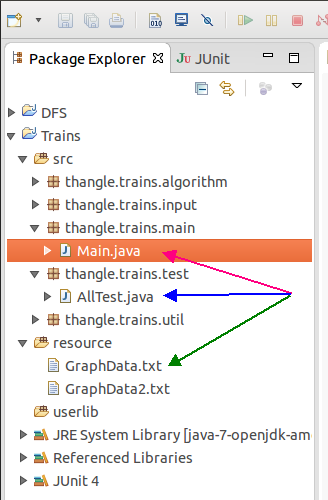
1. How to run the “Trains” application
2. **Setup environment**

* You need the Eclipse Java IDE to import the project (prefer using **Eclipse** **Mars** version, *http://www.eclipse.org*).
* Java JDK 1.7 or 1.8 to build and run the project.

1. **How to run**

* Import project **Trains** to Eclipse and select “*Package Explorer*” as image below.

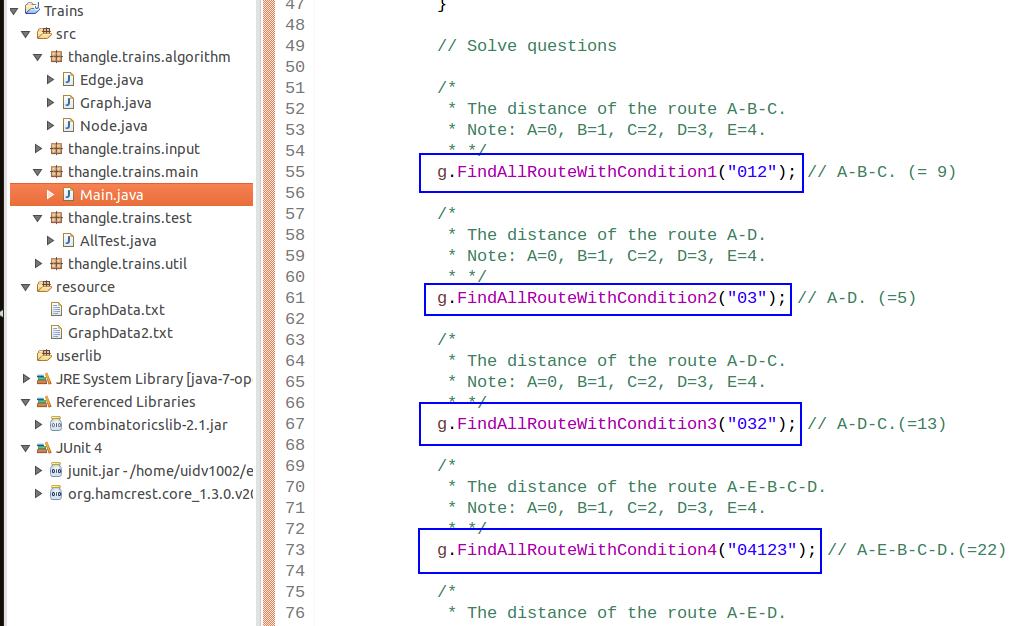


* In **source** folder, select class **main.java** and run it as “*Java Application*” to run the program or go to class **AllTest.java** to run unit test (Junit) as above image.
* In **resource** folder, open file **GraphData.txt** to change input data green arrow in above image.
* The format in file **GraphData.txt** needs follow as in example. To add a new node, the new node name is only letters from “A” to “Z” and *increase one by one*.

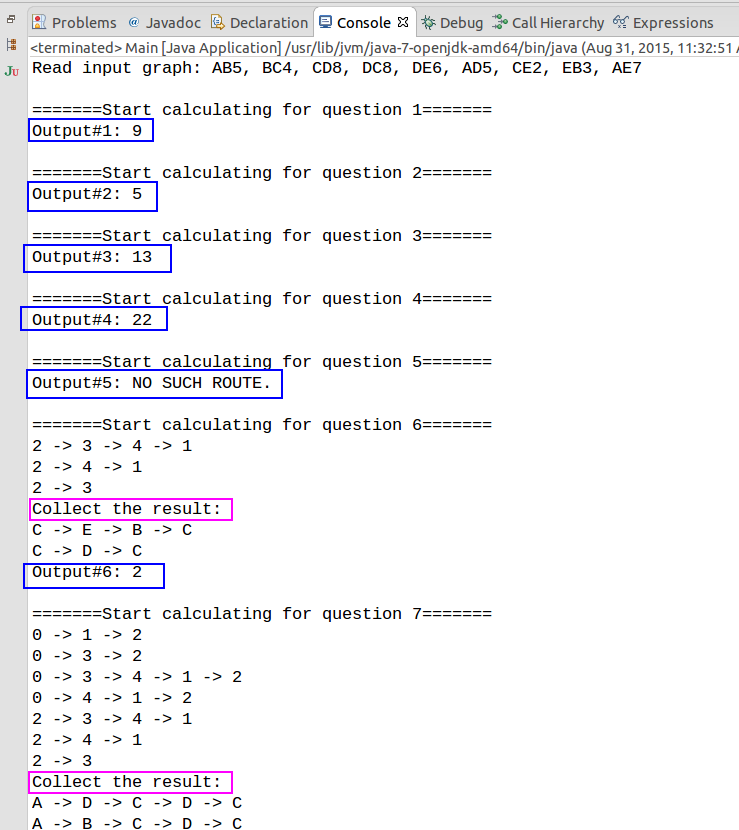
For example: current graph is: **AB5, BC4, CD8, DC8, DE6, AD5, CE2, EB3, AE7**

To add new node, then the new letter should be used is “**F**”. Eg. **AB5, BC4, CD8, DC8, DE6, AD5, CE2, EB3, AE7, EA7, AF6, FD15** (check file **GraphData2.txt** for reference).

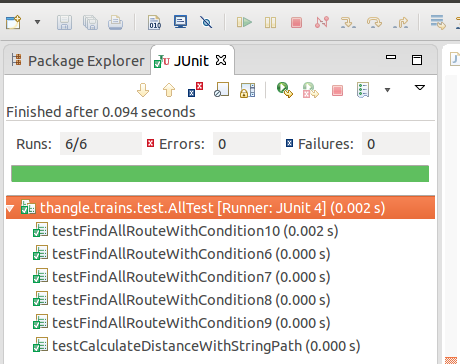
* To change the question, go to class **main** and change arguments of methods as image below



* The results are printed out to Eclipse console (as image below).



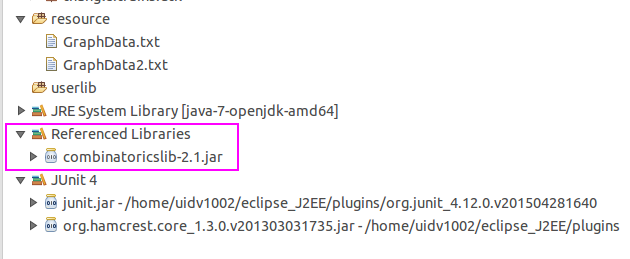
* Unit test result (not do all test cases ☺ )



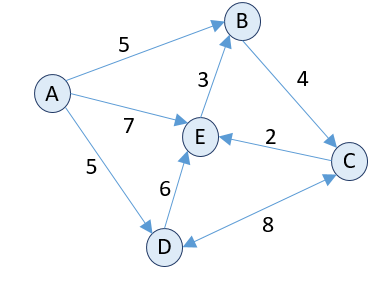
1. Design overview
2. **Software structure**
3. Software includes 5 source code packages as image below.

* **thangle.trains.algorithm**: implementation of algorithms as Depth First Search, Dijkstra, combinatorial. For combinatorial algorithm, I used library at: <https://code.google.com/p/combinatoricslib/>
* **thangle.trains.input:** implementation of reading input text file and process input data.
* **thangle.trains.main:** implementation of main class. Program starts from here.
* **thangle.trains.util:** implementation of some utilities.
* **thangle.trains.test:** implementation of unit test using Junit.

1. The **resoure** folder to store resource files (text files as input data).
2. A user library (*combinatoricslib-2.1.jar*) for combinatorial algorithm is put into the folder */Trains/ userlib* as image below.



1. **The solutions for questions**



* For question 1 to 5: just simple implement a function to calculate the distance. For example, the distance from A-B-C = distance(A-B) + distance(B-C).
* For question 6, 7, 10: to use *depth first search* and *combinatorial* algorithm for calculation.
* For question 8, 9: to use *Dijkstra* and *combinatorial* algorithm for calculation.

1. **Class diagram**

* Below is the class diagram of the program (draw by IBM Rational Rhapsody).

