Step-by-Step Procedures for answering JPLAS Code Writing Problem

1. Firstly, please **extract** the compressed Zip file that we gave with name like “OffJPLAS-task ID”, “BASIC”, “DATASTRUCTURE” and so on.
2. **Create** a new java project with name "OffJPLAS-task ID" (e.g., OffJPLAS-01) in Eclipse.
3. **Copy** all of contents from the extracted folder to src of the created java project.
4. Then, please **open each package** in src and you can see that there are **two java classes** like classname.java and classnameTest.java for each problem. You must **only write** the appropriate source codes **as you like** in classname.java to satisfy the test codes written in classnameTest.java. However, you don’t need to write the main function [public static void main (String args[]) in the source code. The example processes can be seen in figure 1, figure 2 and figure 3.

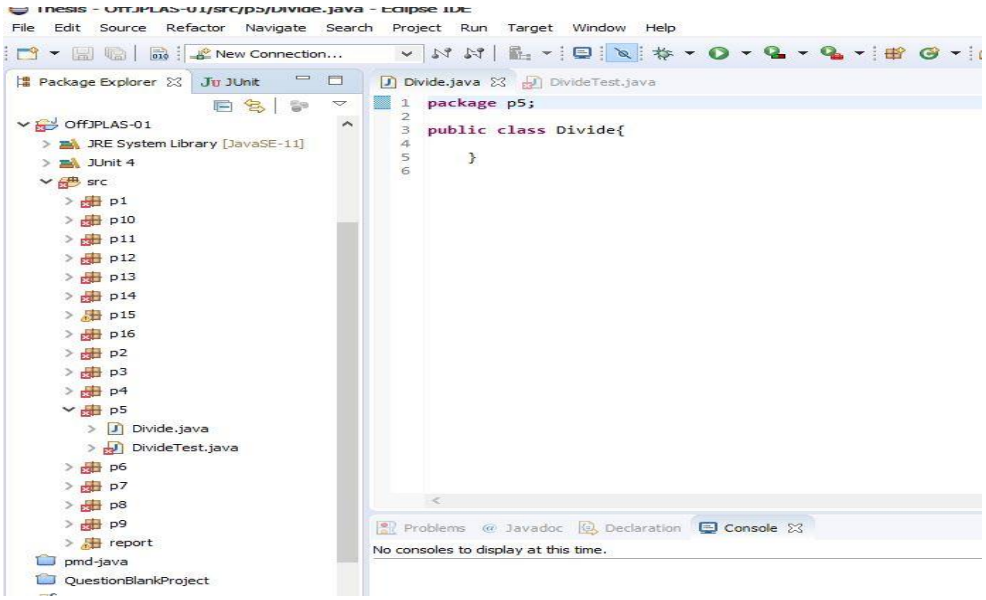

Fig1. Original java class before writing Source code

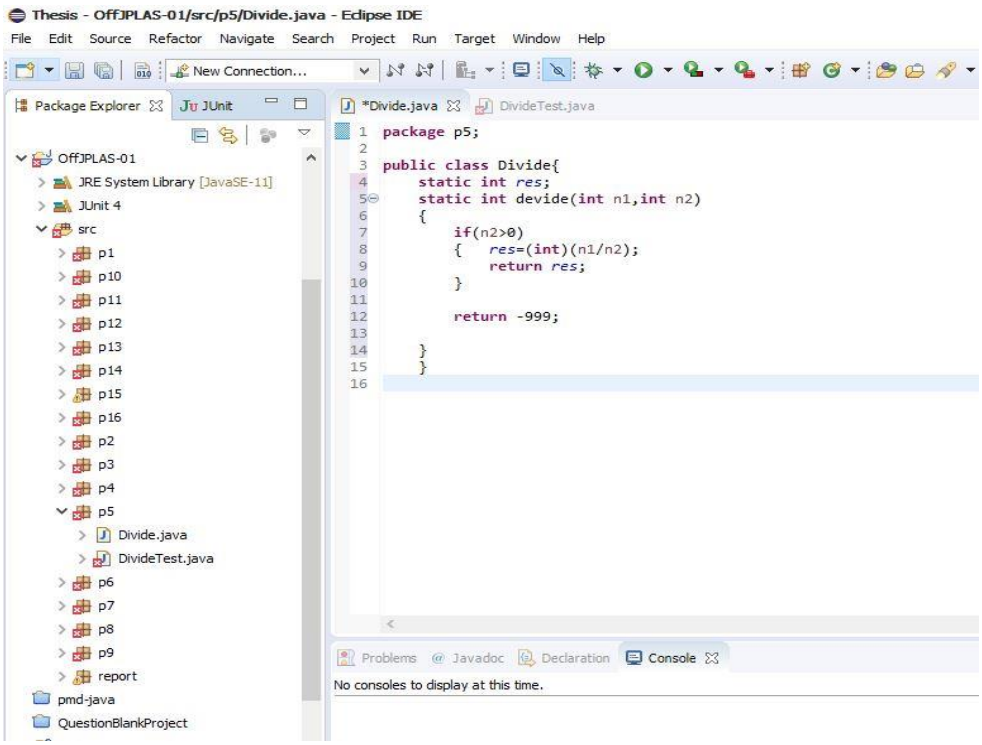

Fig2. Modified java class after writing Source code

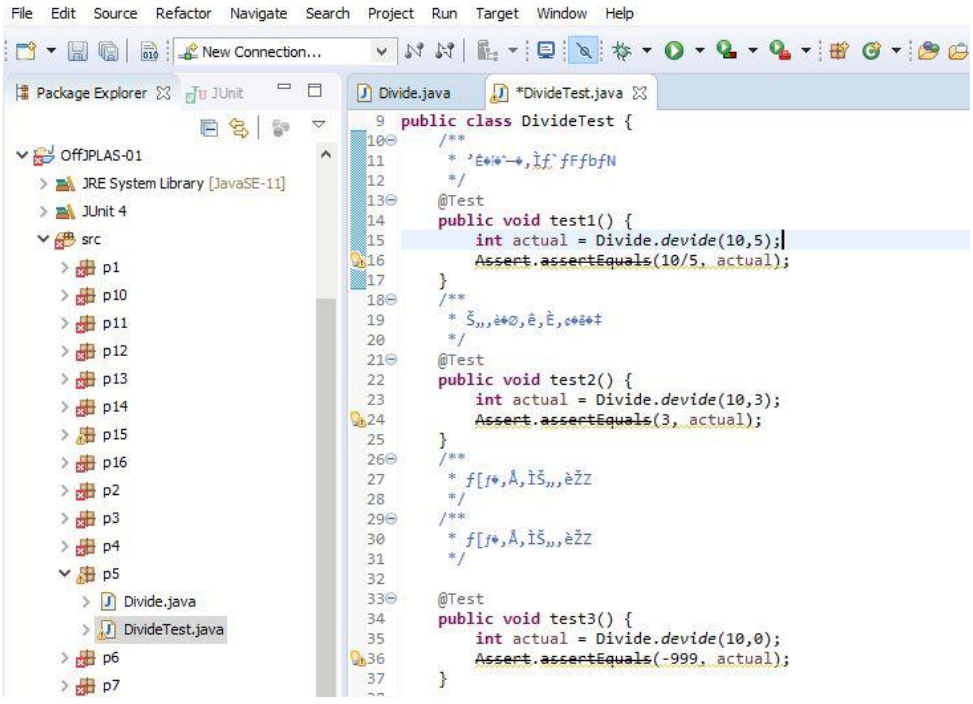

Fig3. Teacher-defined Java class for Test code

5. After this, to solve this code writing problems, you must **import JUnit** Library. To import the JUnit, right-click on the [Off JPLAS-task ID] and click[Build Path] => [Add Libraries] => [JUnit4] => Finish. The example can be seen in figure 4.

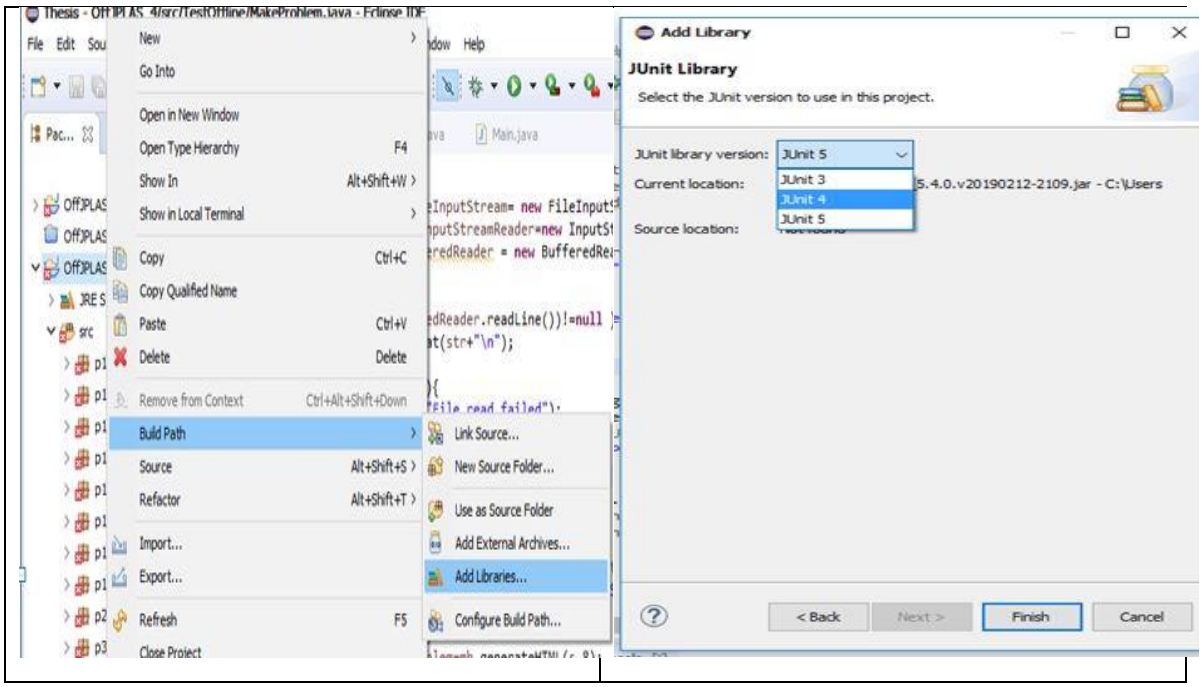

Fig.4 Importing JUnit Library

6. Then, students **can check** by themselves whether or not the source codes satisfy the test codes by right-click on the classnameTest.java and then go to [Run as] => [JUnit Test]. JUnit will show your test result in eclipse IDE. "Green bar" means your source code is correct and "Brown bar" means your code includes failure or error. You can find errors in failure trace. This processes can be seen in figure 5 and figure 6.

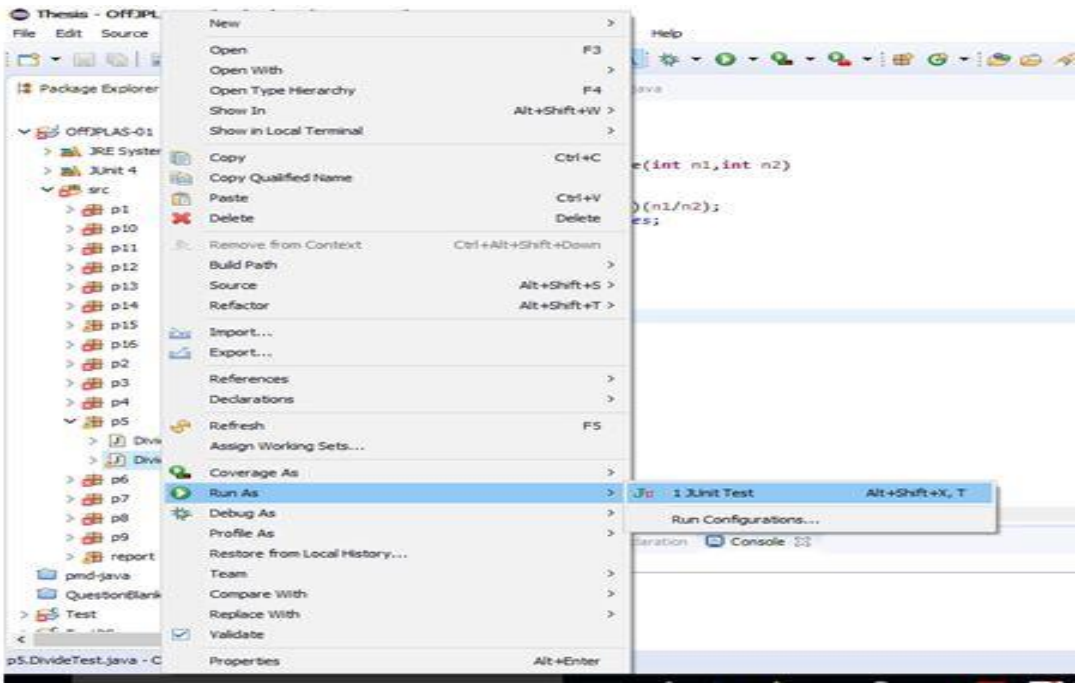

Fig.5 Testing with JUnit

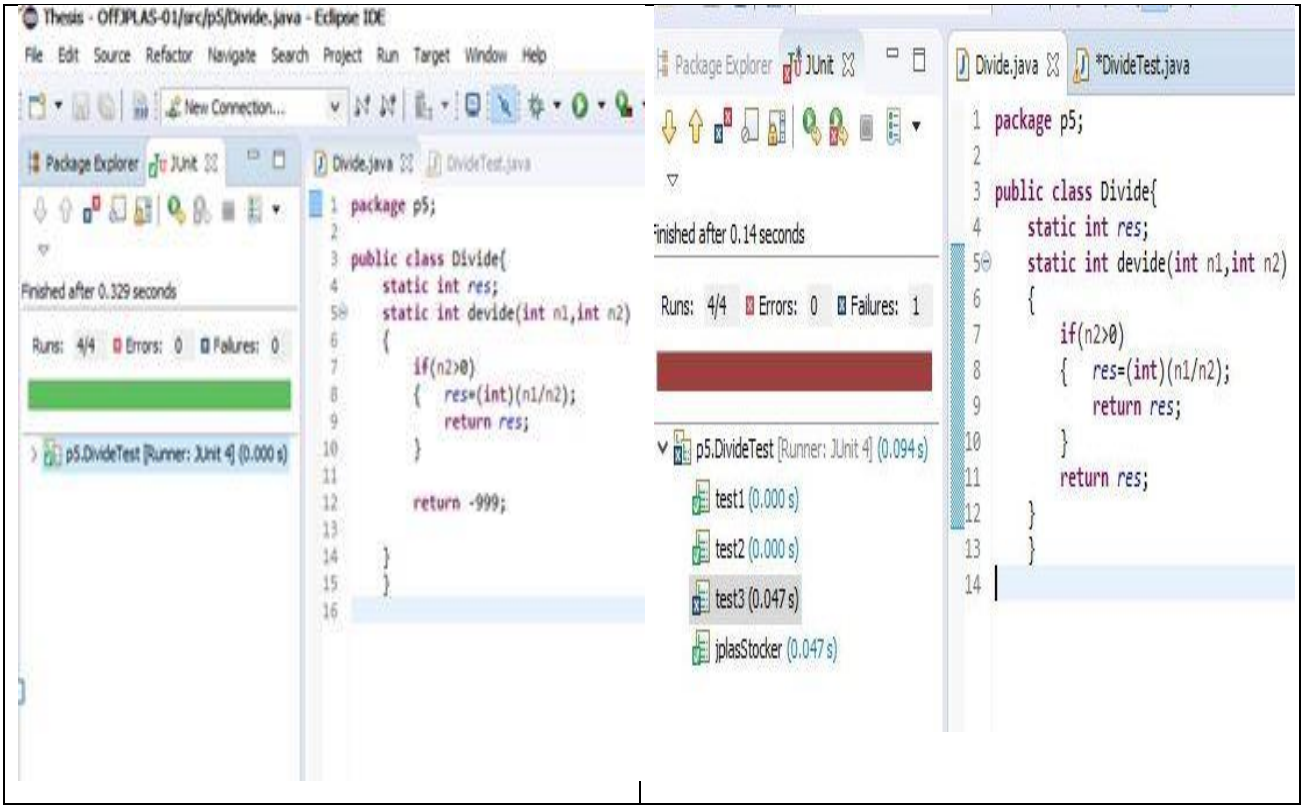

Fig.6 Comparing Testing Results from JUnit

7. Students **can repeat** the verification of test code with JUnit every time they modify the source code, until the test is successful.

8. After solving all of problems from each package, to **leave a test log** ,please **right-click** on this current java project and go to "Export" => "Ant Buildfiles" >> “Next” >> select the project folder OffJPLAS-task ID (e.g., OffJPLAS-01) and click Finish. This can be seen in figure 7 and figure 8.

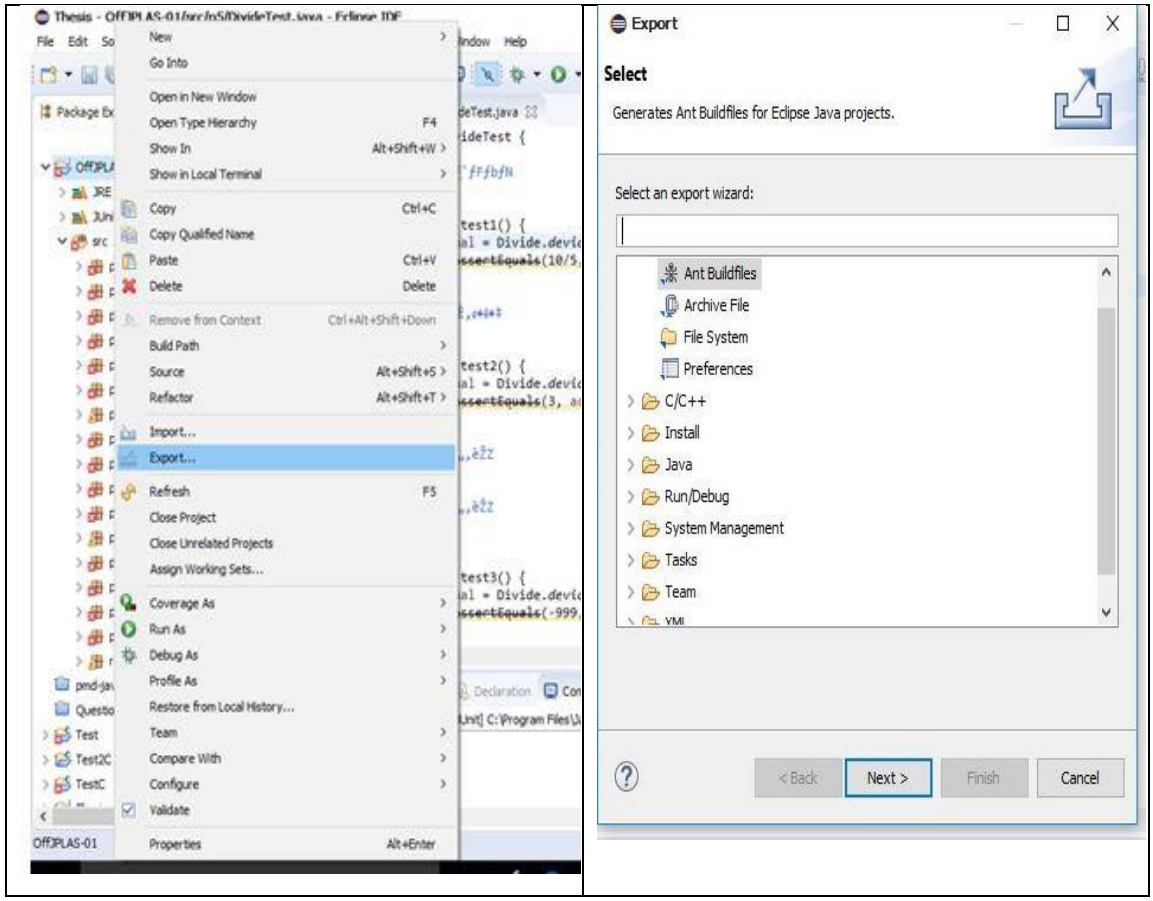

Fig.7 Steps for Generating Build file

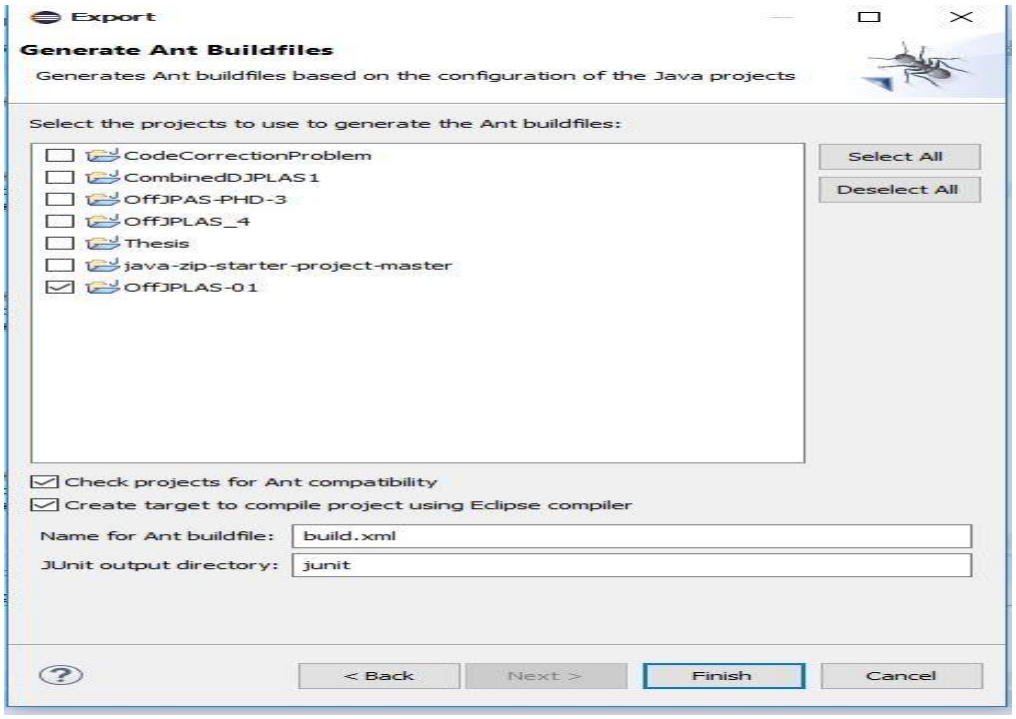

Fig.8 Last Step for Generating Build file

9. Also, **create a folder** named "junit" by yourself under "OffJPLAS-task ID" as shown in following figure 9.

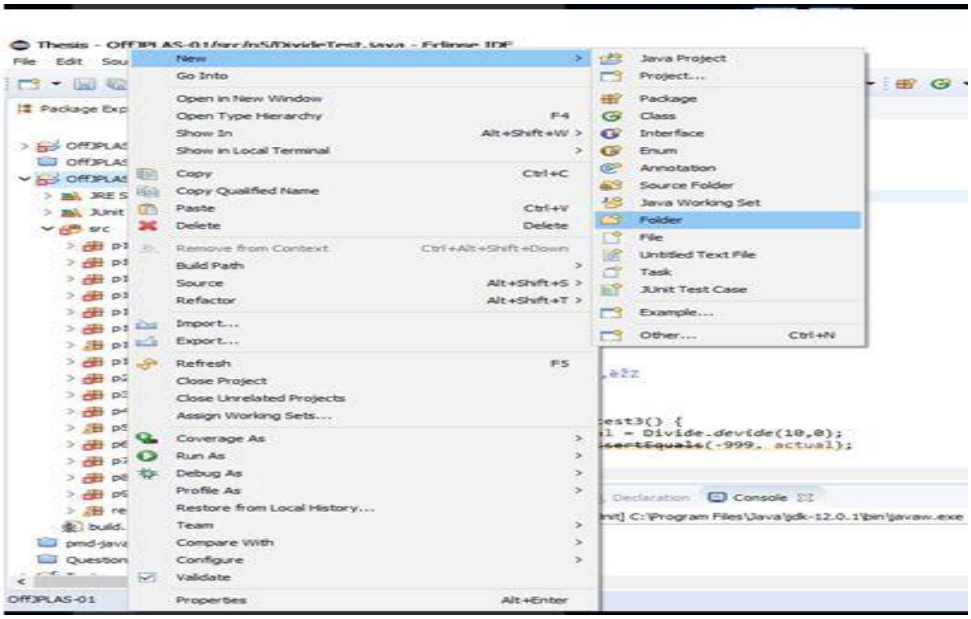

Fig.9 Folder Creation

10. Then, select the **resulted "build.xml"** under the project and right-click on **"build.xml" => "Run as" => "2. Ant build"**. In the "Edit configuration and launch" box, select three types of targets such as "**build [default]",** "**all test classes** to execute (e.g., DivideTest, , MaxItemTest,…)", and "**junit report**". Then, click "apply" and "run". The XML file of the test result of each problem will be outputted to the "junit" folder.
11. After that, you will see the "**BUILD SUCCESSFUL**" information in the console as in the following figure 10.

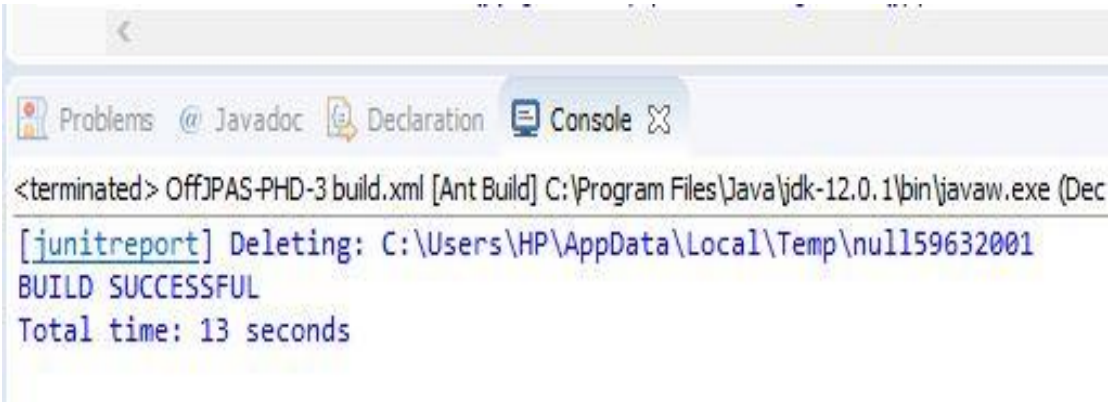

Fig.10 Generation of XML file of the test result of each problem

※ Correspondence when it becomes BUILD FAILED, add tool.jar

「Window」-「Preference」-「Ant」-「Runtime」-「Global Entries」-「Add External JARs」-「C:\Program Files\Java\jdk-****\lib\tool.jar」

12. Then, select **report.java** in the **report folder** under src and run it. When you execute "report.java" of the assignment project, enter the student number (8 digits) in the submission input window and click the "OK" button (see in figure 11).

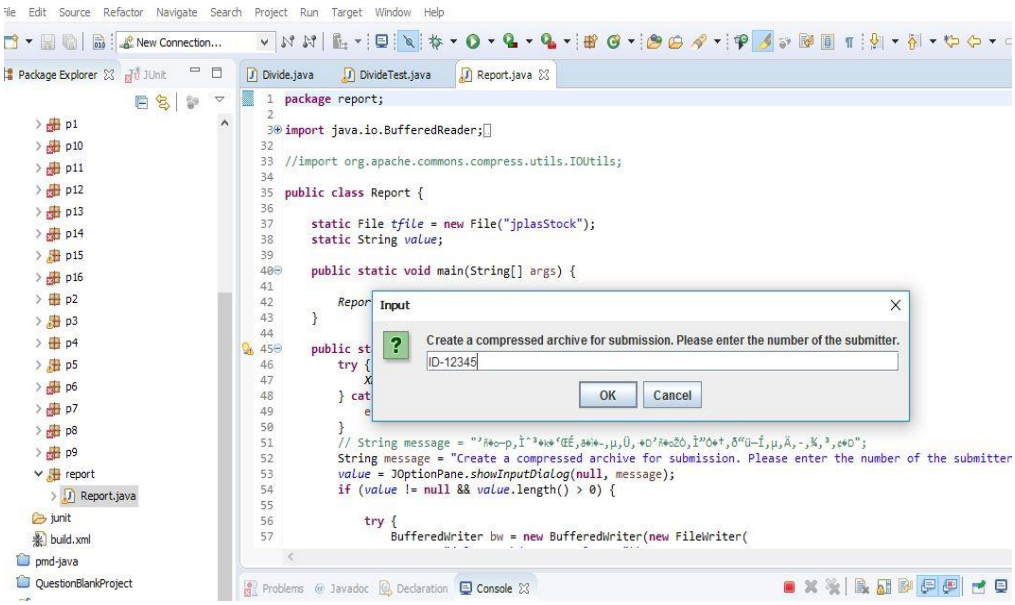

Fig.11 Submission Input Window

13. Then, the “Save” screen will be displayed. Enter the student number again, press the “Save” button and then the answer data is saved as the Zip folder “xxxxxxxx.zip” in your computer (see in figure 12).

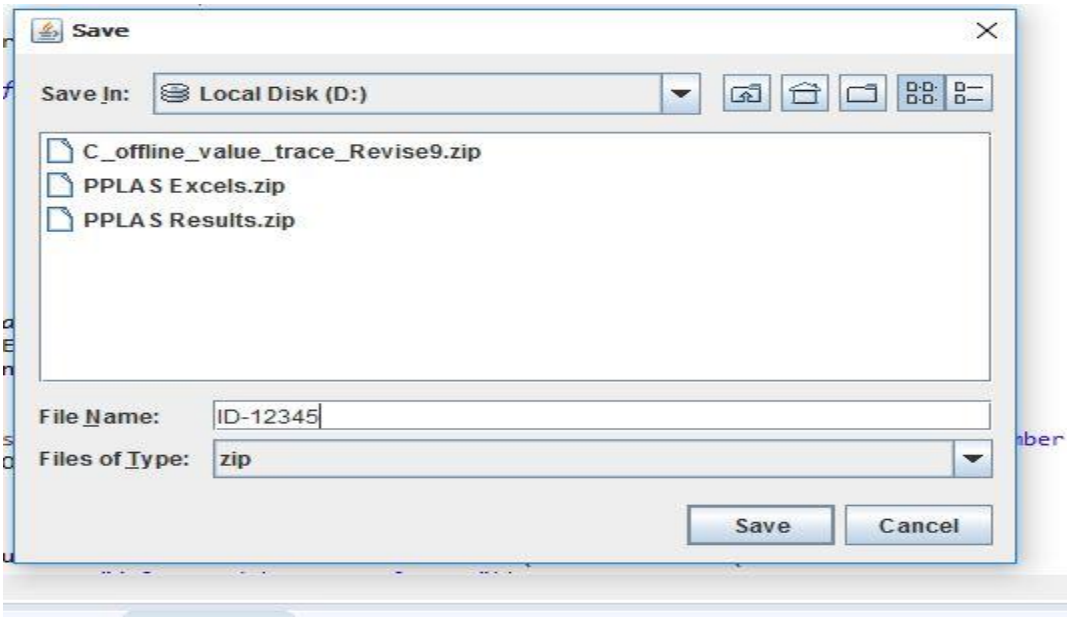

Fig.12 Save Window

14. Submit the answer file “xxxxxxxx.zip” to the teacher using a USB memory or email. Finally, you have successfully solved code writing problems.