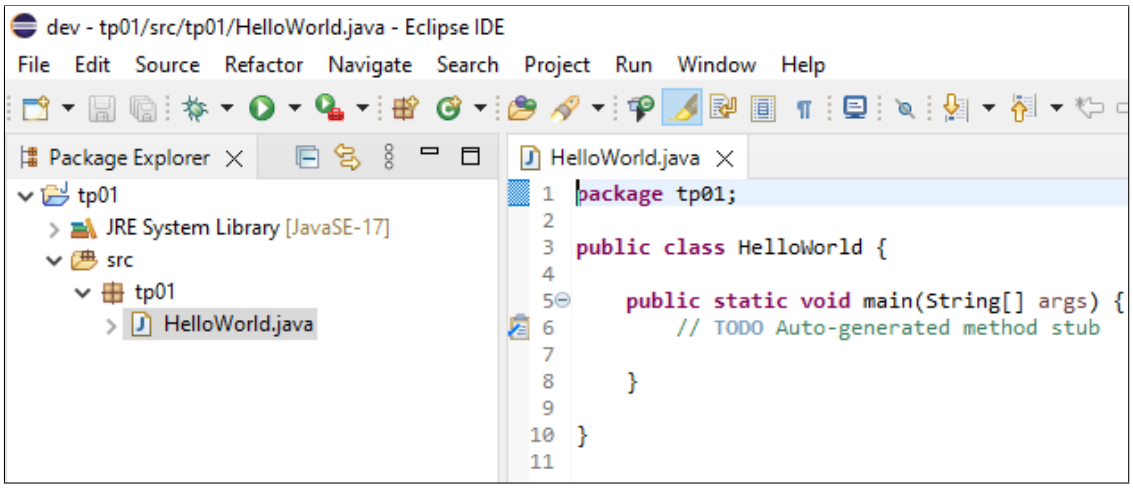
2. A new window like the one in the screenshot below appears.



3. In the `Name` field, write `HelloWorld`, then select the checkbox shown in front of `public static void main(String[] args)`.

4. Click on the `Finish` button.

In the `Package Explorer` view, the class file named `HelloWorld.java` appears. In the central panel to the right of the `Package Explorer`, the Java code for this class is displayed. This panel also serves to edit the class code.



In this class, everything between the character strings `/**` and `*/` and everything between the character string `//` and the end of the line are areas dedicated to writing comments. These areas were generated by Eclipse. Also note the public keywords before the class and main method declarations. We will see the meaning of these keywords in a later lesson.

Remember: A method such as the `main` method is the one that will be executed when the program is launched.

5. This method, generated by Eclipse, is empty. In its body, write the following instruction:

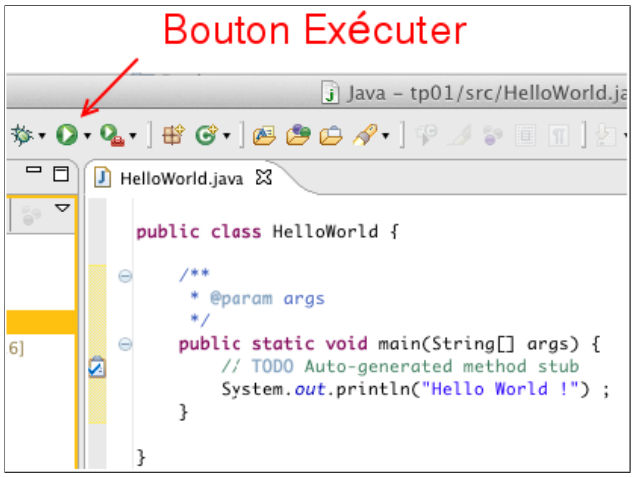
```
System.out.println("Hello World!");
```

6. You should get:

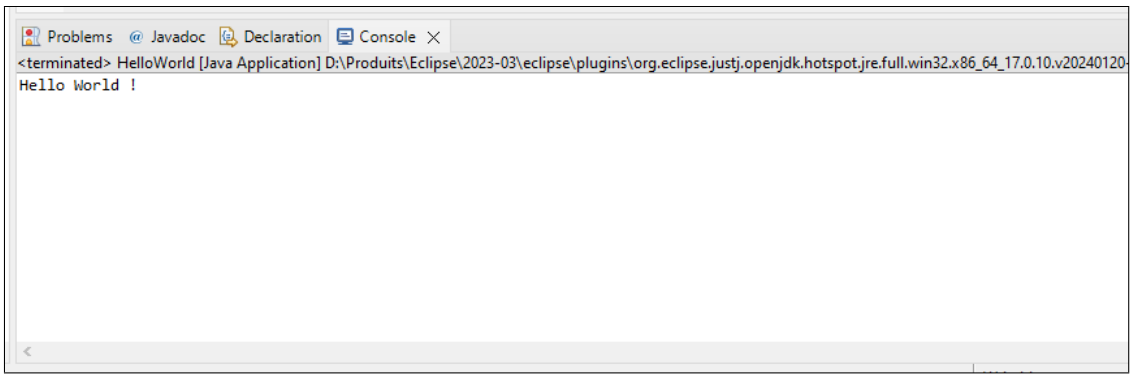
7. Save the file.

By default, Eclipse will compile the class as soon as its file is saved. The function of this instruction is to write the string `"Hello World!"` to what is called the **Console**.

8. To execute this program, click on the round green button with a white triangle inside located in the menu bar (shown below).



Eclipse will then execute the class, which will result in running the `main` method of your class, which serves as the entry point of the program. The `Console` panel will appear at the bottom of the code window. There, you will see the result of the print instruction. The program execution is immediately completed (shown below).



You have just executed your first program!

2 Compiling and executing a Java program via command line (without IDE)

As you have seen, creating a simple class and running its program is very easy with an IDE, which allows for more productive application development. But how would we do it without an IDE?

Java code is first compiled into binary code. The execution of a Java program is then done by *interpreting* this binary code using a JVM (Java Virtual Machine). The installation of this machine is carried out by deploying a JRE (Java Runtime Environment), which provides various

executables for compiling and running a Java program, as well as different utilities such as *javadoc* for automatic generation of documentation in the form of HTML pages.

These various executables are used via a terminal, a tool you used in the INF107 course.

Compile your `Hello World!` program via the command line

As mentioned above, to compile and execute a Java program, you need a JRE environment specific to the given operating system (Windows, Linux, Mac). The virtual machine provided by this environment will interpret the compiled binary code to translate it into system-specific instructions. There are various providers of JRE environments, the most well-known being Oracle and the OpenJDK project.

The executables of the JRE used to compile and execute Java programs are stored in a subdirectory named `Java/<version_name>/bin`, which is created in the operating system's program directory during the installation of the virtual machine.

1. To find out which version of Java is installed, type the following command in a terminal:

```
java -version
```

2. This version corresponds to the most recent version of the Java language supported by the virtual machine. For this exercise, please ensure that the Java version is ***higher than 1.8.***

For a given version of Java, there are two types of virtual machine installations: the JRE installation mentioned earlier and the JDK (Java Development Kit) installation. The JDK installation includes everything contained in a JRE, as well as the Java source code and the *javac* executable used for compilation.

Installing the virtual machine

If you are working on your own computer (and not a school computer) and receive a message stating that the *javac* command is not found during the following exercise, install a JDK on your computer. You can use the JDK provided by Oracle or the one from the OpenJDK project.

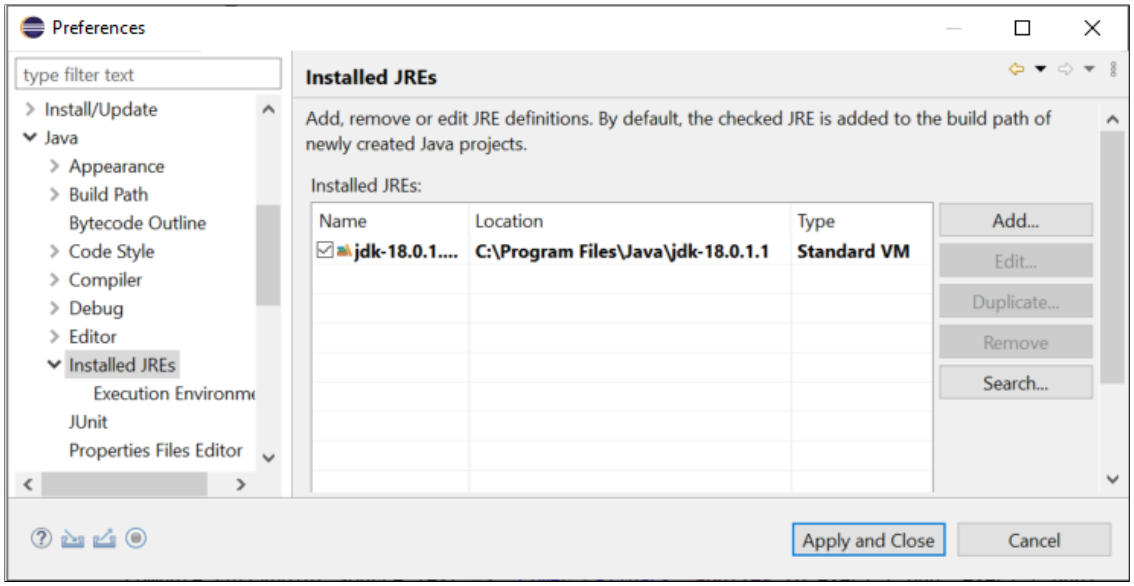
Note 2: Recent versions of Eclipse include their own JDK, which can also be used. If applicable, it will be located in the Eclipse installation directory under:

```
../eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.<version>
```

However, this may depend on how Eclipse was installed (for example, via an installer or an archive) or even on the operating system.

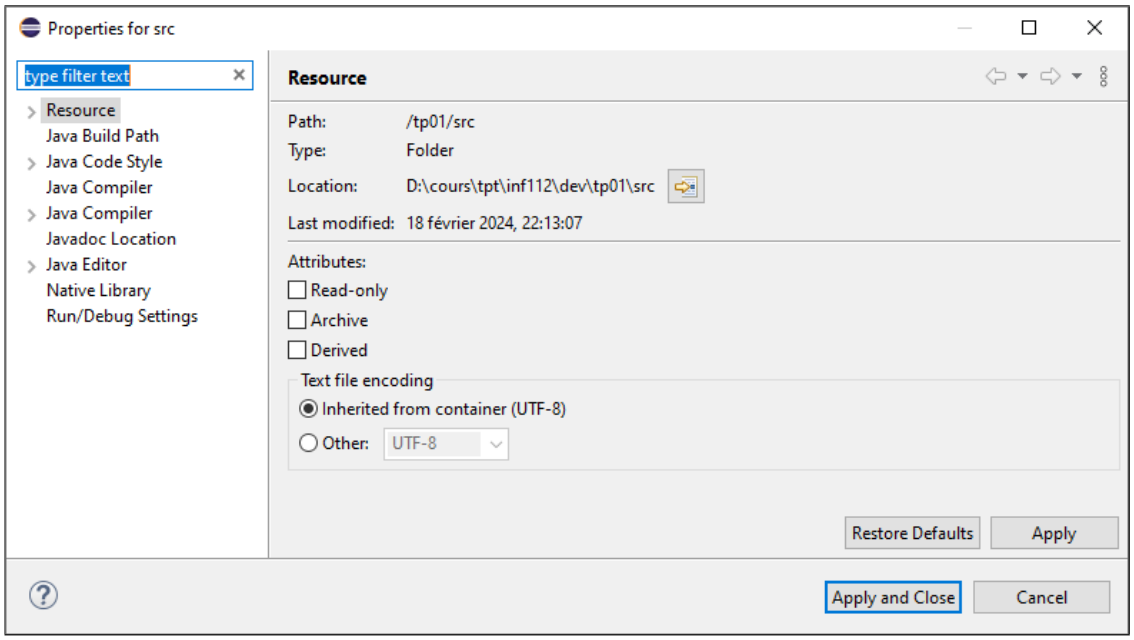
3. To find out the directory of the JVM used by Eclipse, simply click on the menu `Window` >> `Preferences`. In the dialog box that appears, select the branch

`Java/Installed JREs`, as illustrated in the screenshot below. The right side of the window will indicate the installation directory of the JVM used by Eclipse.



4. If you have not already done so, in a terminal, navigate to the `src` directory of your `"Hello World!"` program's Eclipse project.

5. To find this directory, in Eclipse, select the `src` folder in the package explorer, then right-click and select the `Properties` menu. In the window that appears, select the `Resource` branch. The `src` folder's directory is displayed on the right side of the window as indicated by the `Location` label (Shown bellow). A button also allows you to open a file browser directly positioned in the directory.



6. Then type the command

```
javac tp01/HelloWorld.java
```

Examine the contents of the `src/tp01` directory containing the Java file. You will see that running the `javac` program has produced another file named `HelloWorld.class` from the `HelloWorld.java` file. This file contains the binary code generated by compiling the Java code.

It is possible to specify a directory (different from `src`) to `javac` for storing compiled files. For example, during compilation, Eclipse will store the compiled files in a different directory named `bin` (for *binary*) located at the same level as `src` in the project directory.

Execute your `"Hello World!"` program

The `HelloWorld.class` file containing the compiled code can now be executed by the virtual machine.

7. To do this, run the *java* executable with the class name as a parameter. While still being in the `src` directory, type the command:

```
java tp01/HelloWorld
```

8. You should see the following string displayed in the terminal.

```
Hello World!
```

9. It is also possible to perform these two operations (that is, Nr. 6 and 7) in a single command with the Java executable. First, delete the compiled `HelloWorld.class` file and verify that it has been deleted. Then, type the command (see also `Note 3`):

```
java tp01/HelloWorld.java
```

10. You will see the string `Hello World!` displayed in the terminal.

```
Hello World!
```

`Note 3:` Executing the program in this way, note that the compiled file was not saved to disk. It was simply created and then executed without being saved. Thus, it is preferable to first compile a program using `javac` (which is what Eclipse does), as the program can then be executed as many times as desired without recompiling.

Note 4: These commands work for a class contained in a `package` [1] named `tp01` (that is, its `.java` file is directly in the `tp01` subdirectory of `src`). If you have declared your class in another package (for example, a package named `test`), its `.java` file will be contained in a subdirectory named `test` of `src`. To execute this class, you must position yourself in the `src` directory and prefix the class name with its package name `test` instead of `tp01`.

If you try to execute the class in the test subdirectory, you will see an error message such as

```
Error: Could not find or load main class HelloWorld
```

You must therefore verify that you are launching the commands from the `src` directory.

These simple commands show us how Eclipse (or any other IDE) compiles and executes a Java program for us. In Eclipse, compilation is executed each time a Java file is saved. Thus, when a project contains multiple Java files, Eclipse will only recompile the modified files and those that depend on them, which is called *incremental compilation*, minimizing the compilation time.

[1] We will cover the concept of *packages* in a later course.