

Ethic | src | test | java | Unit_Test_I

Project | Ethic | Documents/Ethic | .idea | Development_Process | console_execution.pdf | README.md | sample_execution_via_Int | unit_test_execution.pdf | written_work.jpg | InputFiles | sample.txt | sample_II.txt | src | main | java | 100% classes, 92% methods | Ethic class | Graph 100% method | Graph\$Node.class | Graph.class | sample.txt | resources | test | java | Unit_Test_I | target | pom.xml | README.md | External Libraries | < 1.8 -> /Library/Java/JavaVi | ant-javafx.jar library root | charsets.jar library root | cldrdata.jar library root

Graph.java | Ethic.java | README.md | Development_Process/README.md | sample.txt | Unit_Test_Java | Unit_Test_I | sample.txt

Coverage: 100% classes, 92% lines covered in 'all classes in scope'

Unit_Test_I

Element | Class, % | Method, % | Line, %

apple | com | java | javafx | jdk | META-INF | netscape | oracle | org | resources | sun | toolbarButton... | Ethic | Graph | 100% (1/1) | 88% (8/9) | 72% (45/62) | 100% (2/2) | 100% (26/26) | 98% (194/1...)

Ethic class

Unit_Test_Java | Unit_Test_I | sample.txt

Development_Process/README.md | sample.txt

import java.util.*;
import org.junit.Test;
import org.junit.Assert.*;
import static org.junit.Assert.assertEquals;

public class Unit_Test_I {
 Ethic createExecutable(String routes){
 Ethic ethic = new Ethic();
 ethic.createGraph(Arrays.asList(routes.split(", ")));
 return ethic;
 }
}

@Test
public void sample_input() {
 Ethic sample = createExecutable(routes: "AB5,BC4,CD8,DE6,AD5,CE2,EB3,AE7");
 assertEquals(expected: "9", sample.getDistanceAlongRoute("ABC"));
 assertEquals(expected: "2", sample.getRoutesWithLimitedStops(start: "C", end: "C", maxNumberOfStops: 3));
 assertEquals(expected: "7", sample.getRoutesWithLimitedDistance(start: "C", end: "C", maxDistance: 30));
 assertEquals(expected: "9", sample.getShortestRouteBetweenTwoTowns("A", "C"));

/* Component testing (Testing each function using the graph provided
in the problem description */

@Test //< Question One
public void distanceAlongRoute() {
 Ethic sample = createExecutable(routes: "AB5,BC4,CD8,DE6,AD5,CE2,EB3,AE7");
 assertEquals(expected: "26", sample.getDistanceAlongRoute("ABCDEB"));
 assertEquals(expected: "24", sample.getDistanceAlongRoute("DEBCEBCE"));
 assertEquals(expected: "18", sample.getDistanceAlongRoute("CDCFE"));
 assertEquals(expected: "NO SUCH ROUTE", sample.getDistanceAlongRoute("AC"));
 assertEquals(expected: "NO SUCH ROUTE", sample.getDistanceAlongRoute("BA"));

Run: Unit_Test_I.ShortestRoutesBetweenTwoTowns | Unit_Test_I

Tests passed: 6 of 6 tests – 23 ms

The distance of the route ABC.
The number of trips starting at C and ending at C with a maximum of 3 stops
The number of different routes from C to C with a distance of less than 30
The length of the shortest route (in terms of distance to travel) from A to C
Graph Has Been Initialized

23 ms
13 ms
3 ms
3 ms
2
1 ms
1 ms
2 ms

shortestRoutesBetweenTwoTowns
routesWithLimitedDistance
tokenize
distanceAlongRoute
routesWithLimitedStops
sample_input

4: Run | TODO | Problems | Debug | Terminal | Build

Tests passed: 6 (moments ago)

Event Log | Coverage

421 chars, 7 line breaks | 13:1 | LF | UTF-8 | 4 spaces