

UNIVERSITÉ DE LA ROCHELLE



Master ICONE 1ère année

TP Test

Réaliser par :
Mohammed BENAOU
Walid CHERKAOU

2017-2018

1 Niveau 1

L'exécution de la classe SommeTest avec JUnit ça passe avec succès :

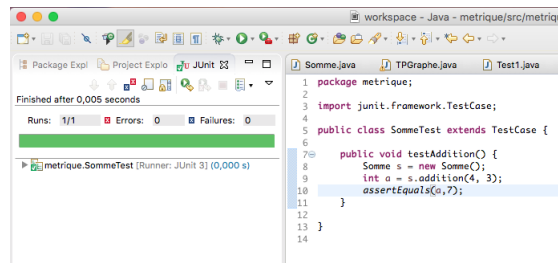
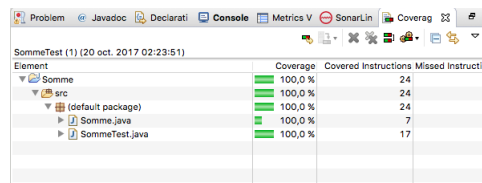


FIGURE 1 – La classe SommeTest

-Calcule de la couverture de test de la classe Somme :



Element	Coverage	Covered Instructions	Missed Instructions
SommeTest (1) (20 oct. 2017 02:23:51)			
▼ Somme	100,0 %	24	
▼ src	100,0 %	24	
▼ (default package)	100,0 %	24	
▼ Somme.java	100,0 %	7	
▼ SommeTest.java	100,0 %	17	

FIGURE 2 – La couverture

2 Niveau 2

2.1 La classe EntreeErrone

```
1 package personnePack;
2
3 public class EntreeErrone extends Exception {
4
5
6
7     public EntreeErrone(String message, Throwable cause) {
8         super(message, cause);
9         // TODO Auto-generated constructor stub
10    }
11
12    public EntreeErrone(String message) {
13        super(message);
14        // TODO Auto-generated constructor stub
15    }
16 }
```

2.2 La fonction setNumTelephone

```
1 public void setNumTelephone(String numTelephone) throws EntreeErrone{
2     if(numTelephone.length()>8)
3         throw new EntreeErrone("le numero est inferieur a 8");
4     else
5     {
6         this.numTelephone=numTelephone;
7     }
8     boolean trouver=false;
9     for(int i=1;i<=1;i++)
10    {
11        String val="0"+i;
12        String twoElements=numTelephone.charAt(0)+numTelephone.charAt(1)
13            +"";
14        if(twoElements==(val)){
15            trouver=true;
16        }
17    }
18    if(trouver == false)
19    {
20        throw new EntreeErrone("le numero de telephone doit etre entre 1
21            et 7");
22    }
23    else
24    {
25        this.numTelephone=numTelephone;
26    }
27 }
```

2.3 Test de SetNumTelephone

```
1 public void testSetNumTelephone() {
2     try{
3
4         P.setNumTelephone("03000");
5         assertTrue(P.numTelephone.compareTo("03000")==0);
6     }
7     catch(EntreeErrone ex){
8         // fail(ex.getMessage());
9     }
10
11 }
```

2.4 Test Node

```
1 package Metrique;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 import org.junit.jupiter.api.Test;
6
7 class NodeTest {
```

```

8     Node n = new Node(1,"test");
9     @Test
10    void testId() {
11        //fail("Not yet implemented");
12        //System.out.println(n.id());
13        assertTrue(n.id()==1);
14        assertFalse(n.id()!=1);
15    }
16
17
18    @Test
19    void testSetColor() {
20        //fail("Not yet implemented");
21        n.setColor("black");
22        assertTrue(n.color()=="black");
23        assertFalse(n.color()!="black");
24    }
25
26
27
28    @Test
29    void testSetShape() {
30        n.setShape("ok");
31        assertTrue(n.shape()=="ok");
32        assertFalse(n.shape()!="ok");
33    }
34
35
36
37    @Test
38    void testSetLabel() {
39        //fail("Not yet implemented");
40        n.setLabel("label1");
41        assertTrue(n.label()=="label1");
42        assertFalse(n.label()!="label1");
43    }
44
45    @Test
46    void testAddSucc() {
47        //fail("Not yet implemented");
48        assertTrue(n.addSucc(new Arc(new Node(2),new Node(3)))==true);
49    }
50
51    @Test
52    void testContainsSucc() {
53        //fail("Not yet implemented");
54        assertFalse(n.containsSucc(new Arc(new Node(1),new Node(5)))==true);
55    }
56
57
58    @Test
59    void testRemoveSucc() {
60        assertFalse(n.removeSucc((new Arc(new Node(2),new Node(3))))==true);
61    }

```

```

62
63  @Test
64  void testAddPred() {
65      assertTrue(n.addPred((new Arc(new Node(2),new Node(3))))==true);
66  }
67
68  @Test
69  void testContainsPred() {
70
71      assertTrue(n.containsPred((new Arc(new Node(1),new Node(2))))==false)
72          ;
73  }
74
75  @Test
76  void testRemovePred() {
77      assertFalse(n.removePred(((new Arc(new Node(2),new Node(3))))==true)
78          ;
79  }
80
81  @Test
82  void testCompareTo() {
83      //fail("Not yet implemented");
84      assertTrue(n.compareTo(new Node (1))==0);
85  }
86
87  @Test
88  void testCompareToFalse() {
89      //fail("Not yet implemented");
90      assertFalse(n.compareTo(new Node (1))==-1);
91  }
92  }

```

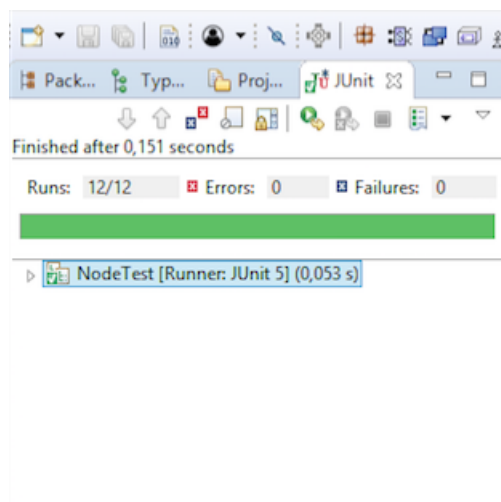


FIGURE 3 – JUnit

Num	classe	variable d'entrée	méthode	résultat
1	NodeTest	id=1 label="test"	testId()	True
2	NodeTest	couleur="black"	testsetcolor()	True
3	NodeTest	shape="ok"	testShape()	True
4	NodeTest	label="label1"	testSetLabel()	True
5	NodeTest	Arc	testAddsucc()	True
6	NodeTest	Arc	testContainsSucc()	True
7	NodeTest	Arc	testRemoveSucc()	True

TABLE 1 –

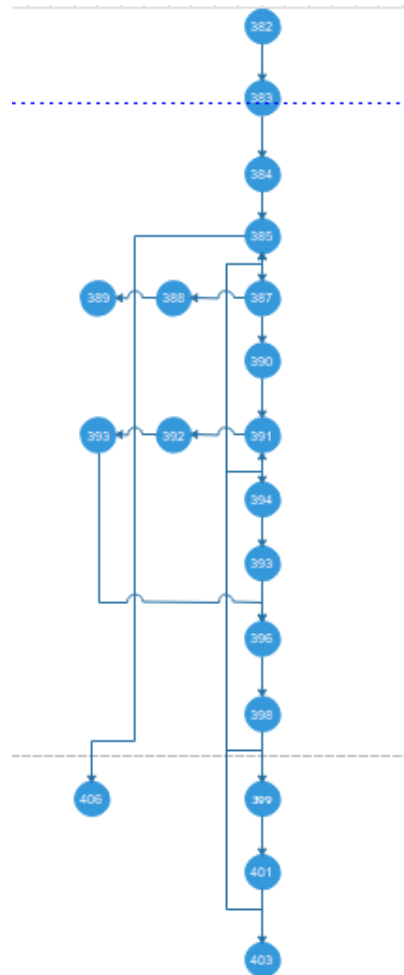


FIGURE 4 – Graphe