

EXECUTION PROCESS

Created by

Sercan Şensülün and Emre KARAKIŞ

Submitted

Yrd.Doç.Dr Tuğkan Tuğlular

TABLE OF CONTENTS

1.INTRODUCTION

2.EXECUTION PROCESS DESCRIPTION

- 1. Project Creation**
- 2. Import Existing Project To Eclipse:**
- 3. Add External JAR files**
- 4. Excel File Path Arrangement**

3.CONCLUSION

1.INTRODUCTION

In this assignment we tried to implement how abstract test cases depending on triangle inequality problem are automated. This implementation explains step by step how you can basically import the project and execute the source code from Eclipse Mars 2 IDE and also adding necessary jar files to the existing project.

2.EXECUTION PROCESS

5. Project Creation

- In order to create a Java Project, initially open the Eclipse IDE and select **File -> New ->Java Project->** and then enter a project name as shown in Figure 1.

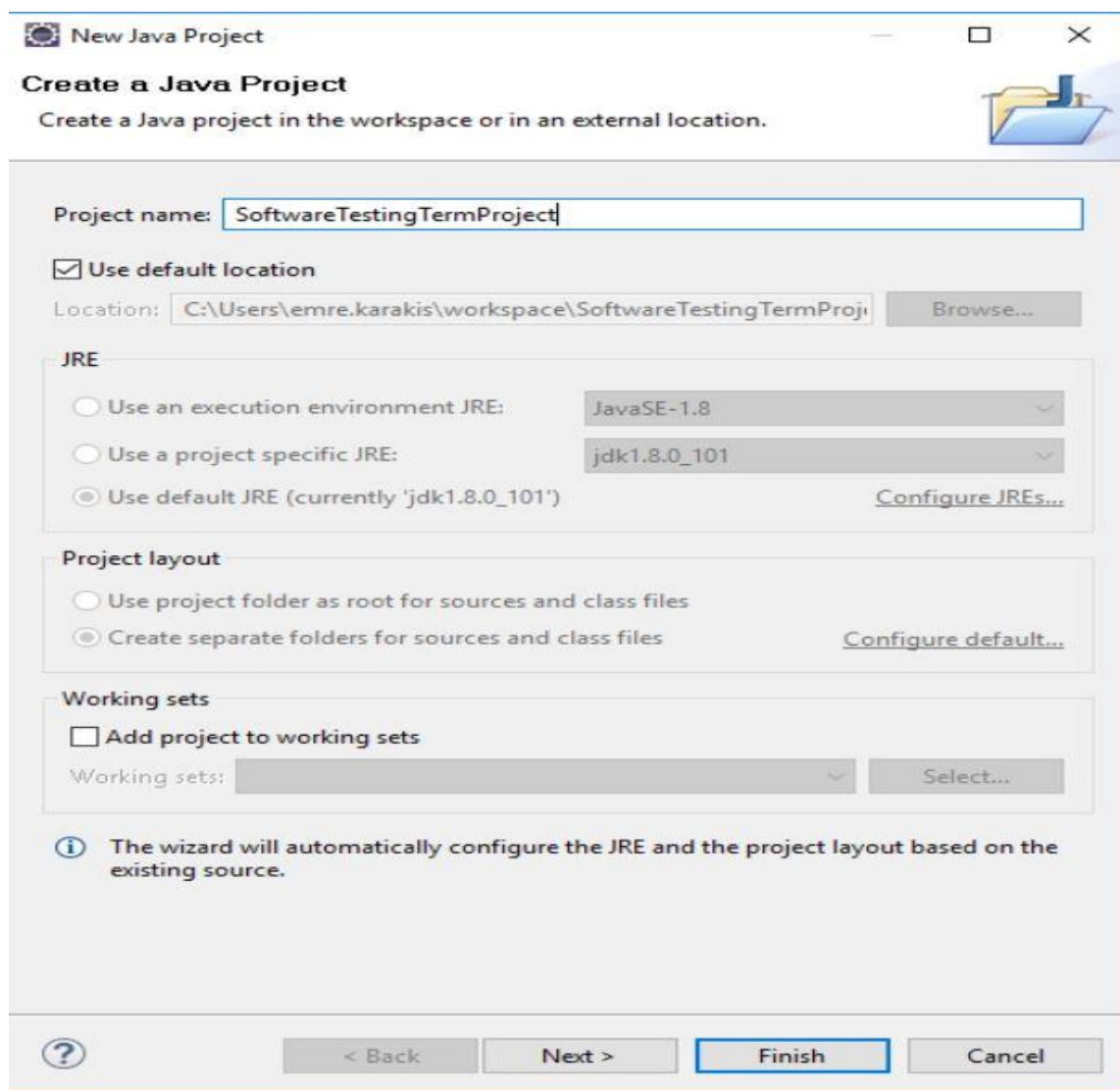


Figure-1

- After entering the project name, press finish button to display a Java Project in the Package Explorer segment.

2.Import Existing Project To Eclipse:

Select **File** and then **Import** as shown in the Figure-2.

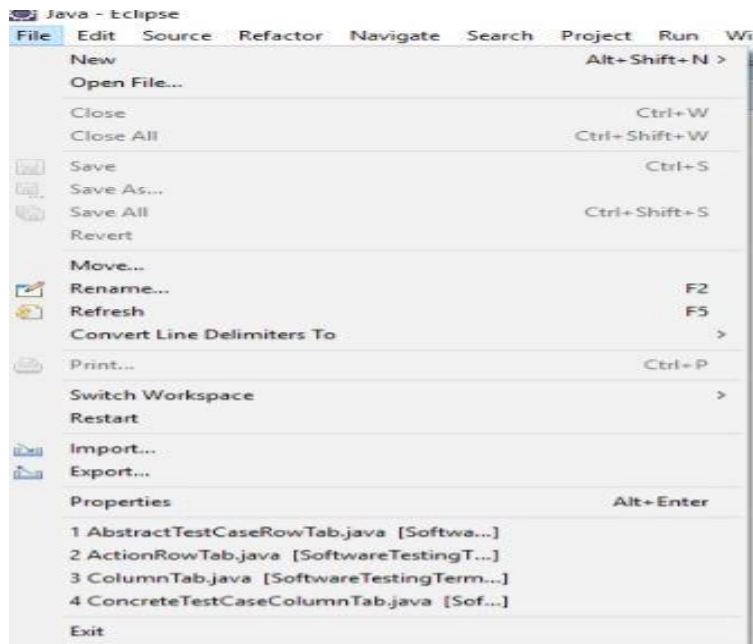


Figure-2

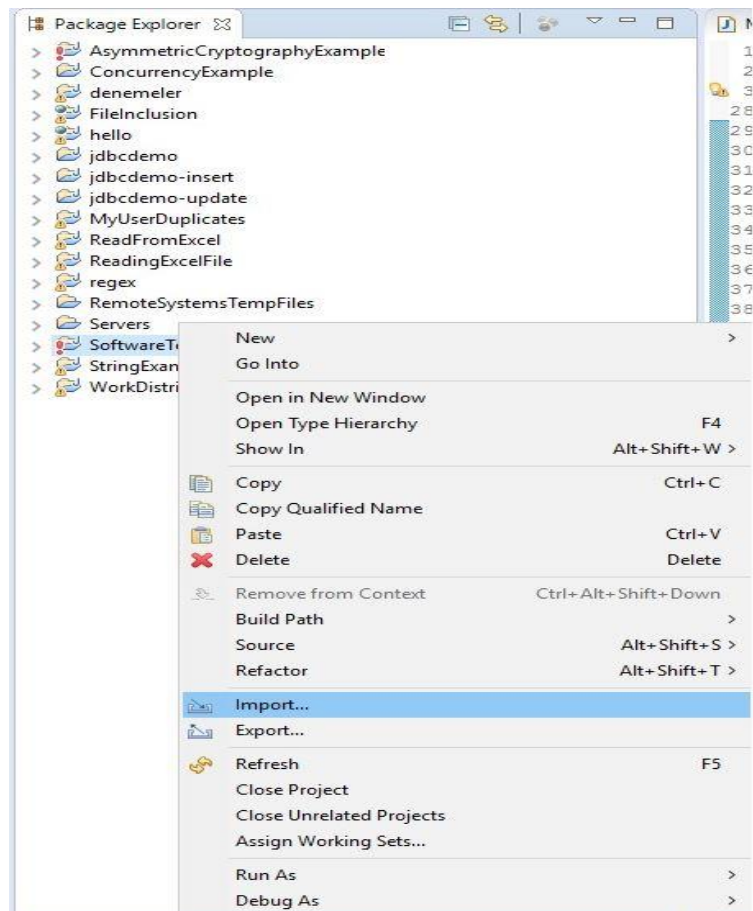


Figure-3

- After selecting Import, Figure-4 will be visible as shown below,

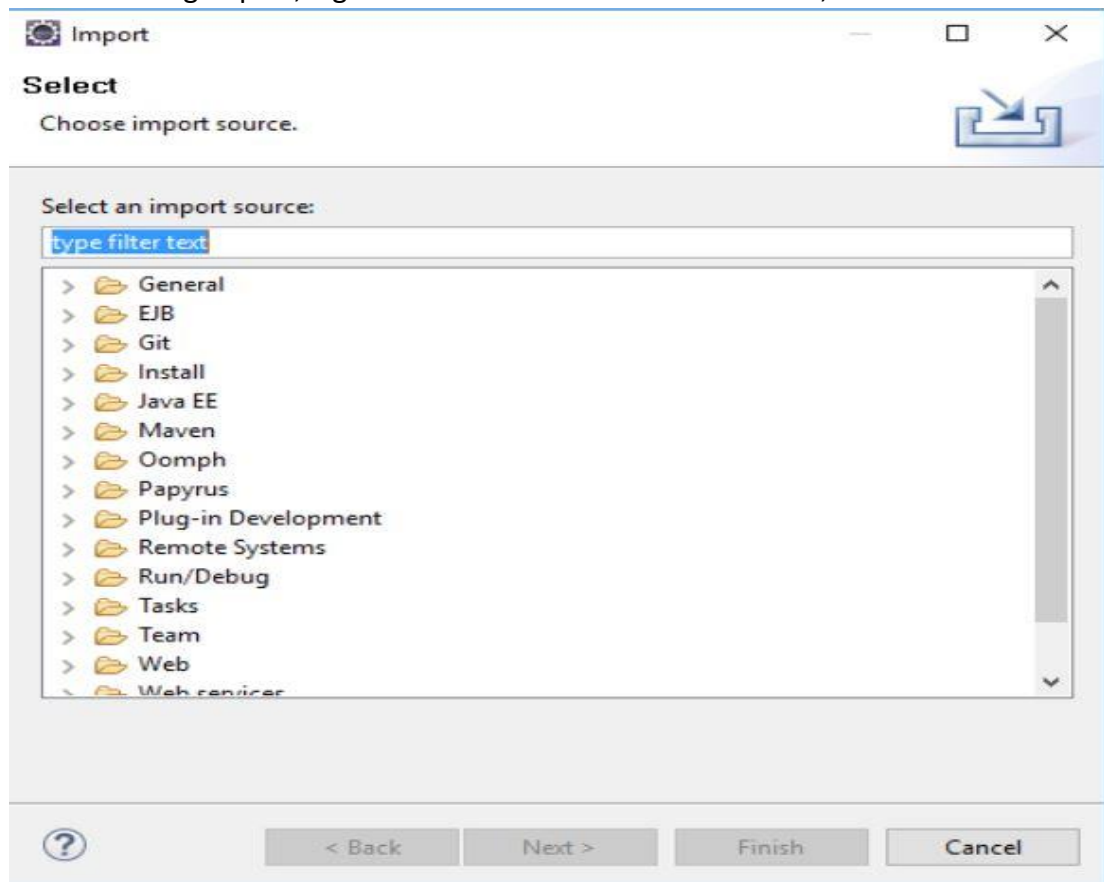


Figure-4

- After that, select **General**

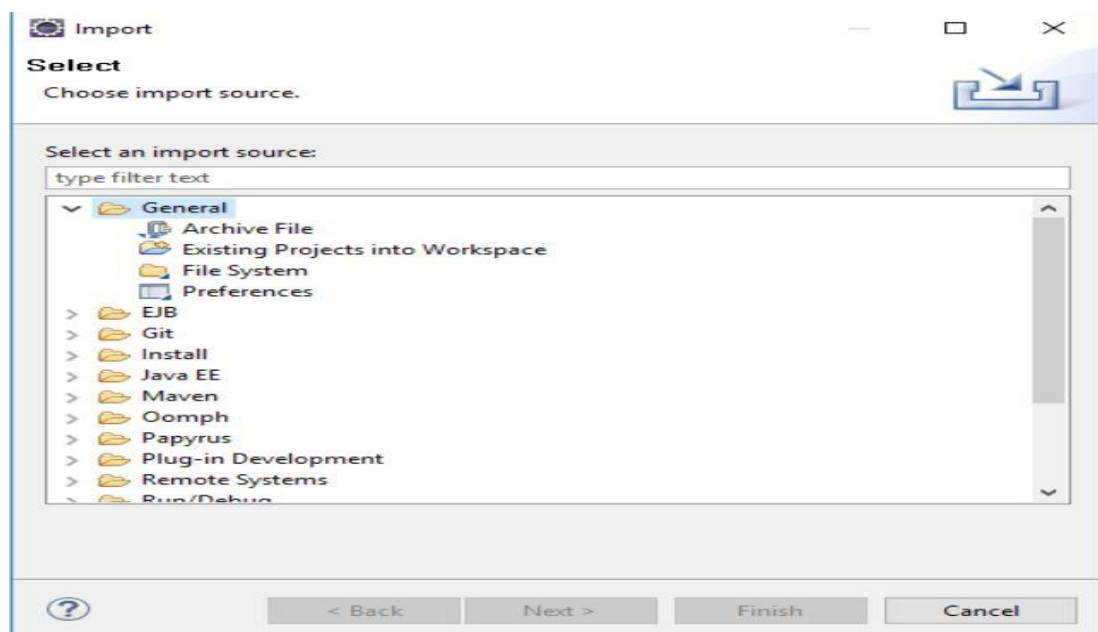


Figure-5

- After that **Select-> File System** and then find out the existing project from your file system by clicking on the **Browse** button.

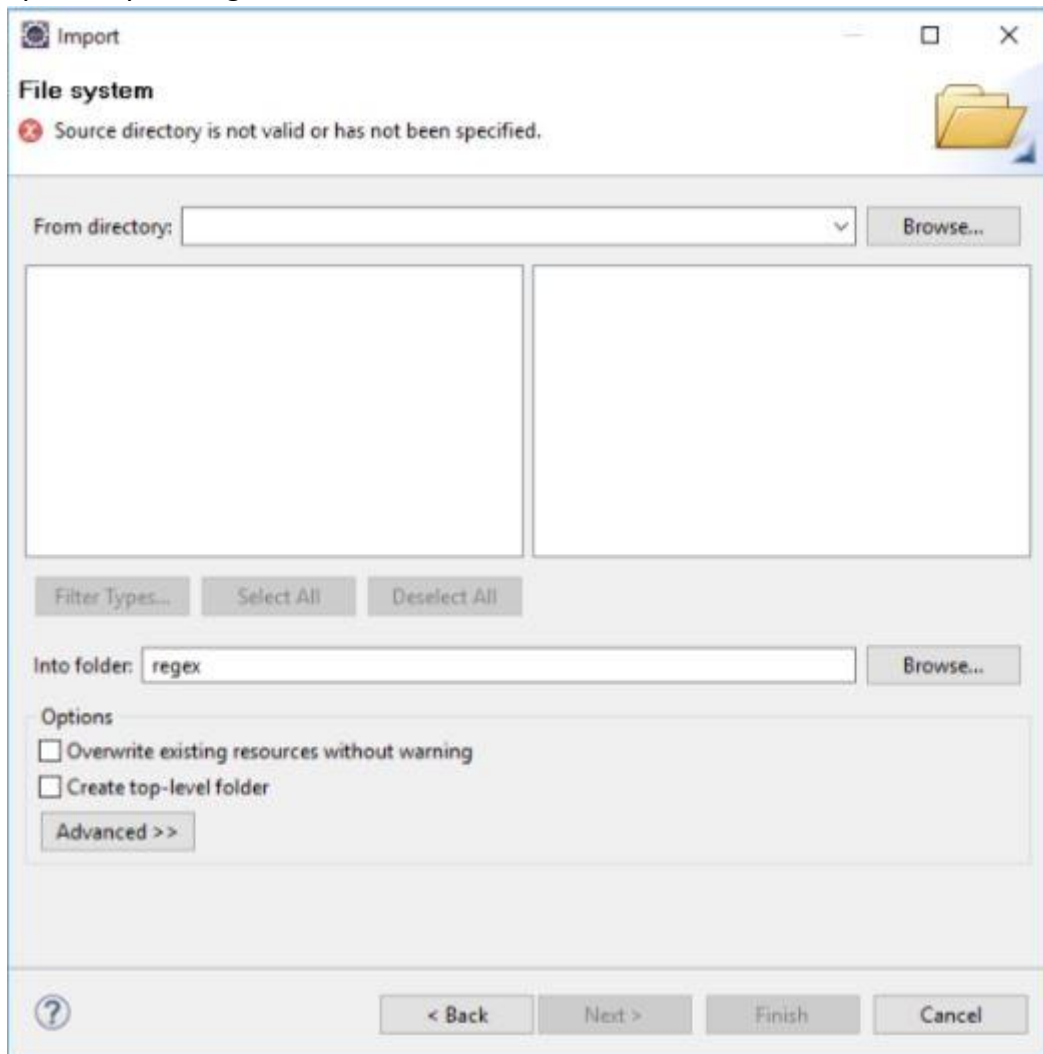


Figure-6

- The next step displays your file system like this

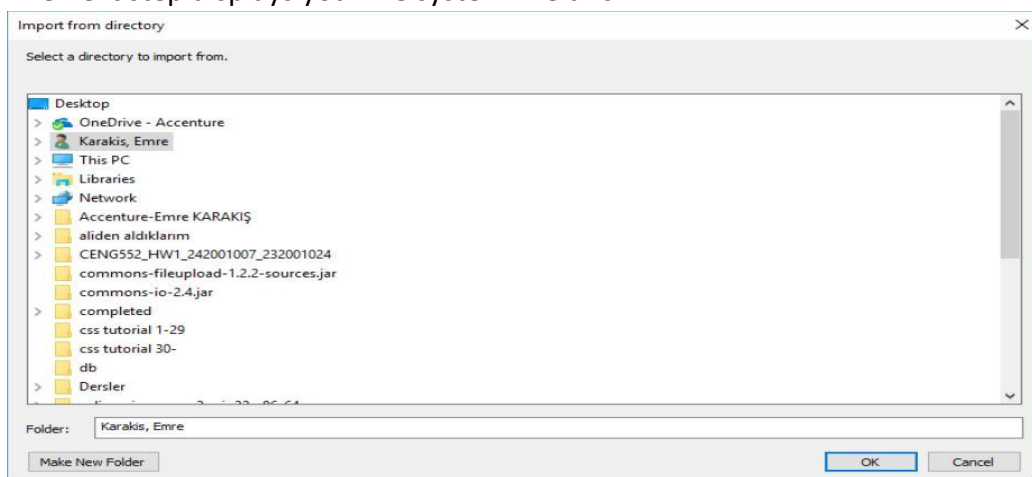


Figure-7

- Choose **CENG552_HW1_24001007_232001024** file in your directory and select **SoftwareTestingTermProject** file and then **From directory segment** will be visible as shown below.

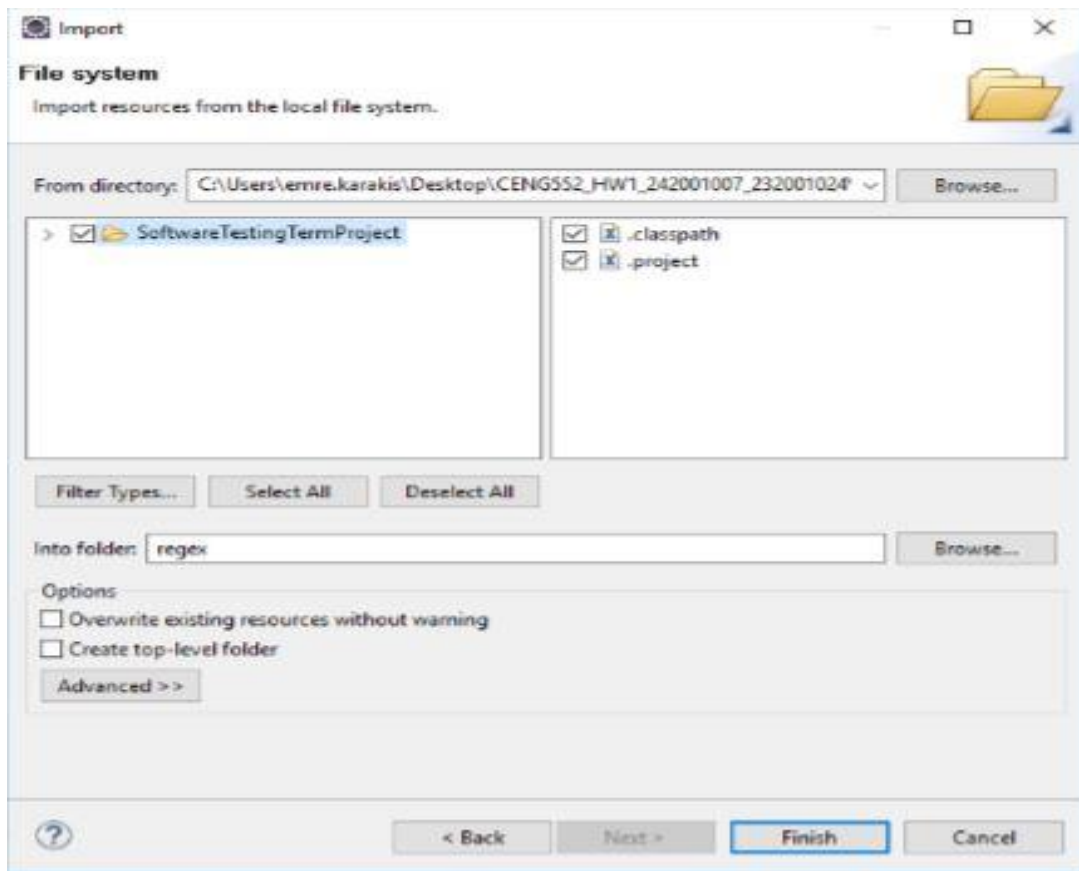


Figure-8

- In the following step , click **Browse** button in the **Into Folder** section as shown in Figure. Select the empty Java project which has previously created in the Eclipse Mars 2.0 IDE and named as "SoftwareTestingTermProject". After choosing SoftwareTestingTermProject file and then click **Finish**.

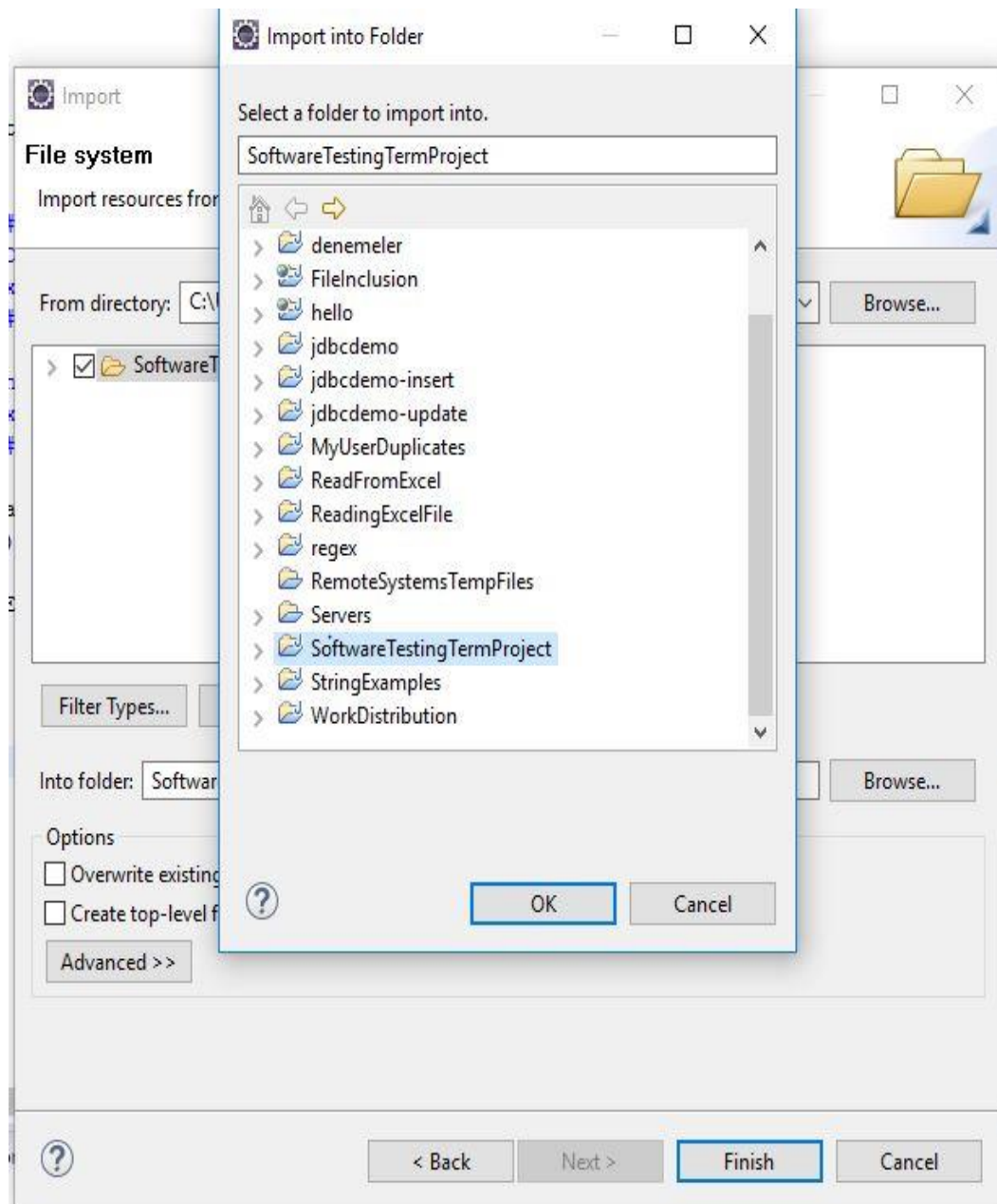


Figure-9

- After clicking **Finish** button, below figure will be visible.
- Click the **Yes to All** and then project will be successfully imported.

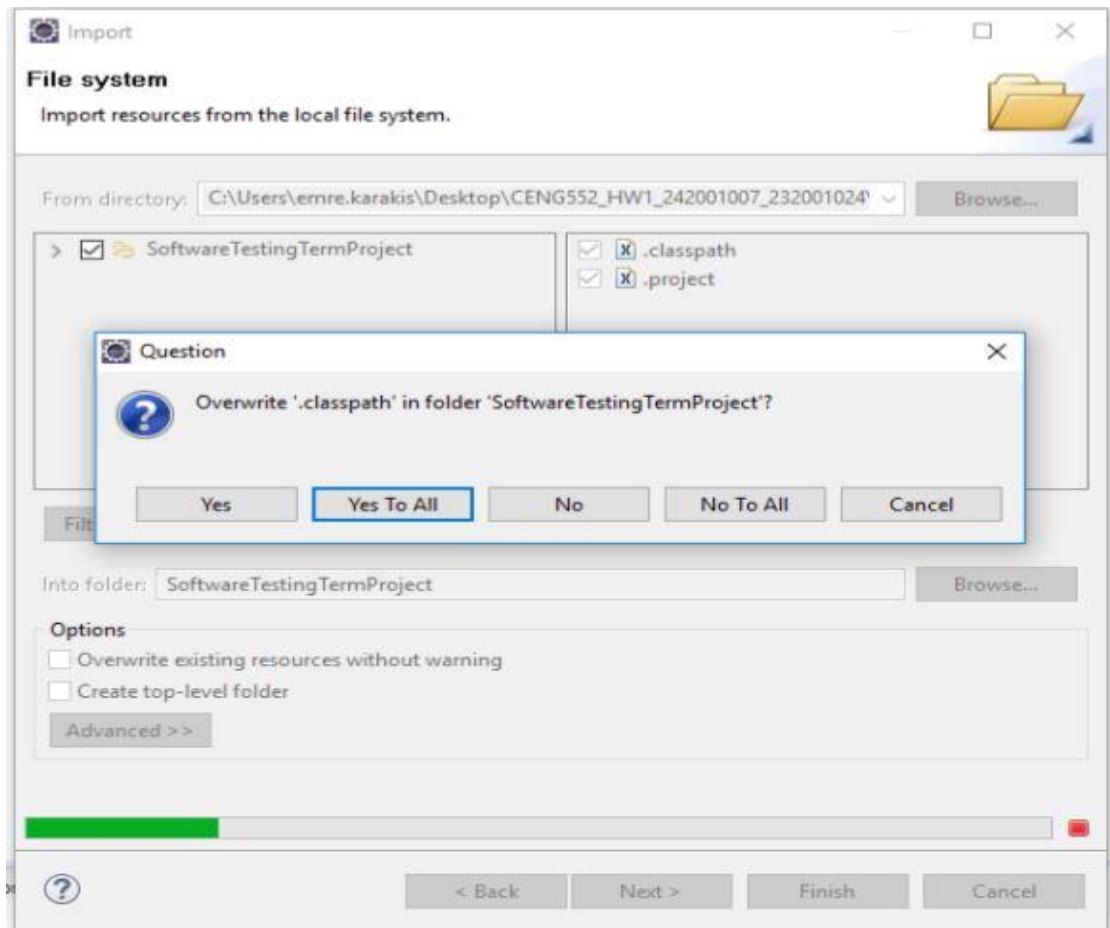


Figure-10

3.Add External JAR files

After importing the existing project in your directory, the project would contain some errors because of lack of jar files as shown below. In order to add jar files, first please go to **jar files** folder which is available under the **CENG552_HW1_24001007_232001024** folder.

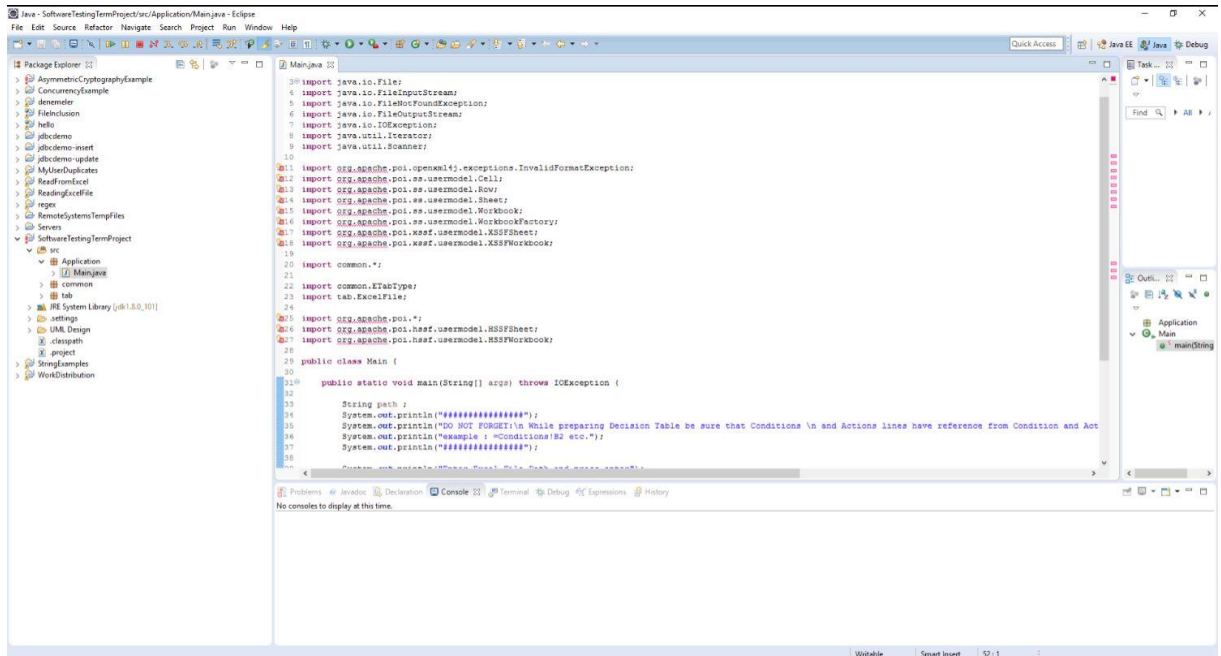


Figure-11

- In order to add jar files, please right click to the project named “SoftwareTestingTermProject” from **Package Explorer** and then select **Build Path -> Configure Build Path** as shown below.

- After selecting Configure Build Path, below figure will be visible.

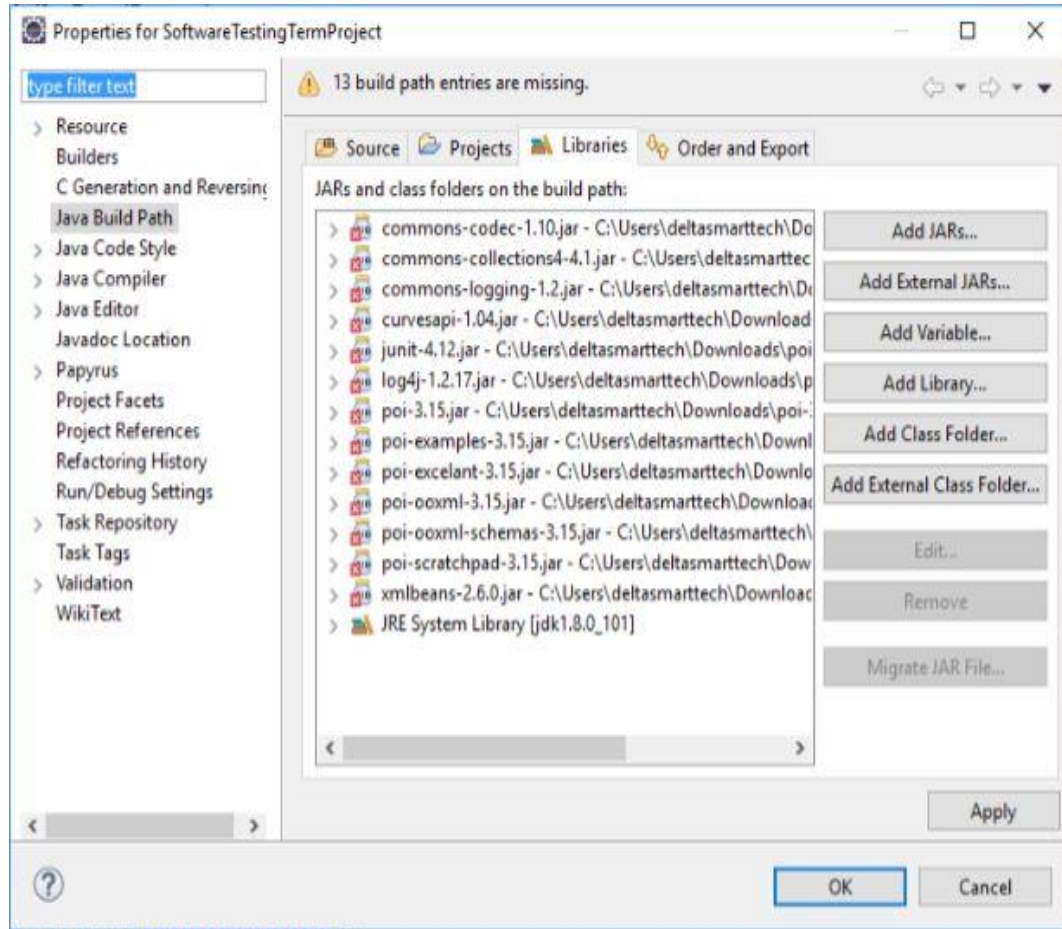


Figure-13

- In that figure, click **Add External JARs** button, go to **jar files** folder which is given in project folder and select firstly and then click **Apply** and then **OK**.

jar files\poi-3.15\poi-3.15.jar

jar files\poi-3.15\poi-examples-3.15.jar

jar files\poi-3.15\poi-excelant-3.15.jar

jar files\poi-3.15\poi-ooxml-3.15.jar

jar files\poi-3.15\poi-ooxml-schemas-3.15.jar

jar files\poi-3.15\poi-scratchpad-3.15.jar

| Name | Date modified | Type | Size |
|----------------------------|------------------|---------------------|----------|
| docs | 13.03.2017 13:38 | File folder | |
| lib | 13.03.2017 13:37 | File folder | |
| ooxml-lib | 13.03.2017 13:37 | File folder | |
| poi-3.15.jar | 17.09.2016 12:50 | Executable Jar File | 2.520 KB |
| poi-examples-3.15.jar | 17.09.2016 12:50 | Executable Jar File | 346 KB |
| poi-excelant-3.15.jar | 17.09.2016 12:50 | Executable Jar File | 31 KB |
| poi-ooxml-3.15.jar | 17.09.2016 12:50 | Executable Jar File | 1.307 KB |
| poi-ooxml-schemas-3.15.jar | 17.09.2016 12:50 | Executable Jar File | 5.722 KB |
| poi-scratchpad-3.15.jar | 17.09.2016 12:50 | Executable Jar File | 1.294 KB |

Figure-14

- After that, repeat the adding external jar process for selecting jar files under **jar files\poi-3.15\ooxml-lib**
curvesapi-1.04.jar
xmlbeans-2.6.0.jar
- After that, repeat the adding external jar process for selecting jar files under **jar files\poi-3.15\lib**
commons-codec-1.10.jar
commons-collections4-4.1.jar
commons-logging-1.2.jar
junit-4.12.jar
log4j-1.2.17.jar

By adding these jar files shown in above, all errors in Eclipse will disappear.

4.Excel File Path Arrangement

- Please open the prepared input excel file with any Microsoft Excel Document
- Come into the directory of the excel file.

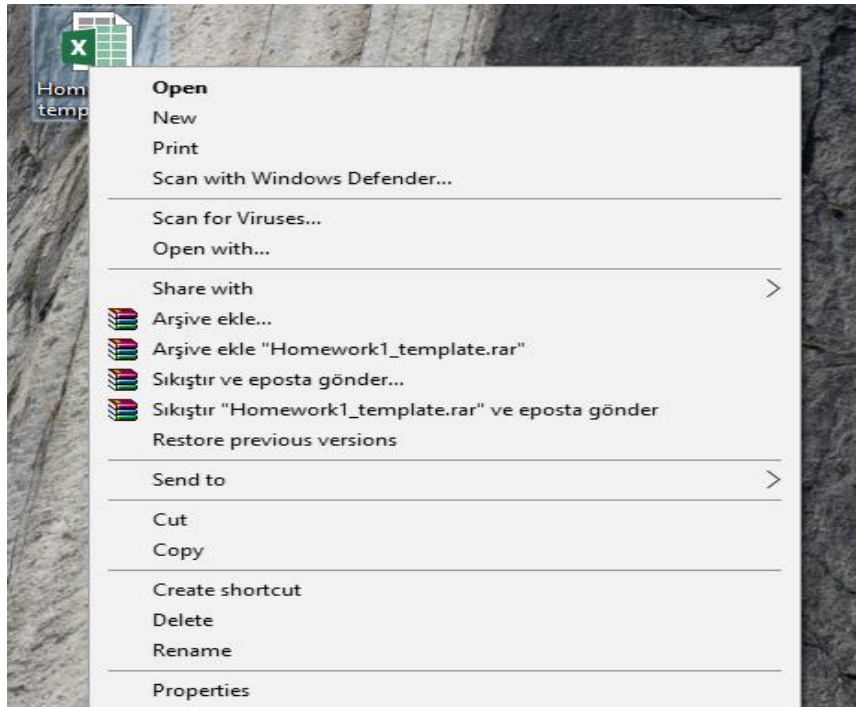


Figure-14

- Go to excel file and right click to the file and then select **Properties** ->

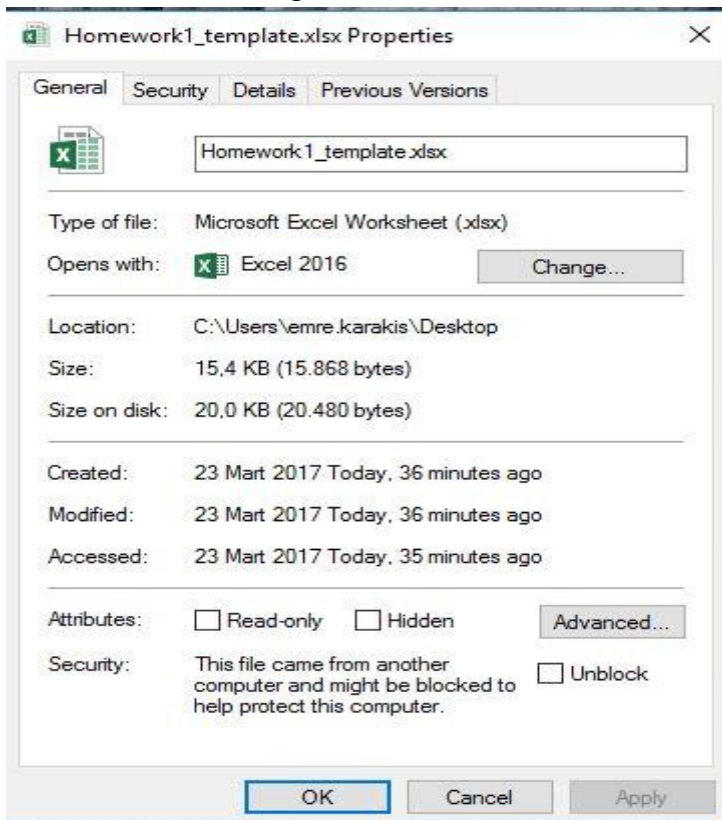


Figure-15

- After that copy the file path for execution such as
“C:\Users\emre.karakis\Desktop\Homework1_template.xlsx”

In that step, everything is available to execute source code from Eclipse IDE. Example execution would seem something like this.

```

#####
We prepared a sample excel file to test under Input Folder
After execution , Please check excel file under Output Folder
#####
Please be sure that Apache Poi library was added as shown on our guidelines
You could find guideline file source under Guidelines folder
You could find required jar files under Jar Files folder
#####
For homework_1, report was prepared under Reports folder
If you get trouble about Jar files or other things in project
please contact us sercansensulun@gmail.com or emrekarakis@gmail.com
#####
Please press enter to execute program

Requirement list ready !
Input List ready !
Output List ready !
Condition list ready !
Action list ready !
Decision Table Ready !
Please check excel file under Output Folder , Its Abstract Test Cases tab was updated.

```

Figure-16

If you see this result, you can check the excel file inside the Output folder. The decision table tab would seem as shown in Figure 17 and Abstract test cases would be generated like Figure-18.

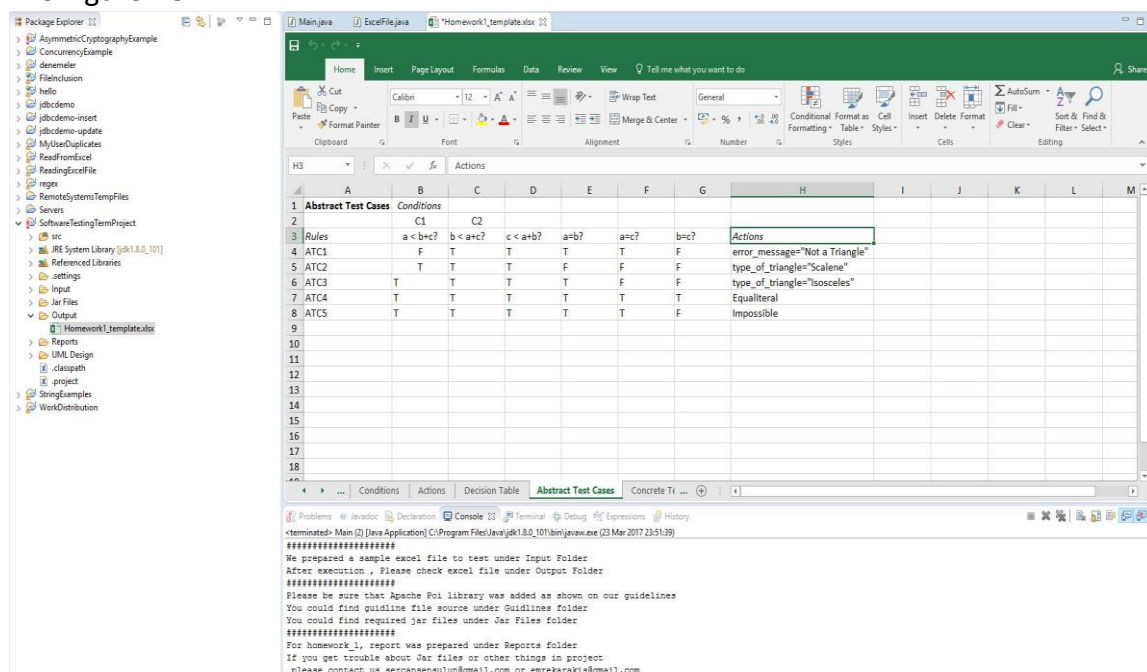


Figure-17

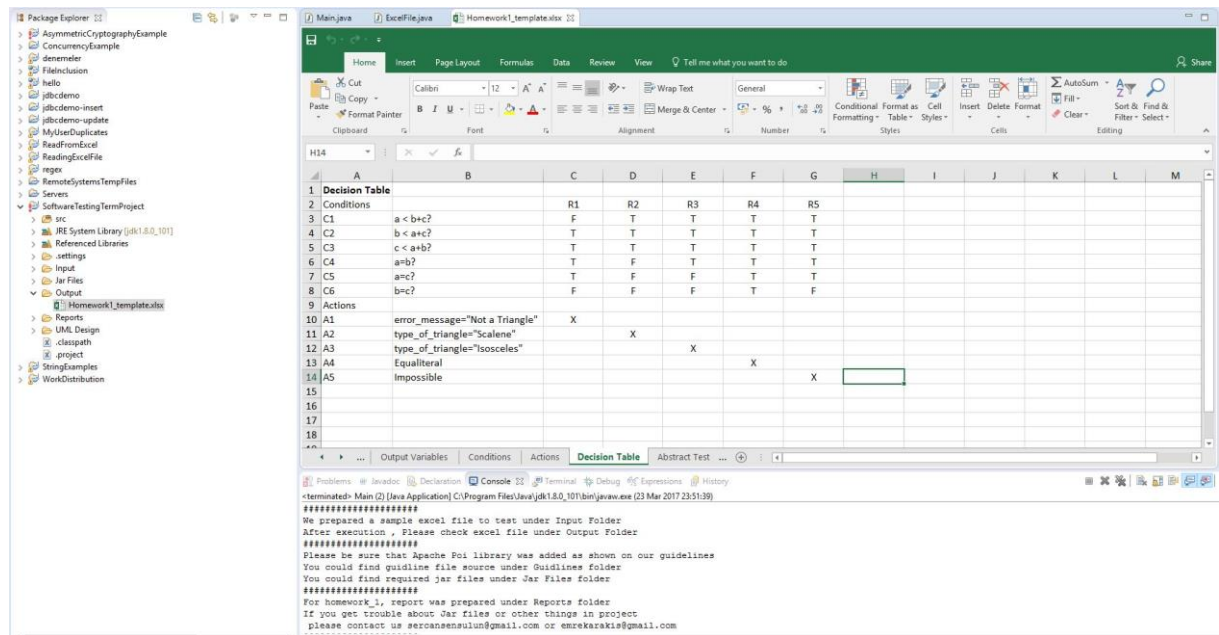


Figure-18

CONCLUSION

As a consequence, in this documentation we tried to explain the basic steps of executions by giving input file to the system and then getting output file including abstract test cases which is automatically filled by the program to successfully implement abstract test case model.