EXECUTION PROCESS

Created by

Sercan Şensülün and Emre KARAKIŞ

Submitted

Yrd.Doç.Dr Tuğkan Tuğlular

TABLE OF CONTENTS

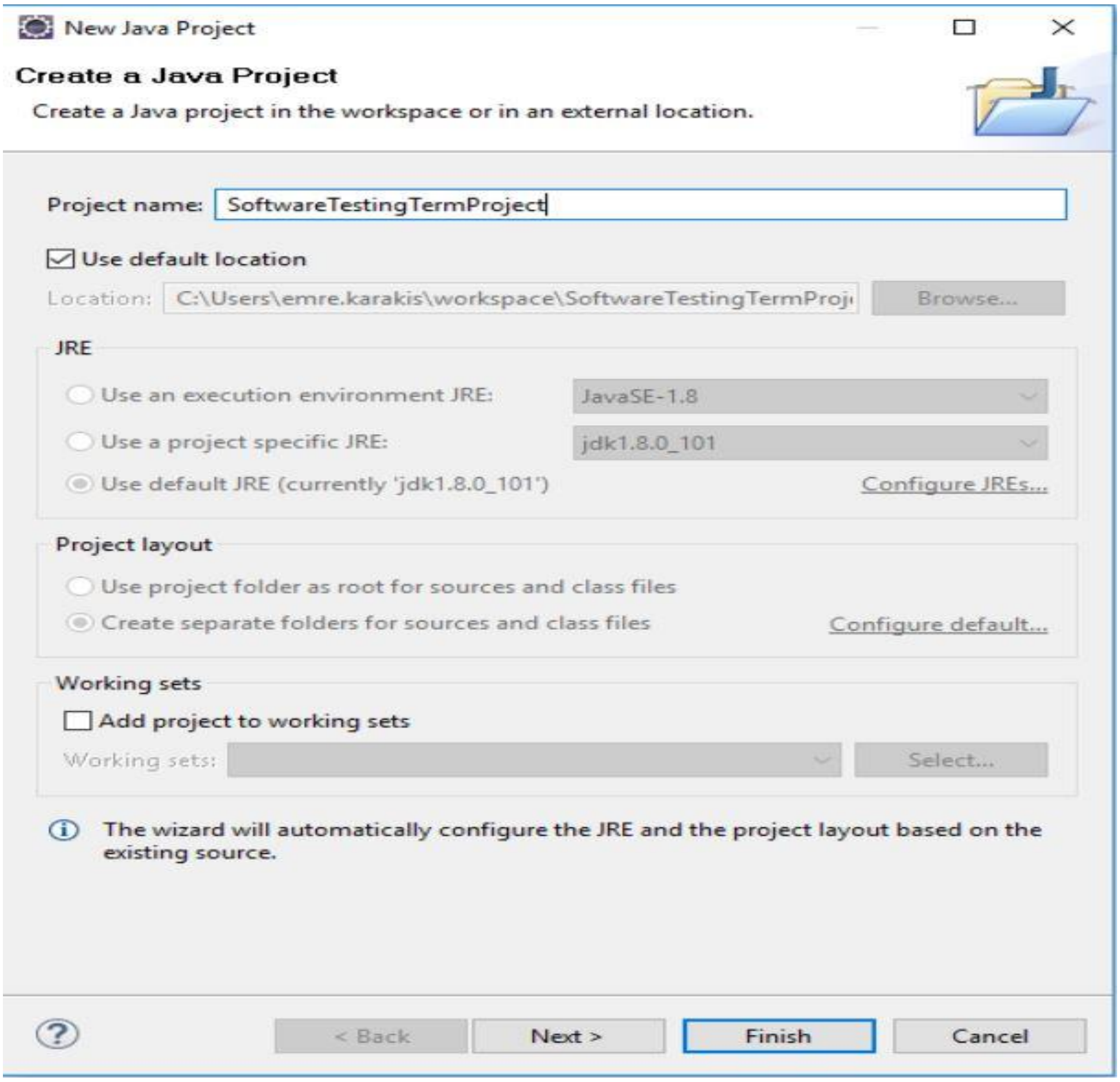
1.INTRODUCTION

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 basicallyimport 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

6. 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.



New Java Project

Create a Java Project

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

Project name: SoftwareTestingTermProject

☑ Use default location

Location: C:\Users\emre.karakis\workspace\SoftwareTestingTermProj

JRE

○ Use an execution environment JRE: JavaSE-1.8

○ Use a project specific JRE: jdk1.8.0_101

● Use default JRE (currently 'jdk1.8.0_101')

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

Working sets

□ Add project to working sets

Working sets:

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

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.

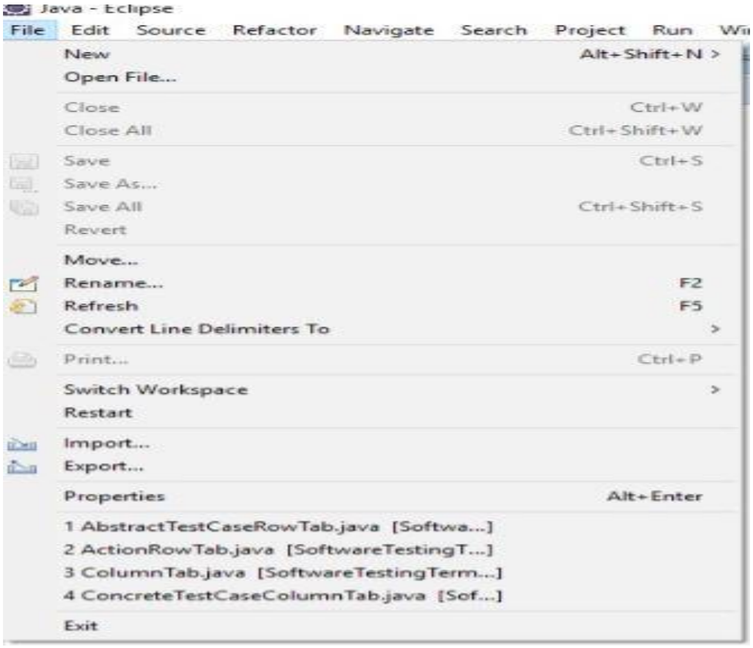


Figure-2

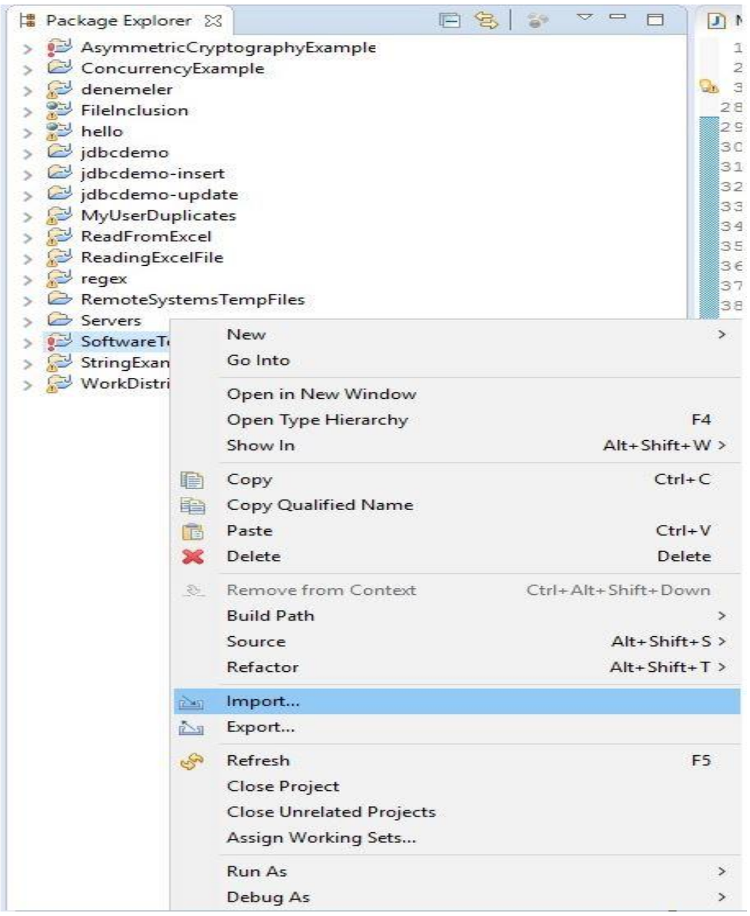
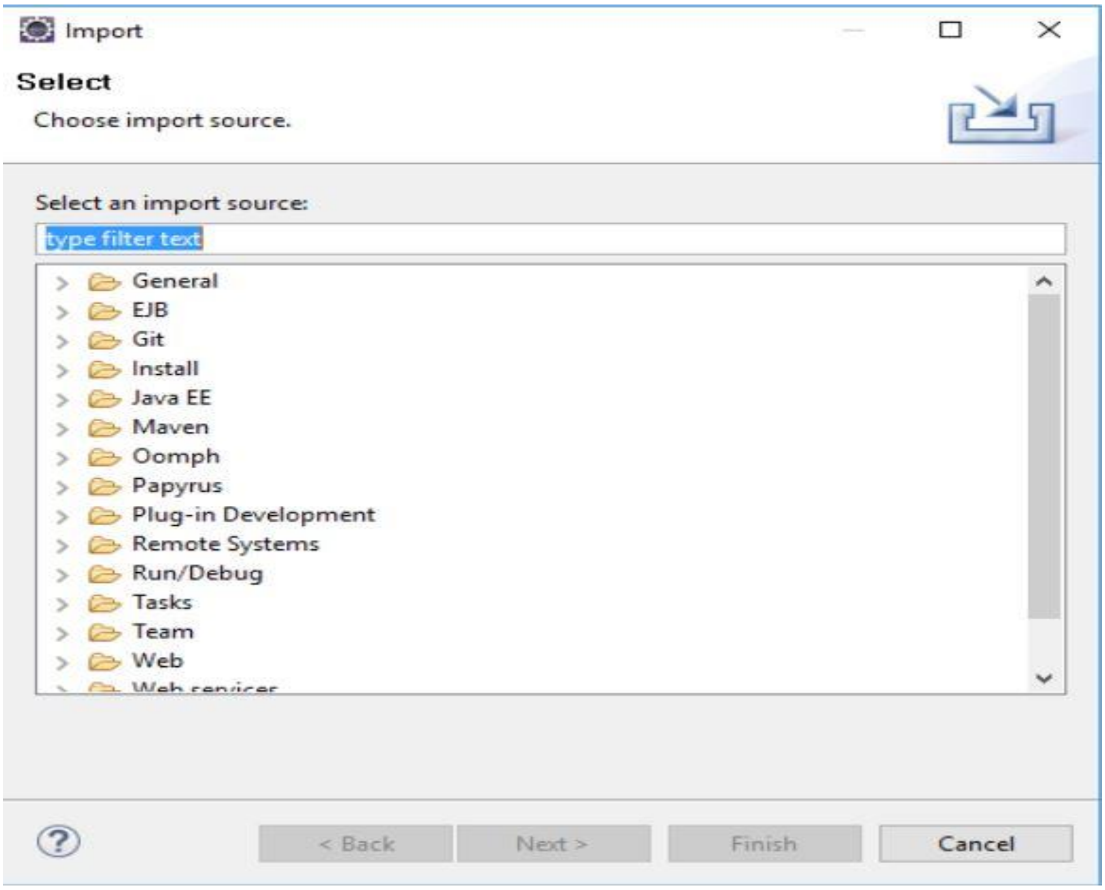

Package Explorer
AsymmetricCryptographyExample
ConcurrencyExample
denemeler
FileInclusion
hello
jdbcdemo
jdbcdemo-insert
jdbcdemo-update
MyUserDuplicates
ReadFromExcel
ReadingExcelFile
regex
RemoteSystemsTempFiles
Servers
SoftwareT
StringExan
WorkDistri
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As

Figure-3

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



Import

Select

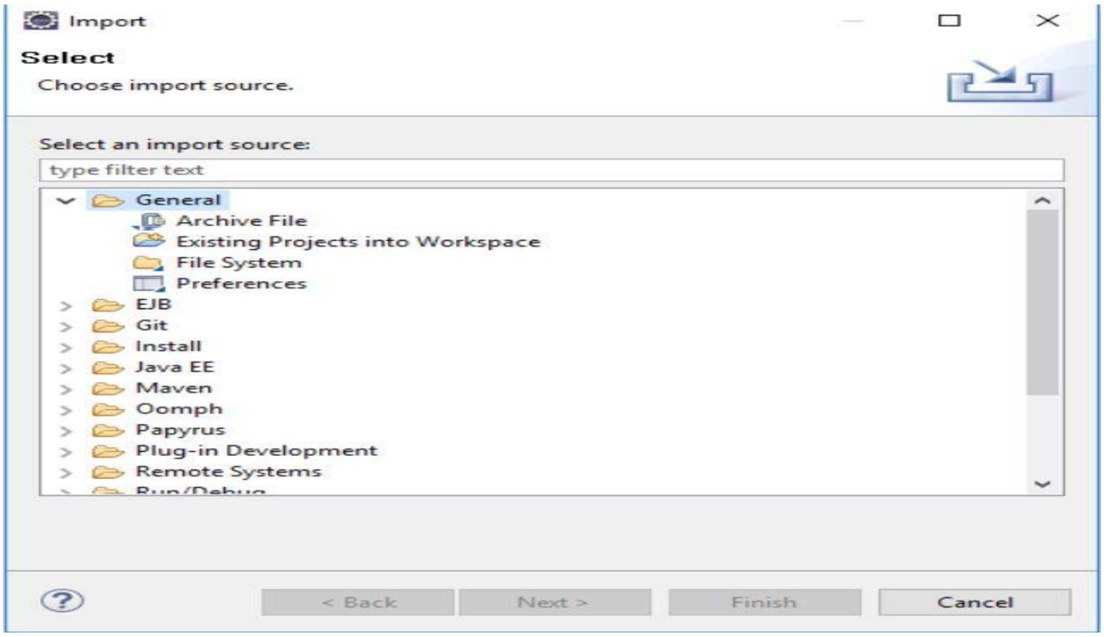
Choose import source.

Select an import source:

type filter text

Figure-4

- After that, select **General**



Import

Select

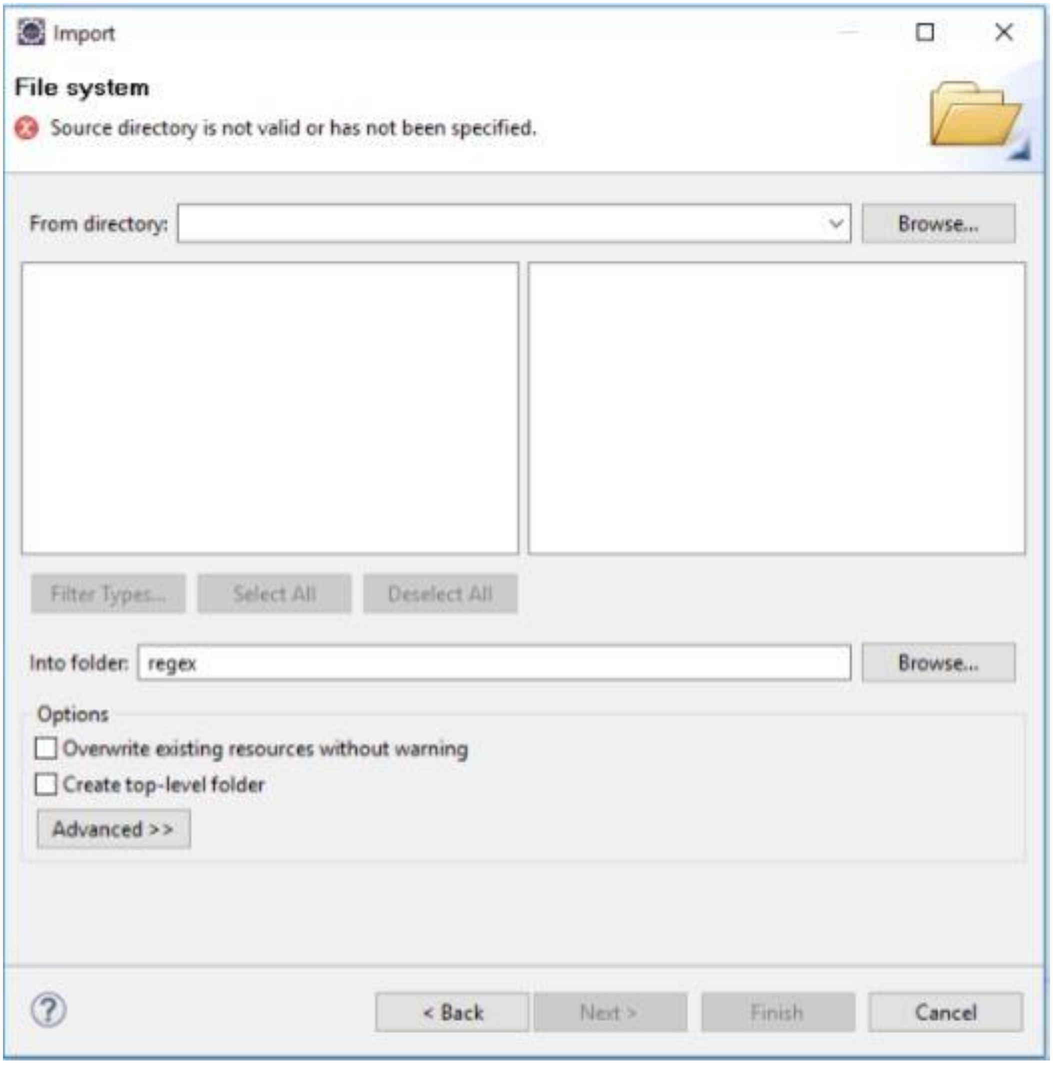
Choose import source.

Select an import source:

type filter text

Figure-5

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



Import

File system

Source directory is not valid or has not been specified.

From directory:

Into folder: regex

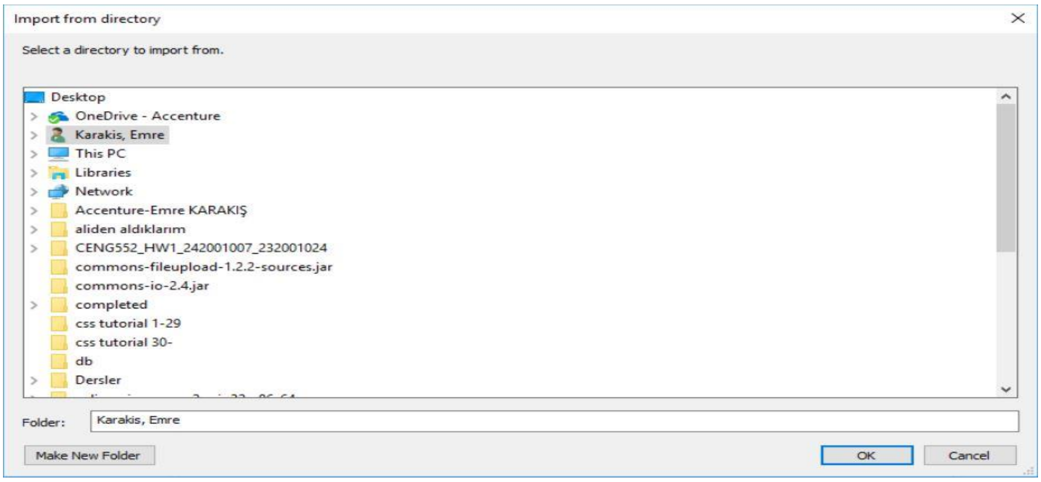
Options

□ Overwrite existing resources without warning

□ Create top-level folder

Figure-6

- The next step displays your file system like this



Import from directory

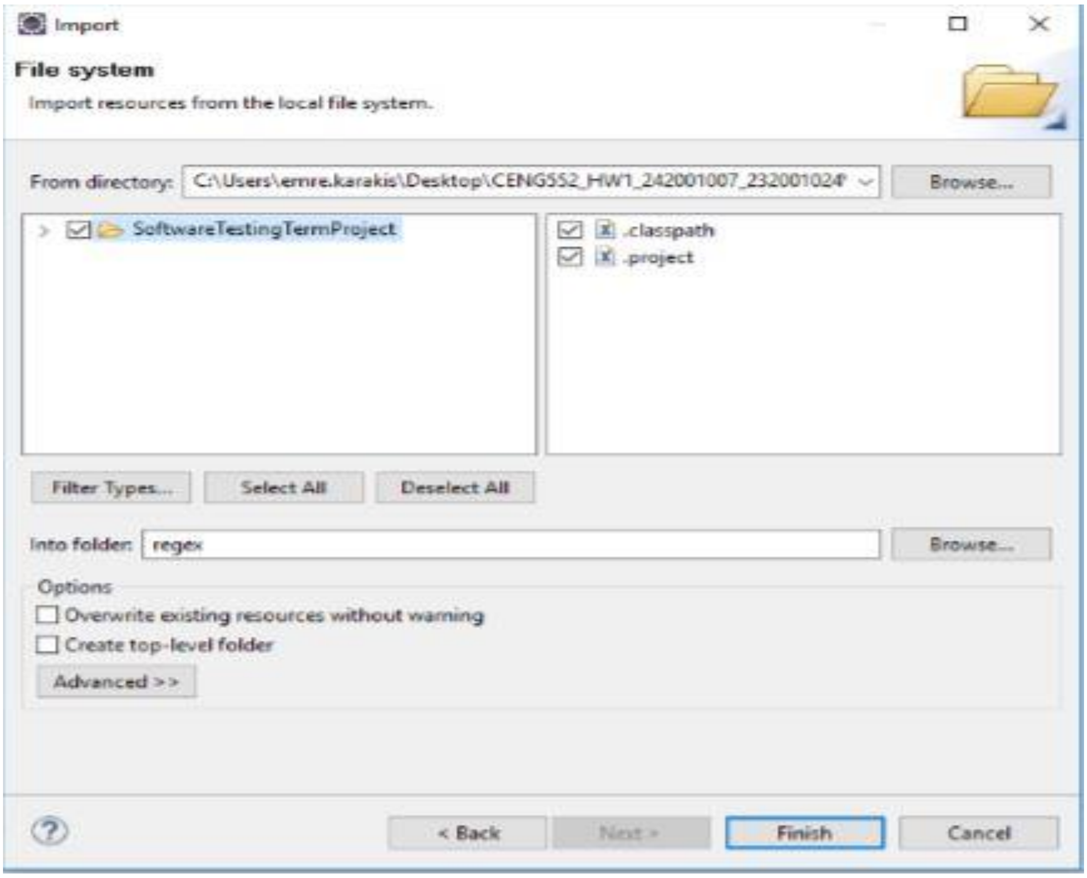
Select a directory to import from.

Desktop
OneDrive - Accenture
Karakis, Emre
This PC
Libraries
Network
Accenture-Emre KARAKIŞ
aliden aldıklarım
CENG552_HW1_242001007_232001024
commons-fileupload-1.2.2-sources.jar
commons-io-2.4.jar
completed
css tutorial 1-29
css tutorial 30-
db
Dersler

Folder: Karakis, Emre

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.



Import

File system

Import resources from the local file system.

From directory: C:\Users\emre.karakis\Desktop\CENG552_HW1_242001007_232001024

☑ SoftwareTestingTermProject

☑ .classpath

☑ .project

Into folder: regex

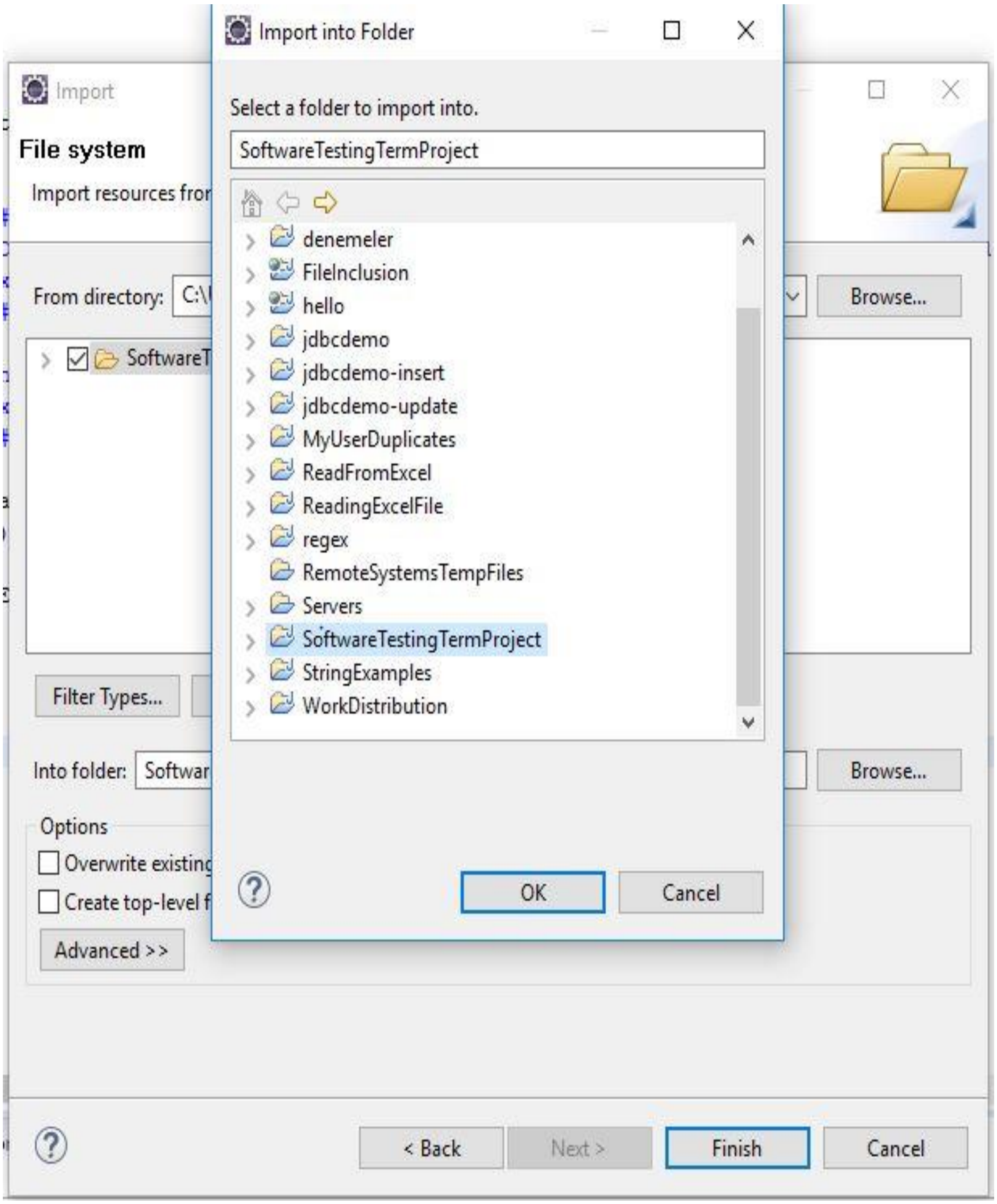
Options

□ Overwrite existing resources without warning

□ Create top-level folder

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**.

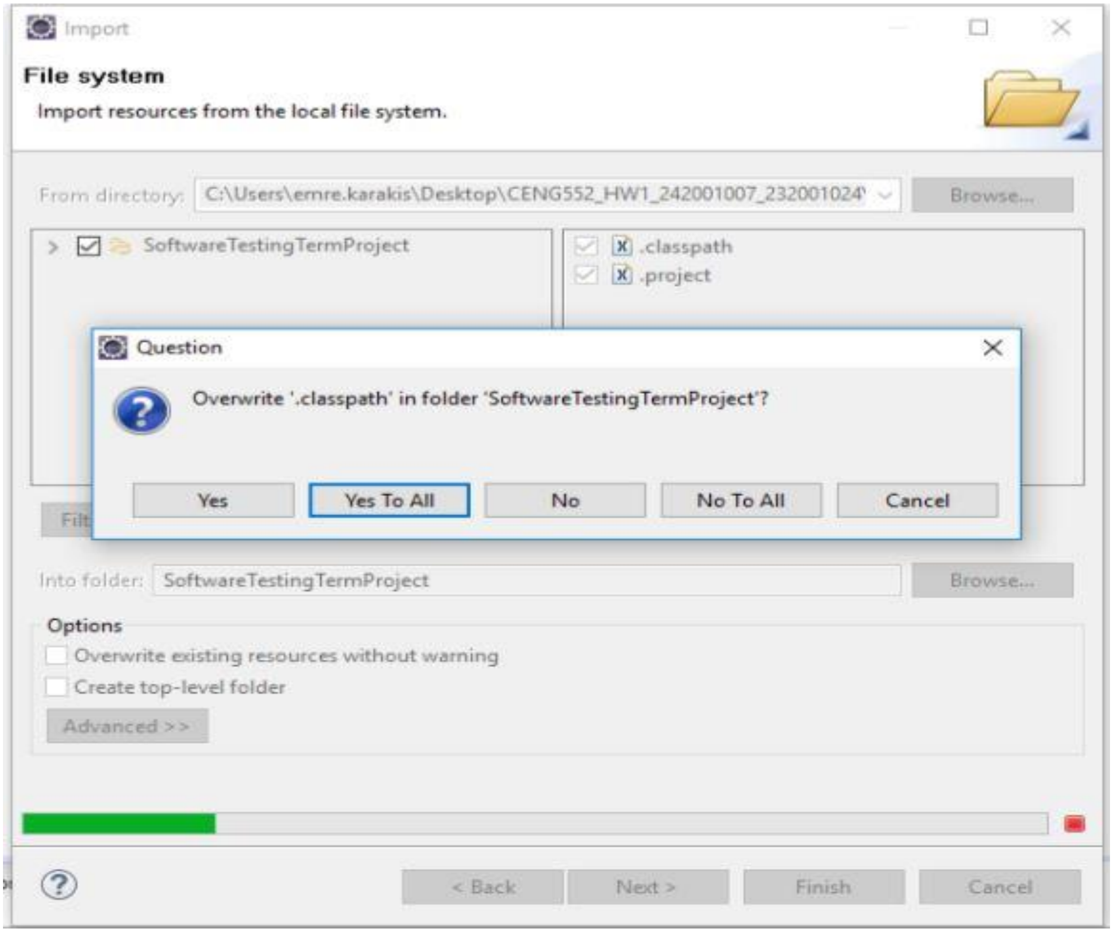


Select a folder to import into.

SoftwareTestingTermProject

Figure-9

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



Import

File system

Import resources from the local file system.

From directory: C:\Users\emre.karakis\Desktop\CENG552_HW1_242001007_232001024\

☑ SoftwareTestingTermProject

☑ .classpath

☑ .project

Question

Overwrite '.classpath' in folder 'SoftwareTestingTermProject'?

Into folder: SoftwareTestingTermProject

Options

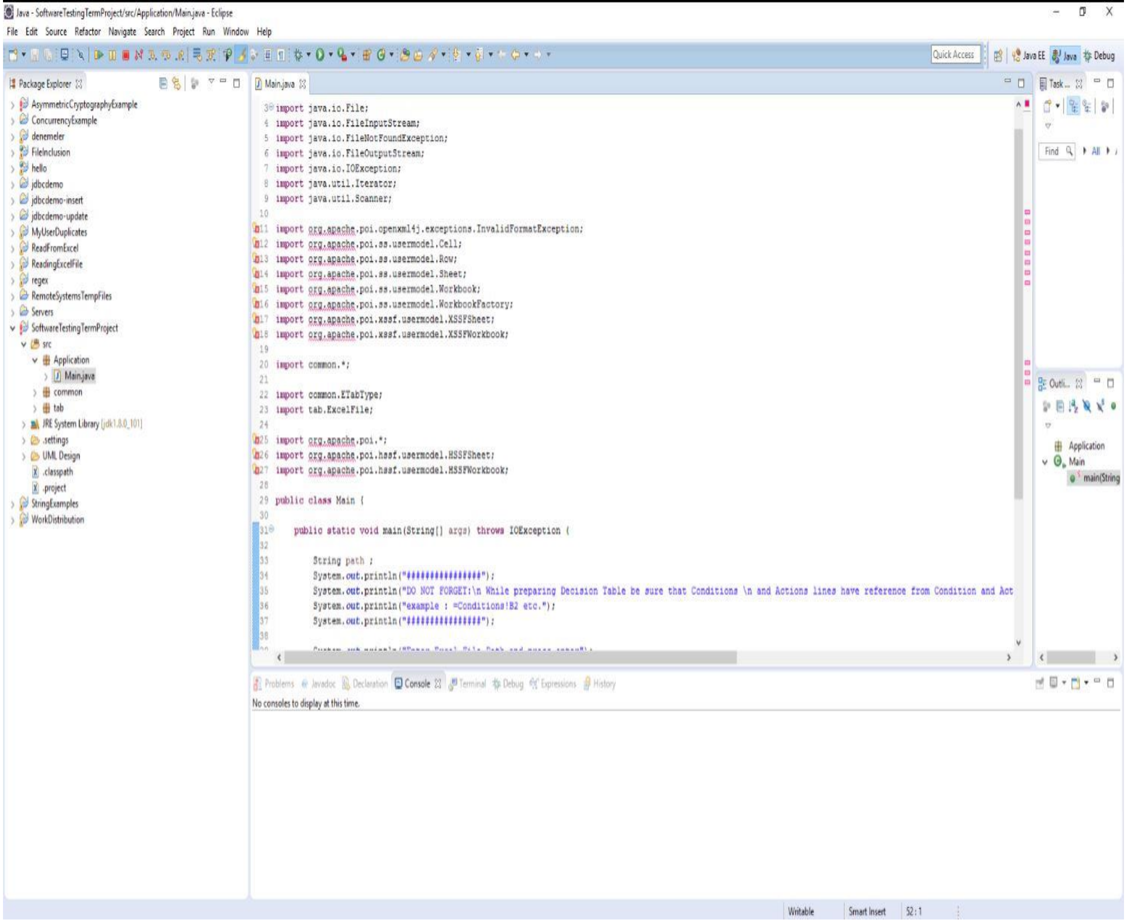
□ Overwrite existing resources without warning

□ Create top-level folder

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.

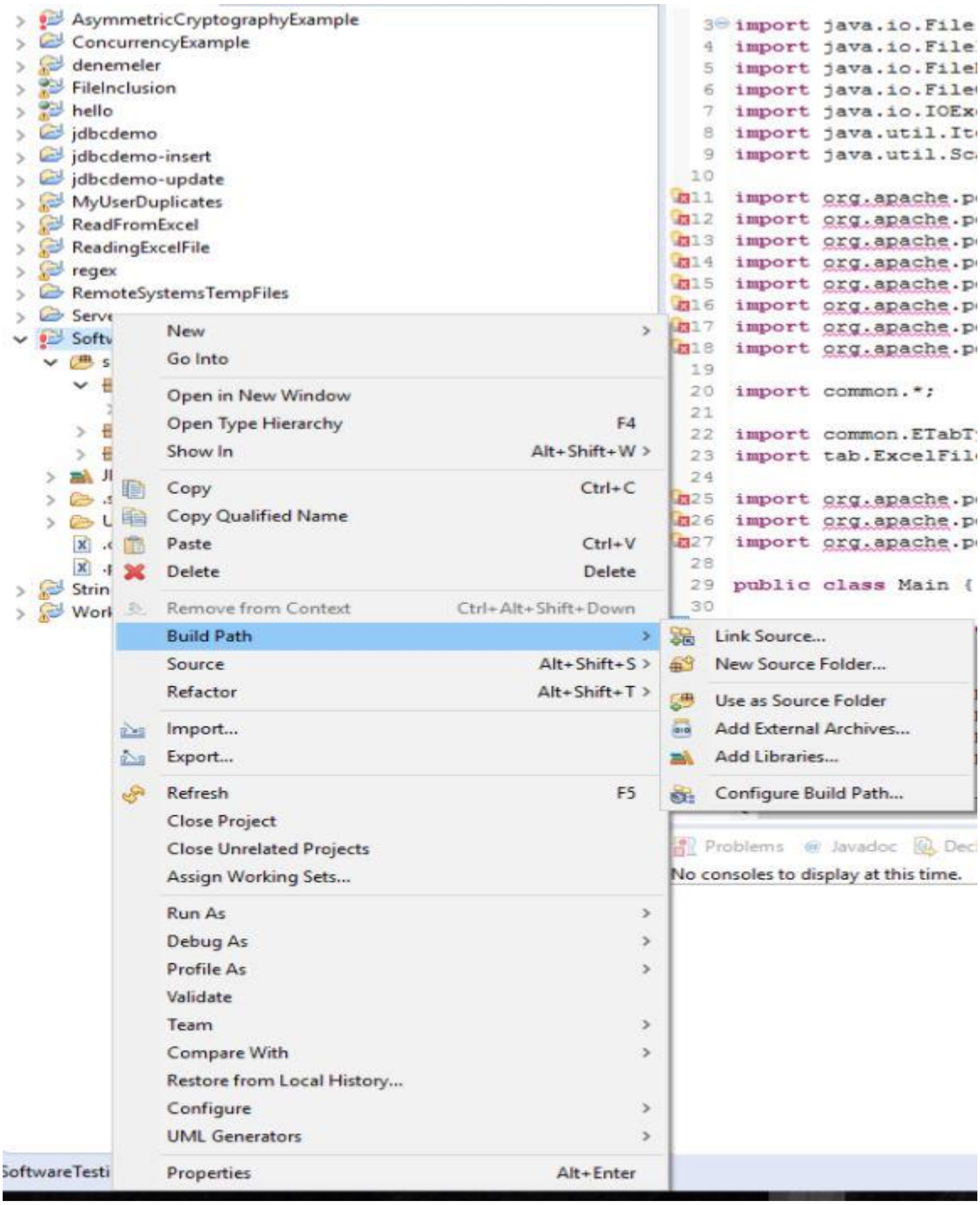


```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Iterator;
import java.util.Scanner;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import common.*;

import common.ETabType;
import tab.ExcelFile;

import org.apache.poi.*;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;

public class Main {

    public static void main(String[] args) throws IOException {

        String path ;
        System.out.println("################");
        System.out.println("DO NOT FORGET:\n While preparing Decision Table be sure that Conditions \n and Actions lines have reference from Condition and Act
        System.out.println("example : =Conditions!B2 etc.");
        System.out.println("################");
```

No consoles to display at this time.

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.

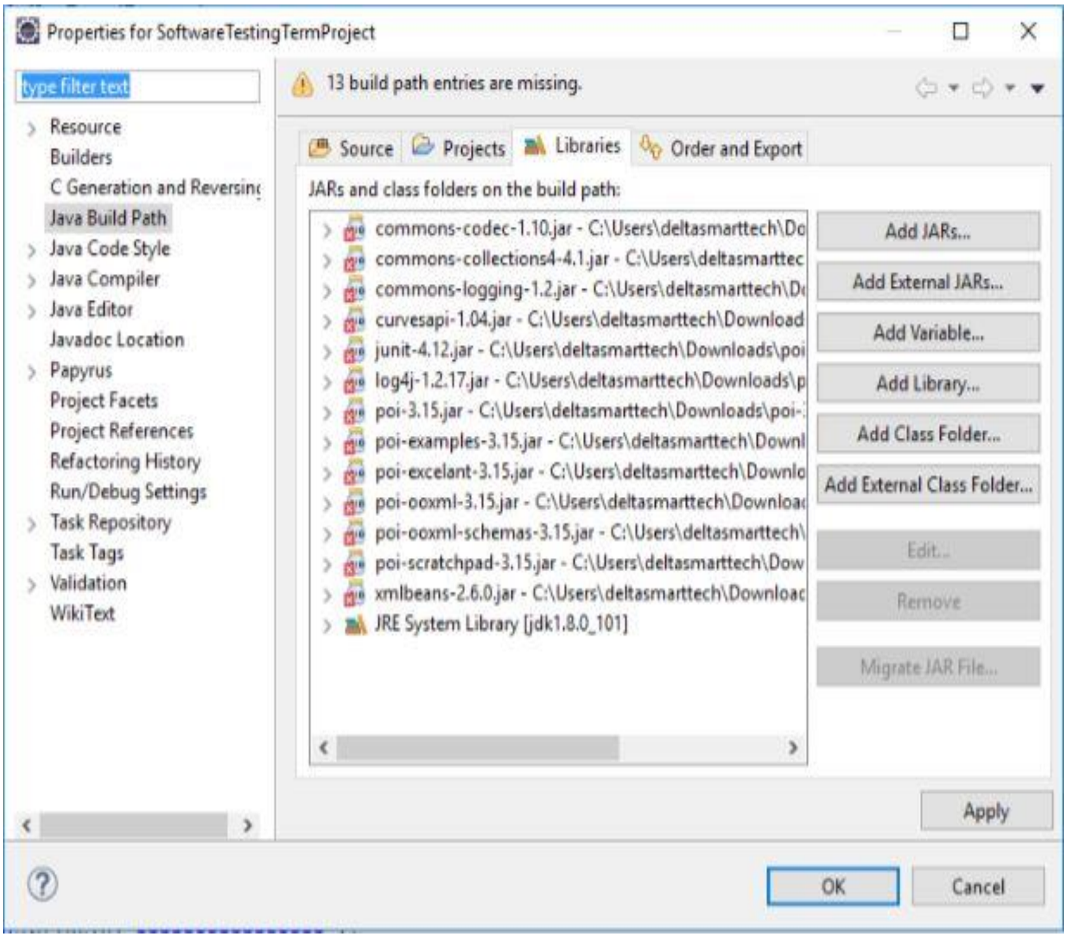


```
import java.io.File
import java.io.File
import java.io.FileI
import java.io.File
import java.io.IOEx
import java.util.It
import java.util.Sc

import org.apache.p
import org.apache.p
import org.apache.p
import org.apache.p
import org.apache.p
import org.apache.p
import org.apache.p
import org.apache.p

import common.*;

import common.ETabT
import tab.ExcelFil

import org.apache.p
import org.apache.p
import org.apache.p

public class Main {
```

No consoles to display at this time.

Figure-12

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



Properties for SoftwareTestingTermProject

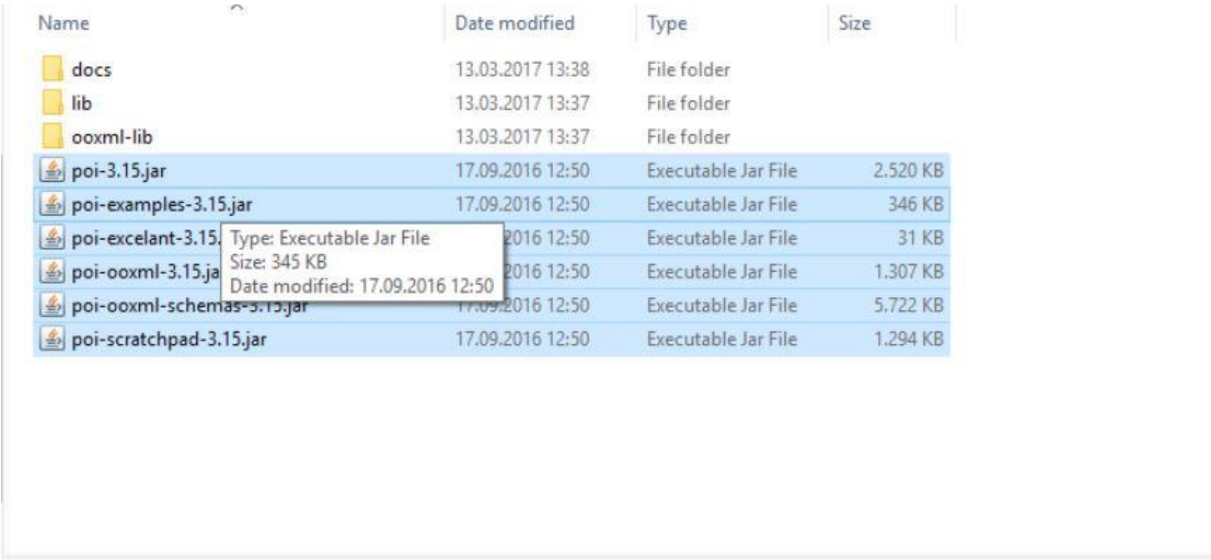
13 build path entries are missing.

JARs and class folders on the build path:

```
commons-codec-1.10.jar - C:\Users\deltasmarttech\Do
commons-collections4-4.1.jar - C:\Users\deltasmarttec
commons-logging-1.2.jar - C:\Users\deltasmarttech\D
curvesapi-1.04.jar - C:\Users\deltasmarttech\Download
junit-4.12.jar - C:\Users\deltasmarttech\Downloads\poi
log4j-1.2.17.jar - C:\Users\deltasmarttech\Downloads\p
poi-3.15.jar - C:\Users\deltasmarttech\Downloads\poi-
poi-examples-3.15.jar - C:\Users\deltasmarttech\Downl
poi-excelant-3.15.jar - C:\Users\deltasmarttech\Downlo
poi-ooxml-3.15.jar - C:\Users\deltasmarttech\Downloa
poi-ooxml-schemas-3.15.jar - C:\Users\deltasmarttech\
poi-scratchpad-3.15.jar - C:\Users\deltasmarttech\Dow
xmlbeans-2.6.0.jar - C:\Users\deltasmarttech\Downloac
JRE System Library [jdk1.8.0_101]
```

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.	2016 12:50	Executable Jar File	31 KB
poi-ooxml-3.15.ja	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

Type: Executable Jar File
Size: 345 KB
Date modified: 17.09.2016 12:50

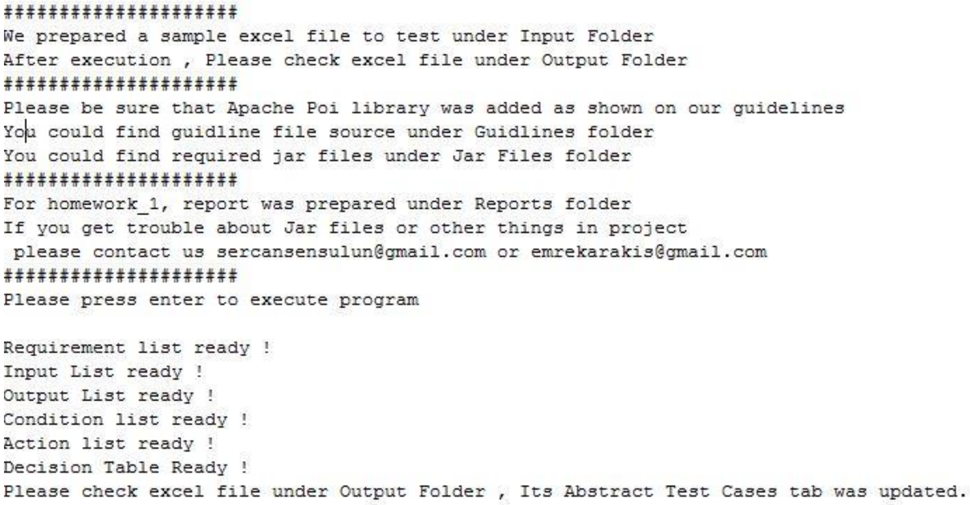
Figure-14

- 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.Program Execution

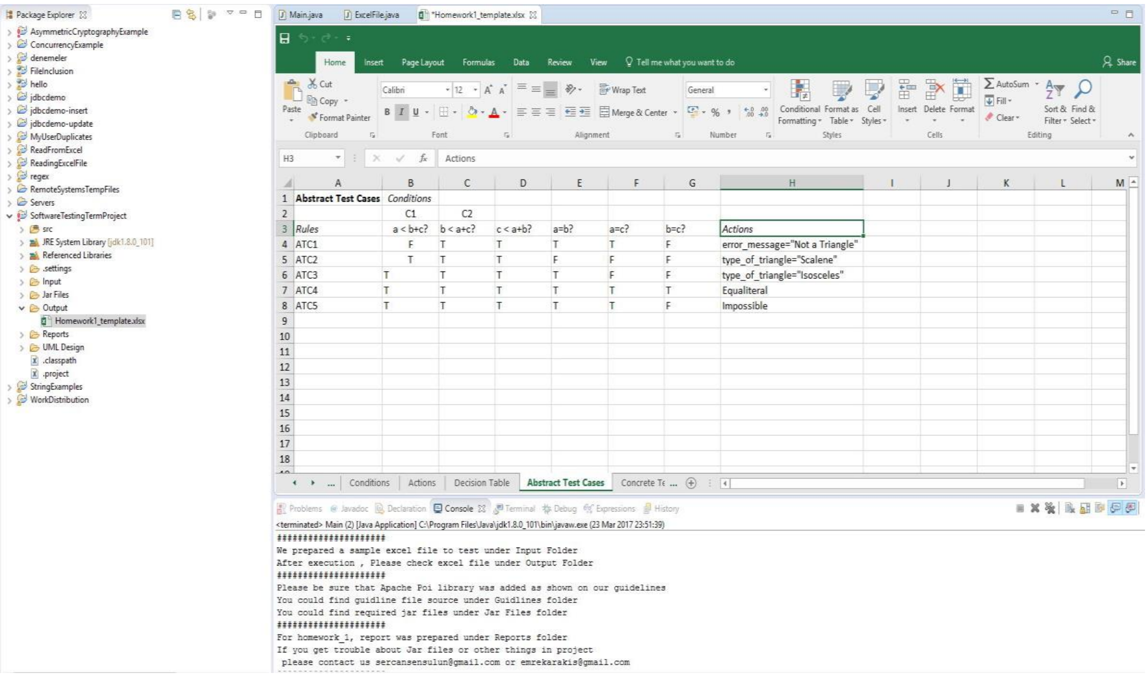
There are two specific folders in the project folder which is called Input and Output.We added a prepared excel file inside the Input folder.The program reads this input excel file and then generates the output excel file as soon as the user presses the “Enter” key.Thus, in order to execute program it is enough to press enter from keyboard. 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 guidline file source under Guidlines 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-15

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.

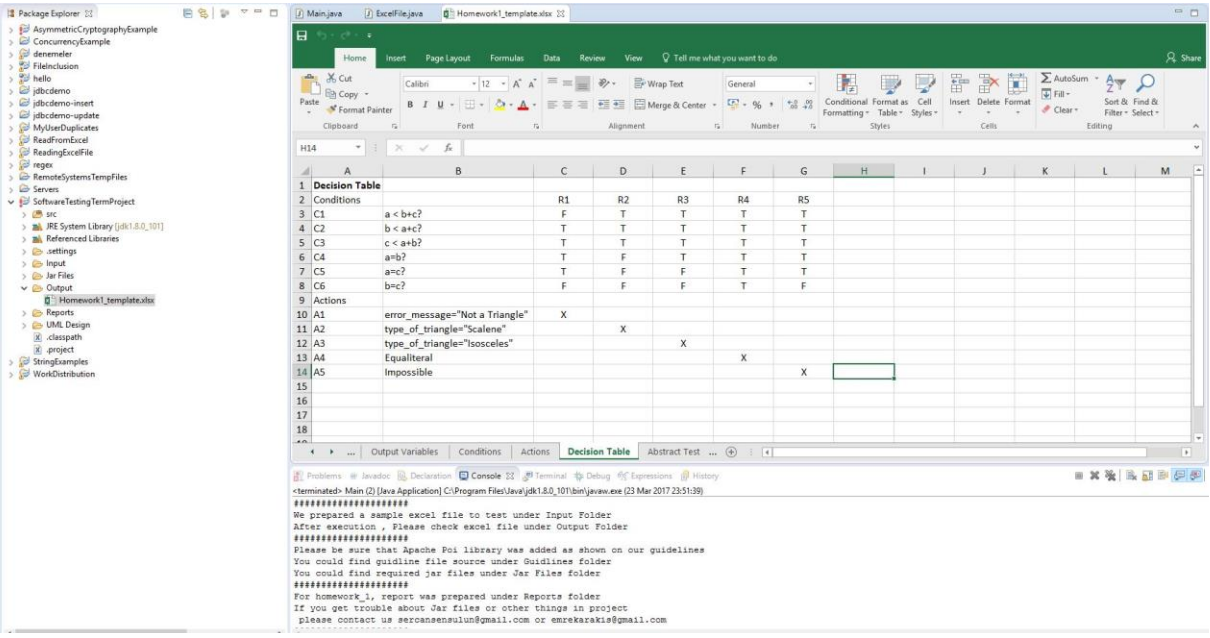


H3 Actions

	A	B	C	D	E	F	G	H
1	Abstract Test Cases	Conditions						
2		C1	C2					
3	Rules	a < b+c?	b < a+c?	c < a+b?	a=b?	a=c?	b=c?	Actions
4	ATC1	F	T	T	T	T	F	error_message="Not a Triangle"
5	ATC2	T	T	T	F	F	F	type_of_triangle="Scalene"
6	ATC3	T	T	T	T	F	F	type_of_triangle="Isosceles"
7	ATC4	T	T	T	T	T	T	Equaliteral
8	ATC5	T	T	T	T	T	F	Impossible

```
<terminated> Main (2) [Java Application] C:\Program Files\Java\jdk1.8.0_101\bin\javaw.exe (23 Mar 2017 23:51:39)
#####################
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 guidline file source under Guidlines 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
```

Figure-16



	A	B	C	D	E	F	G
1	**Decision Table**						
2	Conditions		R1	R2	R3	R4	R5
3	C1	a < b+c?	F	T	T	T	T
4	C2	b < a+c?	T	T	T	T	T
5	C3	c < a+b?	T	T	T	T	T
6	C4	a=b?	T	F	T	T	T
7	C5	a=c?	T	F	F	T	T
8	C6	b=c?	F	F	F	T	F
9	Actions						
10	A1	error_message="Not a Triangle"	X				
11	A2	type_of_triangle="Scalene"		X			
12	A3	type_of_triangle="Isosceles"			X		
13	A4	Equaliteral				X	
14	A5	Impossible					X
15							
16							
17							
18							

```
<terminated> Main (2) [Java Application] C:\Program Files\Java\jdk1.8.0_101\bin\javaw.exe (23 Mar 2017 23:51:39)
#####################
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 guidline file source under Guidlines 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
```

Figure-17

5. Concrete Test Case Implementation

For the concrete Test Cases, we did not implement sat4j.However, in the abstract test cases,we have previously generated 11 test cases, thus it is such a scenerio that we can optimize the test cases by filtering with the satisfiability condition.In some cases, abtstract cases are written then it does not reflect the actual point of view. Such inconsistencies would not assessed as a test case because these cases are impossible to consider, by that way we would have a little bit smaller list of test set.In homework 2, we add an additional implementation of Don’t Care(X) operations.When the user is asked whether a specific condition is satisfiable or not, then he/she answers the question for eleven test cases as Y/N or y/n.Then, concrete test cases are generated in the excel file within the output file in project.

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 implenment abstract test case model.