

## Contents

Modulazation of the JDK under Project Jigsaw .....	2
Create Project Single Module with Eclipse.....	3
Multi Class in same Module .....	10
One Project has Multi Module .....	11
Summary .....	12

## Modulazation of the JDK under Project Jigsaw

Java Module System is a major change in Java 9 version. Java added this feature to collect java packages and code into a single unit called module. The versions earlier of java, there was no concept of module to create modular java applications, that why size of application increased and difficult to move around. Even JDK itself was too heavy in size.

That fixed situation, Java 9 restructured JDK into set of modules so that we can use only required for our project. Apart from JDK, java also allows us to create our own modules so that we can develop module based application.

The module system includes various tools and options that are given bellow:

Includes various to the java tools javac, jlink and java where can specify module paths that locates to the location of module

Modular Jar file is introduced. This Jar contains module-info.class file in its root folder.

JMOD format is introduced, which is a packaging format similar to jar except it can include native code and configuration files.

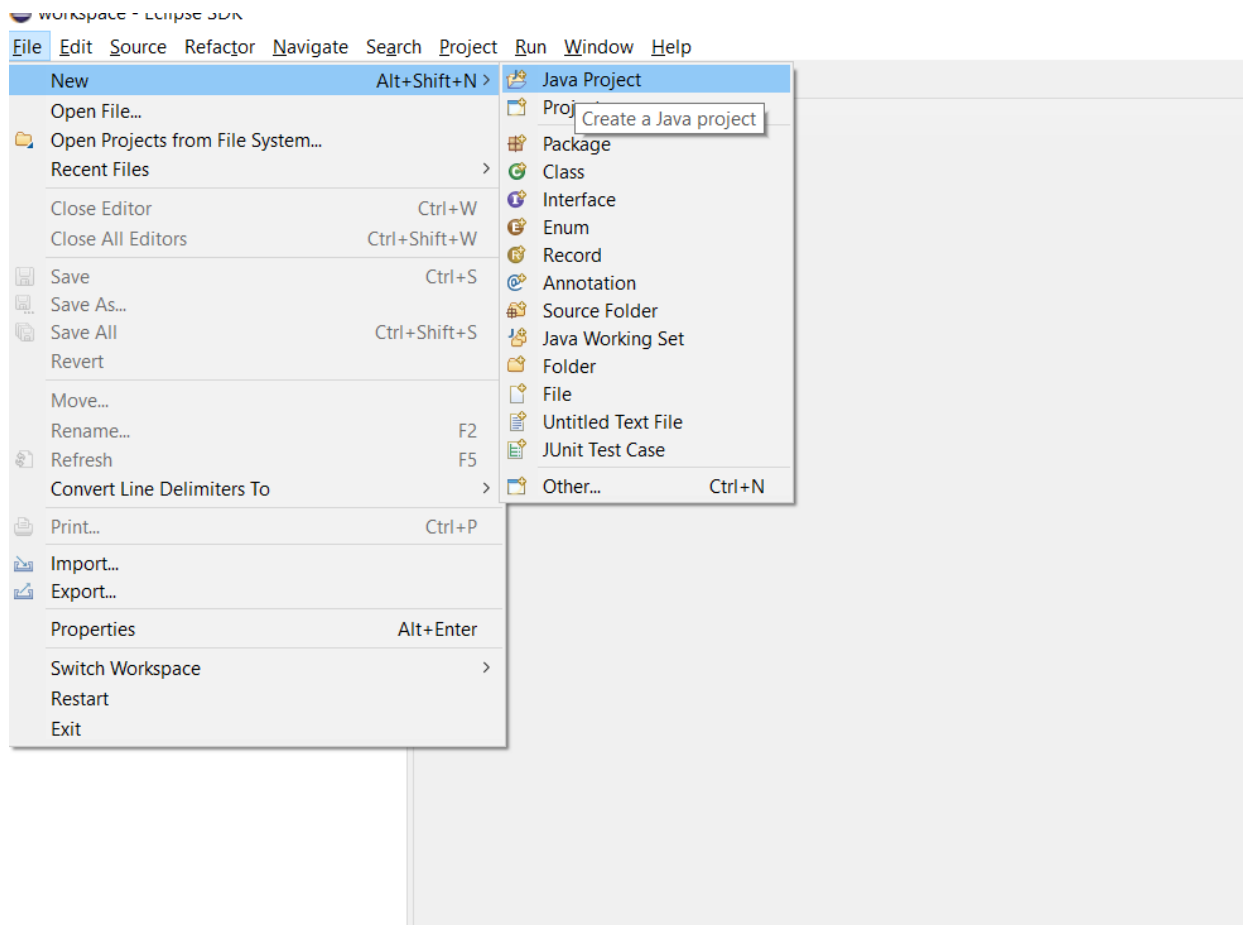
The Jdk and jre both are reconstructed to accommodate modules. It improves performance, security and maintainability.

Java defined a new URI schema for naming modules, classes and resources.

Java 9 Module System has a "java.base" module. It's known as Base Module. It's an independent module and does not depend on any other modules. By Default all other modules are dependent on "Java.base".

To use **a module**: include the **module jar** file into **modulepath** instead of the classPath. A **module jar** file added to **classPath** is **normal jar file** and module-info.class file will be ignored.

# Create Project Single Module with Eclipse



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name:

DemoProjectSe9

☐ Use default location

Location:

D:\Project\sharefullcode\Learn Java\NewFeatureSE9\Structure Project Module

Browse...

JRE

☐ Use an execution environment JRE:

JavaSE-1.8

☐ Use a project specific JRE:

jdk 9

☒ Use default JRE 'jdk 9' and workspace compiler preferences

Configure JREs...

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files

Configure default...

Working sets

☐ Add project to working sets

New...

Working sets:

Select...

The wizard will automatically configure the JRE and the project layout based on the existing source.

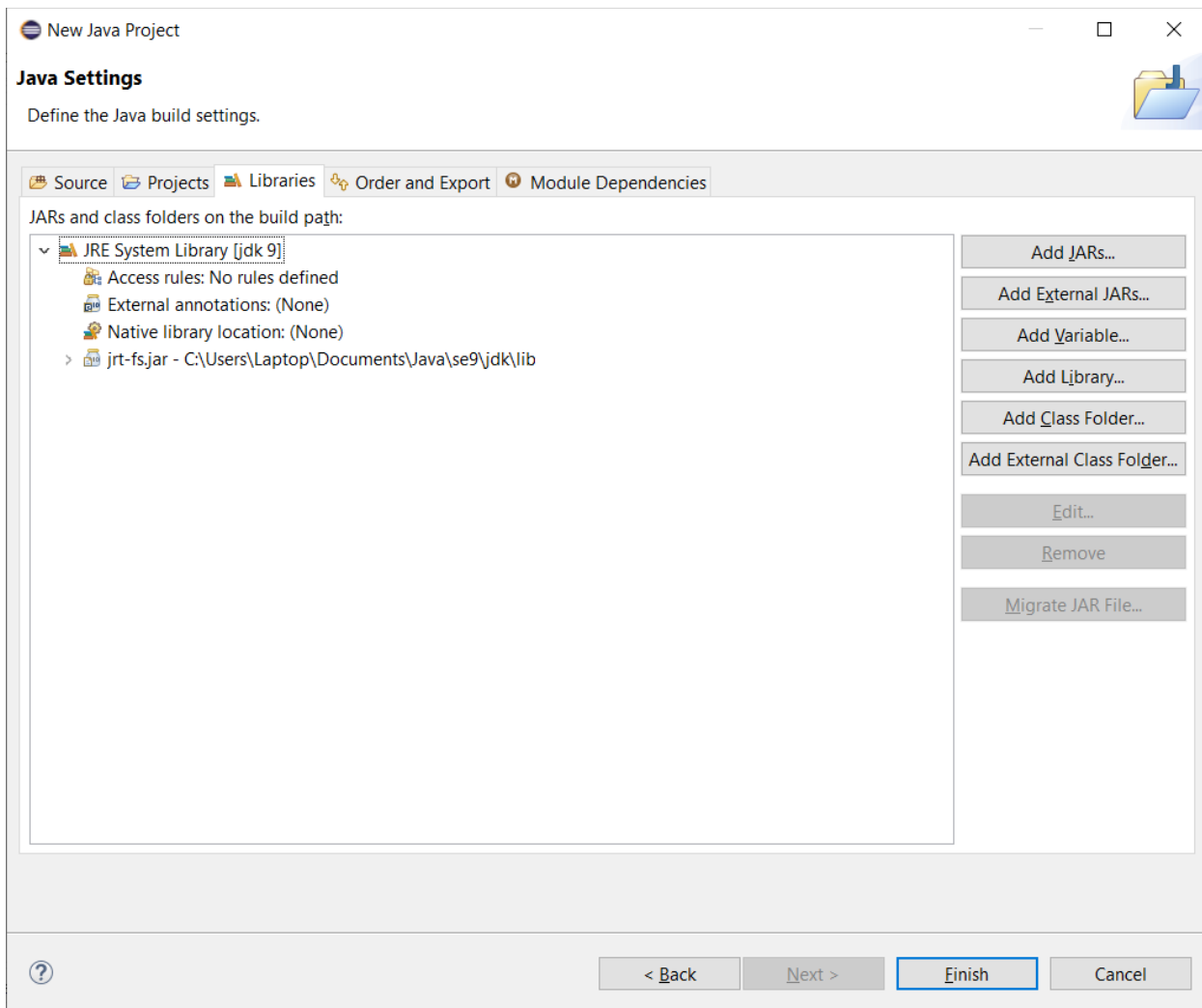
< Back

Next >

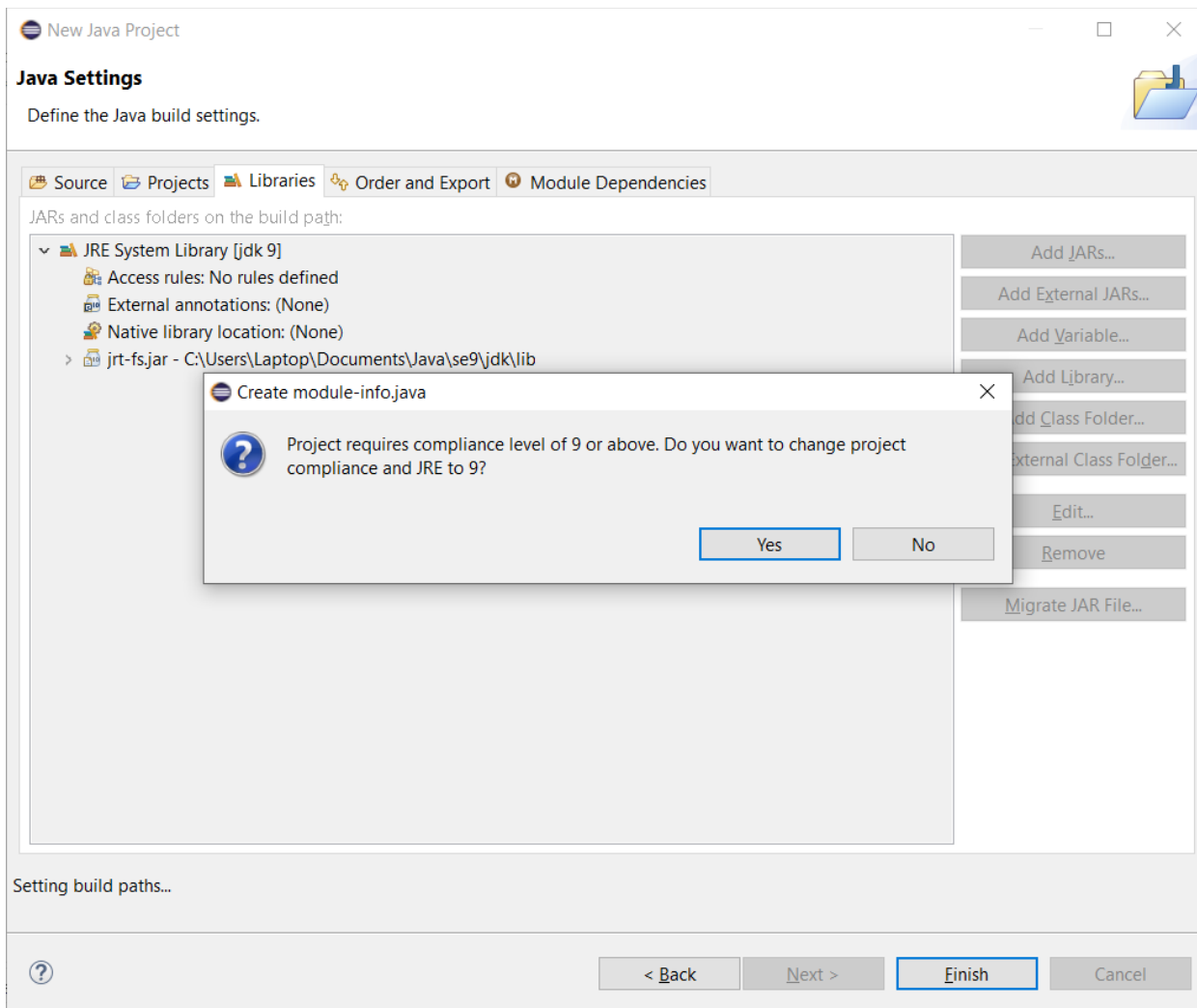
Finish

Cancel

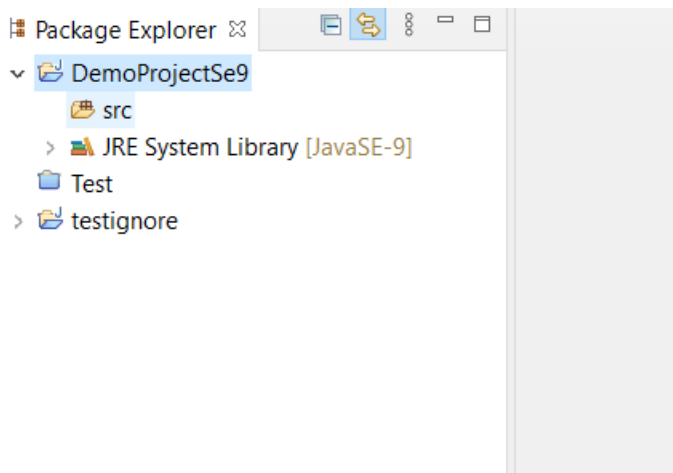
Next



Choice Finish



Choice Yes then Project created



New a class HelloWorld With package com.sharefullcode.blogspot reserves domain

New Java Class

**Java Class**

Create a new Java class.

Source folder: DemoProjectSe9/src Browse...

Package: com.sharefullcode.blogspot Browse...

☐ Enclosing type: Browse...

Name: HelloWorld

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

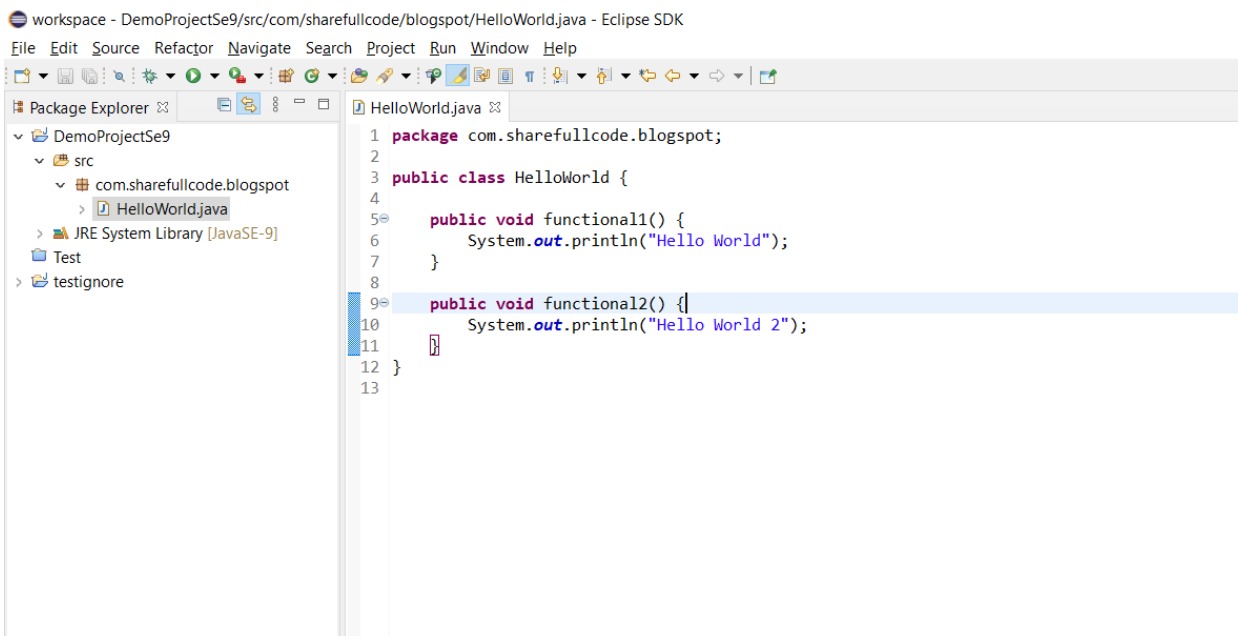
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

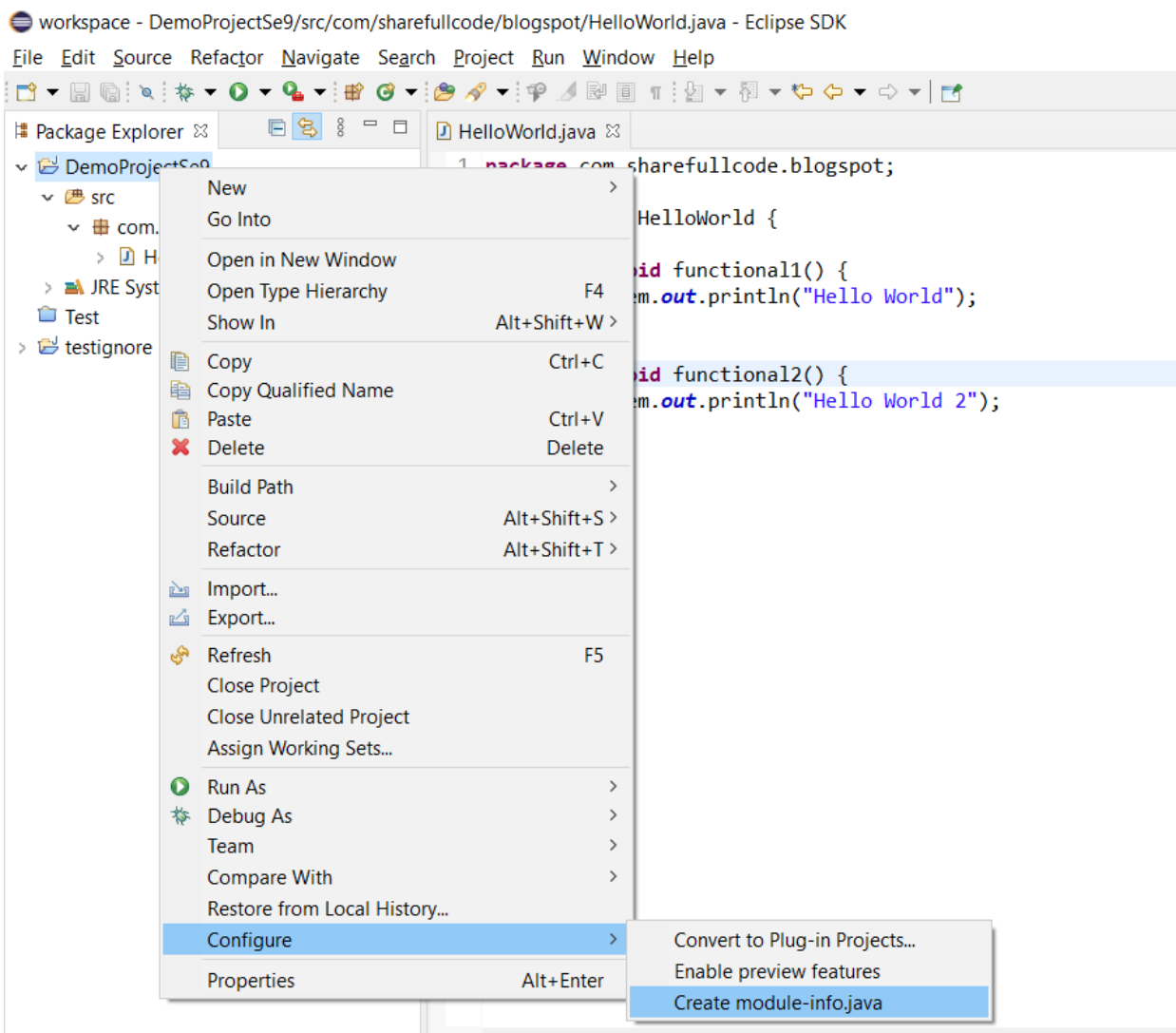
☐ Generate comments

Finish Cancel

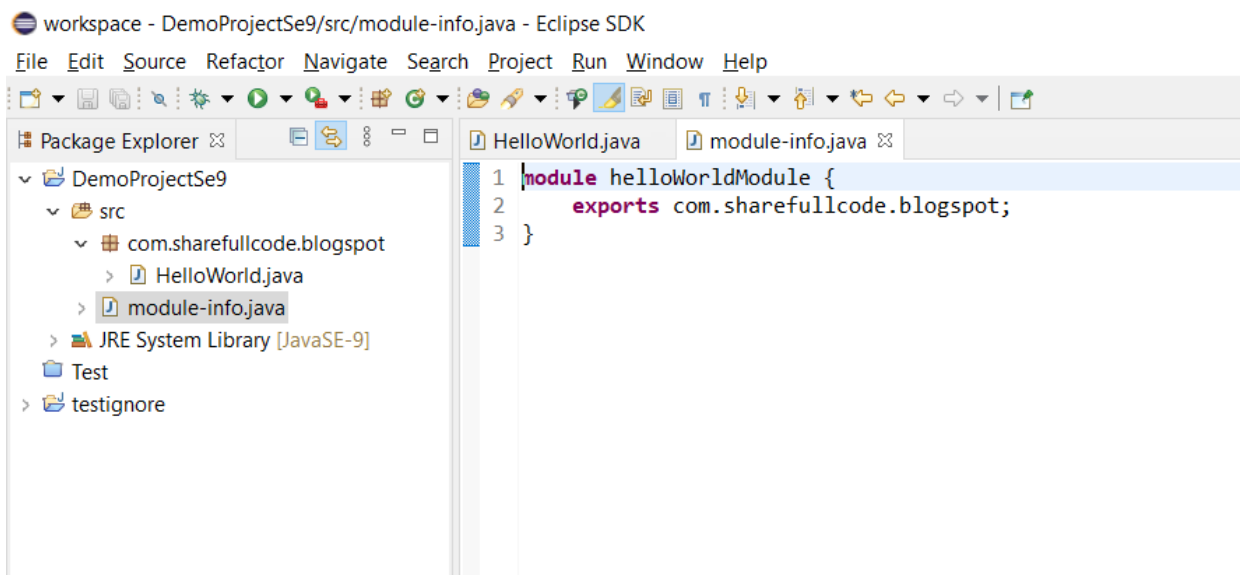
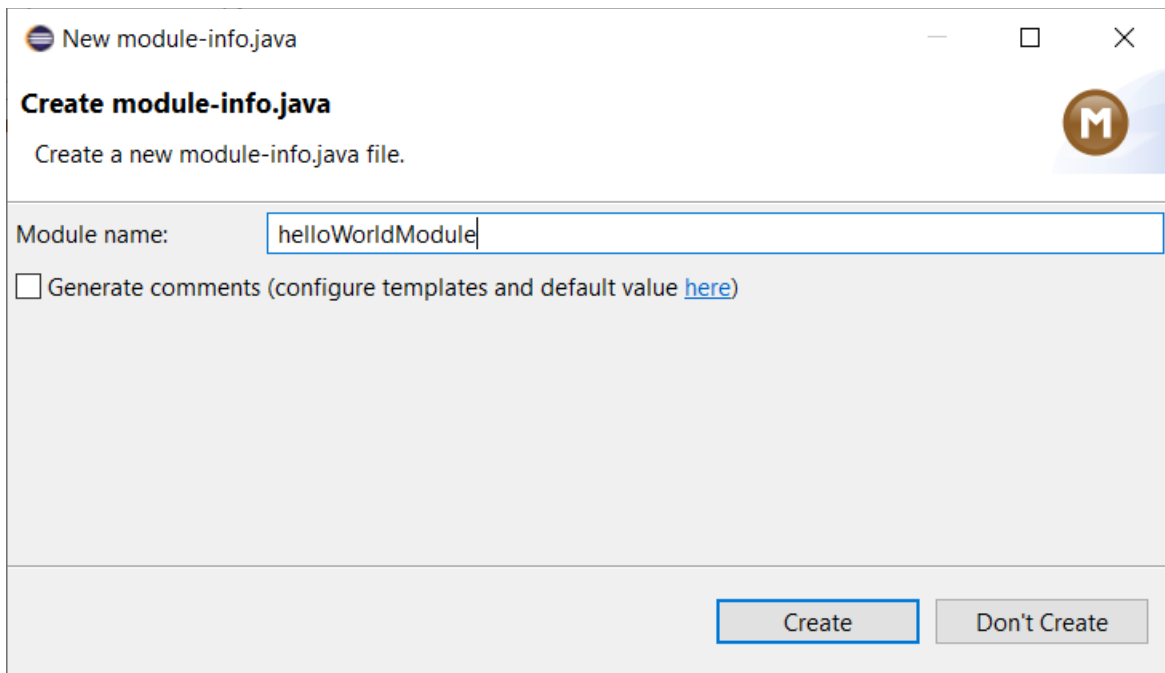
Create some functions



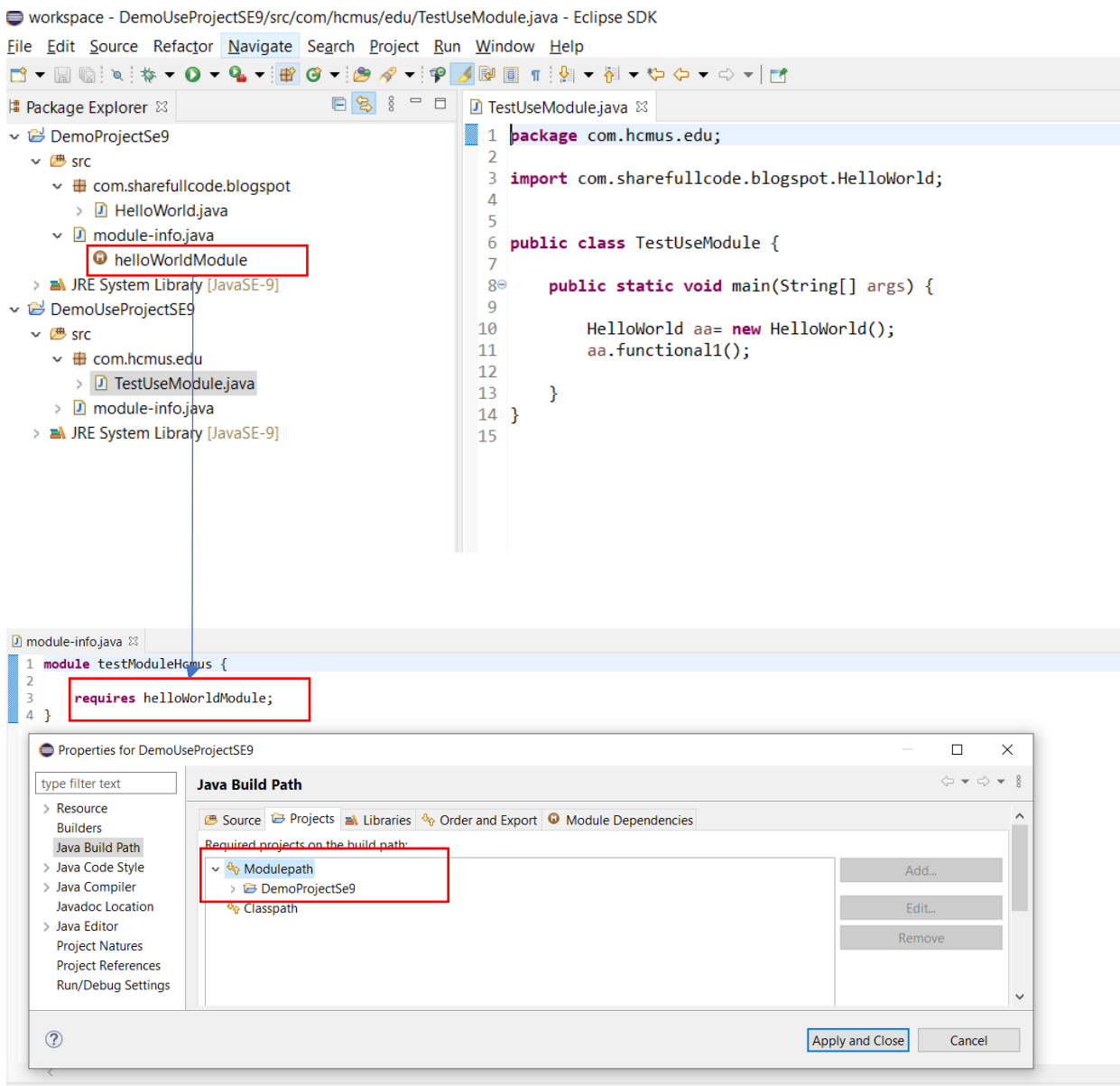
Create module-info.class from Config menu Click right mouse





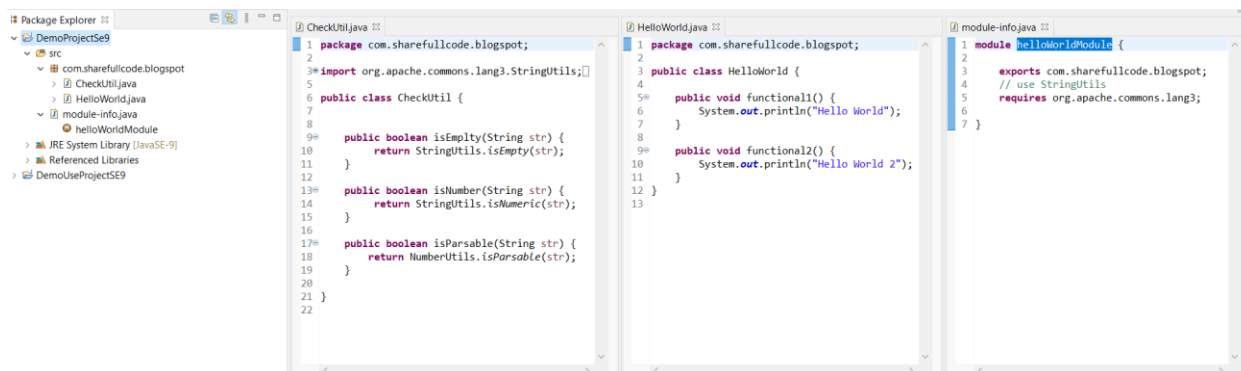


That use a module others , you must add require module in file module-infor.class and add modulePath example bellow:



Run Success.

## Multi Class in same Module



## Use it



```
TestUseModule.java
1 package com.hcmus.edu;
2
3 import com.sharefullcode.blogspot.CheckUtil;
4
5
6
7 public class TestUseModule {
8
9     public static void main(String[] args) {
10
11         // Action two class with same one module helloWorldModule.
12         HelloWorld aa= new HelloWorld();
13         aa.function1();
14
15         CheckUtil check= new CheckUtil();
16         System.out.println(check.isEmpty("sdfs"));
17
18     }
19 }
20
```

```
module-info.java
1 module testModuleHcmus {
2
3     requires helloWorldModule;
4 }
```

Problems Javadoc Declaration Console

<terminated> TestUseModule [Java Application] C:\Users\Laptop\Documents\Java\se9\jdk\bin\javaw.exe (Dec 30, 2020 1:14:00 PM - 1:14:00 PM)

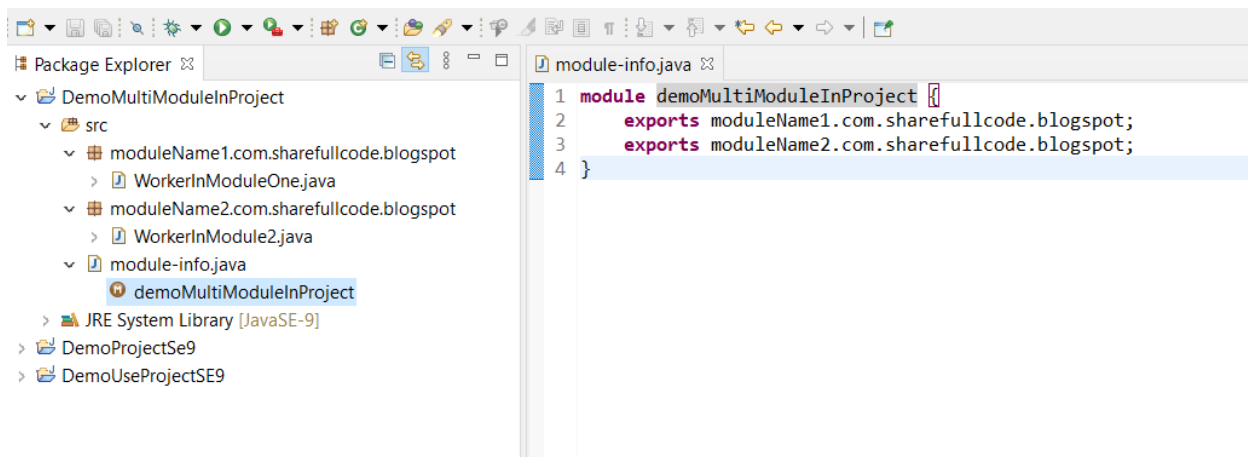
Hello World  
false

## One Project has Multi Module

You can create with command but not create on IDE because not accpect you can refer here:

<https://www.logicbig.com/tutorials/core-java-tutorial/modules/multi-module-mode.html>

Project has multi packages on a module .

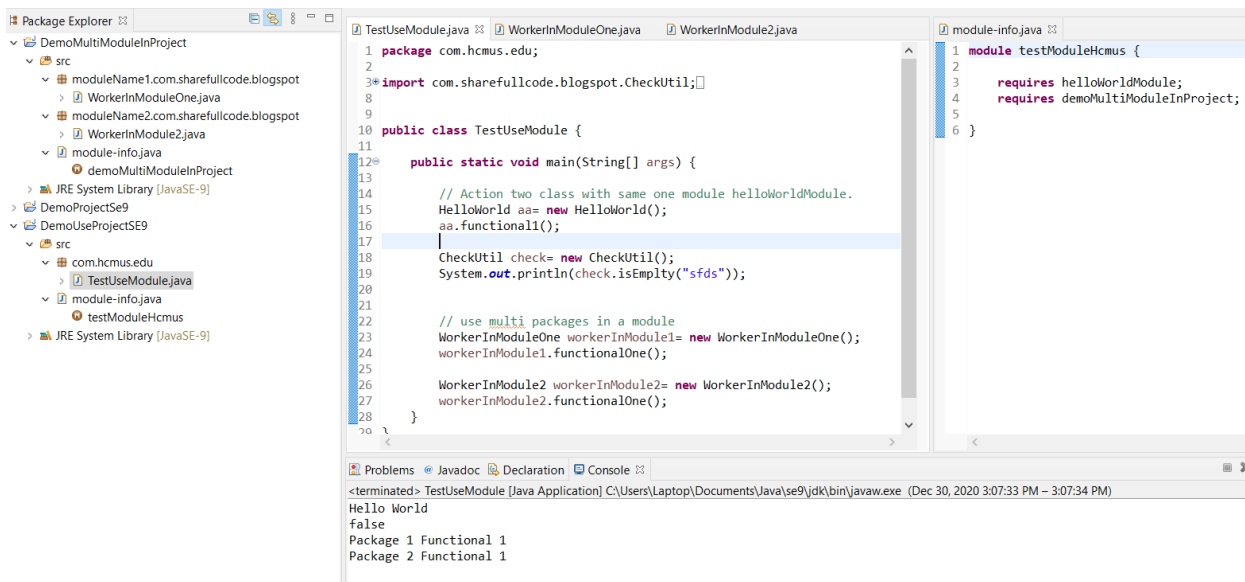


```
module-info.java
1 module demoMultiModuleInProject {
2     exports moduleName1.com.sharefullcode.blogspot;
3     exports moduleName2.com.sharefullcode.blogspot;
4 }
```

Package Explorer

- DemoMultiModuleInProject
  - src
    - moduleName1.com.sharefullcode.blogspot
      - WorkerInModuleOne.java
    - moduleName2.com.sharefullcode.blogspot
      - WorkerInModule2.java
    - module-info.java
      - demoMultiModuleInProject
  - JRE System Library [JavaSE-9]
  - DemoProjectSe9
  - DemoUseProjectSE9

## Use it



## Summary

You can create project with java 9 use modular structure use keyword **exports** the packages in module-info.class and That was used in the others projects by use keyword **requires**. Also use module jar similar a jar normal when you add classpath instead modulepath on project version se9 and later.

**This section is non commercial mainly sharing and advance knowlage for java.This tutorials has referenced document from the list below if you has complain for license, i will remove all from internet. Thank you all everything.**

<https://www.javatpoint.com/java-9-module-system>

<https://howtodoinjava.com/java9/java-9-modules-tutorial/>