

Ethic > src > test > java > Unit_Test_I

Project

Ethic ~\Documents\Ethic

Development_Process

console_execution.pdf

README.md

sample_execution_via_IntelliJ.pdf

sample_III.pdf

sample_III_execution.pdf

unit_test.pdf

written_work.jpg

written_work.pdf

InputFiles

sample.txt

sample_II.txt

sample_III.txt

src

main

java 100% classes, 92% lines covered

Ethic 88% methods, 72% lines covered

Ethic.class

Graph 100% methods, 99% lines covered

Graph\$Node.class

Graph.class

resources

test

java

Unit_Test_I

Unit_Test_II

target

Ethic.iml

pom.xml

README.md

External Libraries

< 1.8 > /Library/Java/JavaVirtualMachines

ant-javafx.jar library root

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

```
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( regex: ",")));
        return ethic;
    }

    @Test
    public void sample_input() {
        Ethic sample = createExecutable( routes: "AB5,BC4,CD8,DC8,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: 36));
        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,DC8,DE6,AD5,CE2,EB3,AE7");
        assertEquals( expected: "26", sample.getDistanceAlongRoute("ABCDEB"));
        assertEquals( expected: "24", sample.getDistanceAlongRoute("DEBCEBCE"));
        assertEquals( expected: "18", sample.getDistanceAlongRoute("CDCE"));
        assertEquals( expected: "NO SUCH ROUTE", sample.getDistanceAlongRoute("AC"));
        assertEquals( expected: "NO SUCH ROUTE", sample.getDistanceAlongRoute("BA"));
    }
}
```

Coverage: All in Ethic

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

Element	Class, %	Method, %	Line, %
apple			
com			
java			
javafx			
javax			
jdk			
META-...			
netasca...			
oracle			
org			
resour...			
sun			
toolbar...			
Ethic	100% (1/1)	88% (8/9)	72% (45/62)
Graph	100% (2/2)	100% (26/26)	99% (198/200)

Run: All in Ethic

Tests passed: 7 of 7 tests - 25 ms

<default package> 25 ms

Unit_Test_I 21 ms

Unit_Test_II 4 ms

sample_input_III 4 ms

/Library/Java/JavaVirtualMachines/jdk1.8.0_271.jdk/Contents/Home/bin/java ...

IntelliJ IDEA coverage runner

sampling ...

include patterns:

exclude patterns:

Graph Has Been Initialized

Graph Has Been Initialized

Graph Has Been Initialized

The distance of the route ABC.

4: Run

6: Problems

TODO

Terminal

Build

Event Log

Tests passed: 7 (moments ago)

44:1 LF UTF-8 4 spaces