Faculté des Sciences Mathématiques, Physiques et Naturelles de Tunis
Département des Sciences Informatique
Filière : 4eme Ingénieur en Informatique

Qualité Logiciel

Mini-projet:Test code DS 2019

Proposé par : Taha Bennani

Réalisé par : Ahmed Zayati

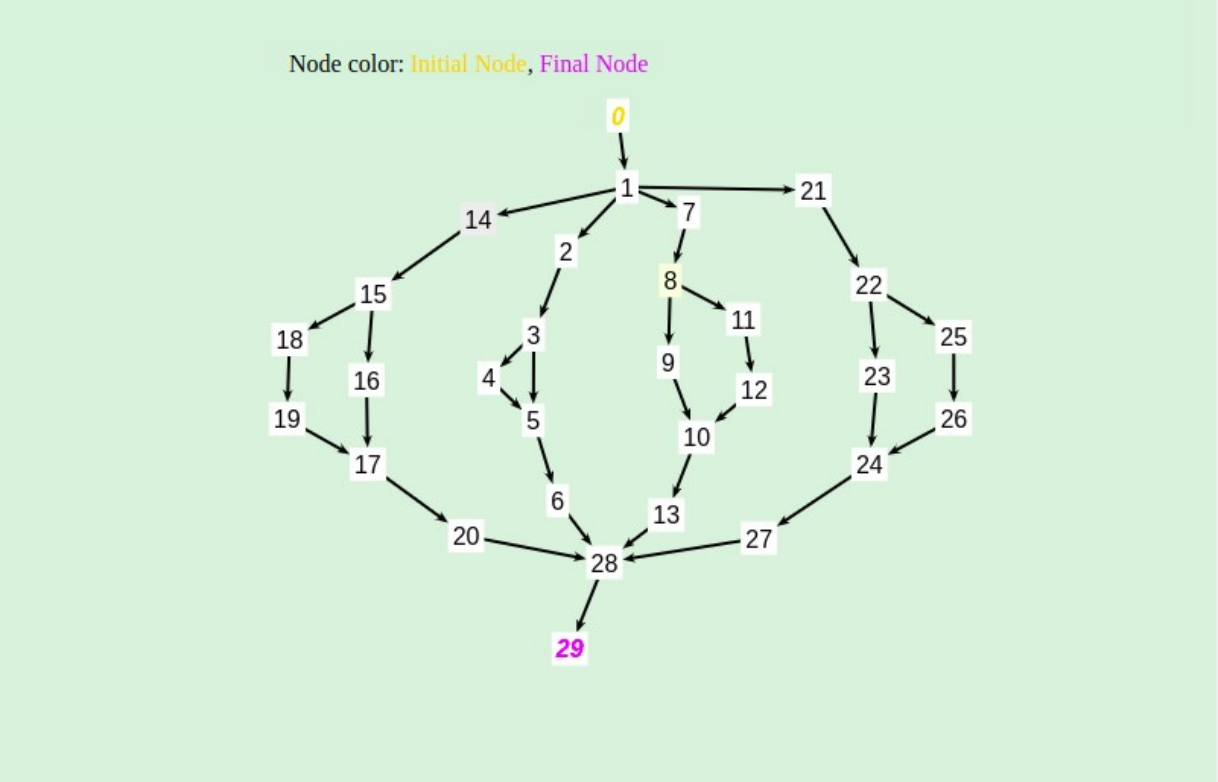
Introduction:

Au cours de ce mini-projet et à l'aide de "Graph Coverage Web Application" j'ai dessiné le graphe de flux de controle puis j'ai déterminé les tests requirement et les tests path pour node coverage,edge coverage ,edge pair coverage et prime path coverage.

On plus j'ai déterminé le taux de couverture pour les cas de test proposés dans le DS et j'ai modifié le code de FSM et les cas de test

suivant les questions de DS.(modification de doesReachesFinalState,ajout de doesNotReachesState ...).

Question 1:Graphe du flux de controle du code source fonctionnel.



Question 2: Node coverage,test requirement et test paths.

30 requirements are needed for Nodes
[0]
[1]
[14]
[7]
[2]
[21]
[3]
[4]
[5]
[6]
[28]
[29]
[8]
[9]
[10]
[11]
[12]
[13]
[22]
[23]
[24]
[25]
[26]
[27]
[15]
[16]
[17]
[18]
[19]
[20]

7 test paths are needed for Node Coverage
[0,1,2,3,4,5,6,28,29]
[0,1,21,22,23,24,27,28,29]
[0,1,14,15,16,17,20,28,29]
[0,1,7,8,9,10,13,28,29]
[0,1,21,22,25,26,24,27,28,29]
[0,1,14,15,18,19,17,20,28,29]
[0,1,7,8,11,12,10,13,28,29]

Question 3: Edge coverage,test requirement et test paths.

36 requirements are needed for Edges
[0,1]
[1,14]
[1,7]
[1,2]
[1,21]
[2,3]
[3,4]
[3,5]
[4,5]
[5,6]
[6,28]
[28,29]
[7,8]
[8,9]
[9,10]
[8,11]
[11,12]
[12,10]
[10,13]
[13,28]
[21,22]
[22,23]
[23,24]
[22,25]
[25,26]
[26,24]
[24,27]
[27,28]
[14,15]
[15,16]
[16,17]
[15,18]
[18,19]
[19,17]
[17,20]
[20,28]

8 test paths are needed for Edge Coverage
[0,1,2,3,5,6,28,29]
[0,1,2,3,4,5,6,28,29]
[0,1,21,22,23,24,27,28,29]
[0,1,14,15,16,17,20,28,29]
[0,1,7,8,9,10,13,28,29]
[0,1,21,22,25,26,24,27,28,29]
[0,1,14,15,18,19,17,20,28,29]
[0,1,7,8,11,12,10,13,28,29]

Edge pair coverage ,test requirement et test paths.

42 requirements are needed for Edge-Pairs
1. [0,1,14]
2. [0,1,7]
3. [0,1,2]
4. [0,1,21]
5. [1,14,15]
6. [1,7,8]
7. [1,2,3]
8. [1,21,22]
9. [2,3,4]
10. [2,3,5]
11. [3,4,5]
12. [3,5,6]
13. [4,5,6]
14. [5,6,28]
15. [6,28,29]
16. [7,8,9]
17. [7,8,11]
18. [8,9,10]
19. [9,10,13]
20. [8,11,12]
21. [11,12,10]
22. [12,10,13]
23. [10,13,28]
24. [13,28,29]
25. [21,22,23]
26. [21,22,25]
27. [22,23,24]
28. [23,24,27]
29. [22,25,26]
30. [25,26,24]
31. [26,24,27]
32. [24,27,28]
33. [27,28,29]
34. [14,15,16]
35. [14,15,18]
36. [15,16,17]
37. [16,17,20]
38. [15,18,19]
39. [18,19,17]
40. [19,17,20]
41. [17,20,28]
42. [20,28,29]

4

8 test paths are needed for Edge-Pair Coverage

Test Paths	Test Requirements that are toured by test paths directly
[0,1,14,15,16,17,20,28,29]	[0,1,14], [1,14,15], [14,15,16], [15,16,17], [16,17,20], [17,20,28], [20,28,29]
[0,1,7,8,9,10,13,28,29]	[0,1,7], [1,7,8], [7,8,9], [8,9,10], [9,10,13], [10,13,28], [13,28,29]
[0,1,2,3,5,6,28,29]	[0,1,2], [1,2,3], [2,3,5], [3,5,6], [5,6,28], [6,28,29]
[0,1,21,22,23,24,27,28,29]	[0,1,21], [1,21,22], [21,22,23], [22,23,24], [23,24,27], [24,27,28], [27,28,29]
[0,1,2,3,4,5,6,28,29]	[0,1,2], [1,2,3], [2,3,4], [3,4,5], [4,5,6], [5,6,28], [6,28,29]
[0,1,7,8,11,12,10,13,28,29]	[0,1,7], [1,7,8], [7,8,11], [8,11,12], [11,12,10], [12,10,13], [10,13,28], [13,28,29]
[0,1,21,22,25,26,24,27,28,29]	[0,1,21], [1,21,22], [21,22,25], [22,25,26], [25,26,24], [26,24,27], [24,27,28], [27,28,29]
[0,1,14,15,18,19,17,20,28,29]	[0,1,14], [1,14,15], [14,15,18], [15,18,19], [18,19,17], [19,17,20], [17,20,28], [20,28,29]

Edge pair coverage ,test requirement et test paths.

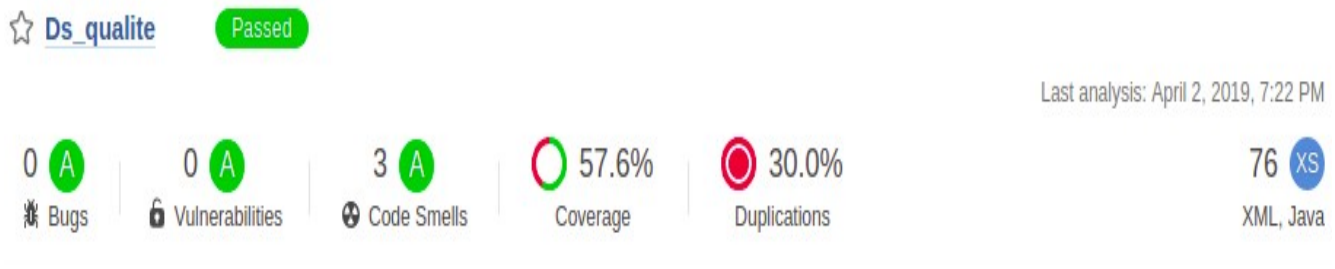
8 requirements are needed for Prime Paths
1. [0,1,14,15,18,19,17,20,28,29]
2. [0,1,7,8,11,12,10,13,28,29]
3. [0,1,21,22,25,26,24,27,28,29]
4. [0,1,2,3,4,5,6,28,29]
5. [0,1,7,8,9,10,13,28,29]
6. [0,1,14,15,16,17,20,28,29]
7. [0,1,21,22,23,24,27,28,29]
8. [0,1,2,3,5,6,28,29]

8 test paths are needed for Prime Path Coverage

Test Paths	Test Requirements that are toured by test paths directly
[0,1,14,15,18,19,17,20,28,29]	[0,1,14,15,18,19,17,20,28,29]
[0,1,7,8,11,12,10,13,28,29]	[0,1,7,8,11,12,10,13,28,29]
[0,1,21,22,25,26,24,27,28,29]	[0,1,21,22,25,26,24,27,28,29]
[0,1,2,3,4,5,6,28,29]	[0,1,2,3,4,5,6,28,29]
[0,1,7,8,9,10,13,28,29]	[0,1,7,8,9,10,13,28,29]
[0,1,14,15,16,17,20,28,29]	[0,1,14,15,16,17,20,28,29]
[0,1,21,22,23,24,27,28,29]	[0,1,21,22,23,24,27,28,29]
[0,1,2,3,5,6,28,29]	[0,1,2,3,5,6,28,29]

Test Paths	Test Requirements that are toured by test paths with sidetrips
[0,1,14,15,18,19,17,20,28,29]	None
[0,1,7,8,11,12,10,13,28,29]	None
[0,1,21,22,25,26,24,27,28,29]	None
[0,1,2,3,4,5,6,28,29]	None
[0,1,7,8,9,10,13,28,29]	None
[0,1,14,15,16,17,20,28,29]	None
[0,1,21,22,23,24,27,28,29]	None
[0,1,2,3,5,6,28,29]	None

Question 4:Taux de couverture (à l'aide de sonarqube)



Question 5:Modification de la classe FSM pour que le cas de test reachesFinalState ne mene pas à des défaillances.

```
public void transition(int input){
    switch(state){
        case 0:
            if (input==1){
                state=1;
            }
            break;
        case 1:
            if (input==3){
                state=2;
            }else{
                state=0;
            }
            break;
    /* case 2:
        if (input==3){
        state=3;
        }else{
        state=0;
        }
        break;*/
        case 2:
            if (input==7){
                state=4;
            }else{
                state=0;
            }
            break;
    }
}
```

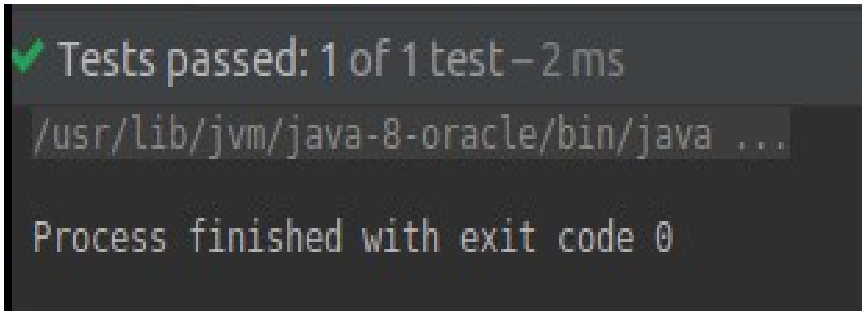
```
@Test
public void reachesFinalState(){
        FSM f=new FSM();
        f.transition( input: 1);
        f.transition( input: 3);
        f.transition( input: 3);
        f.transition( input: 7);
        assertTrue(f.isAcc());

}
```

```
✔ Tests passed: 1 of 1 test – 2 ms
/usr/lib/jvm/java-8-oracle/bin/java ...

Process finished with exit code 0
```

Question 6:

Le cas de test doesNotReachesFinalState

```
@Test
public void reachesFinalState(){
        FSM f=new FSM();
        f.transition( input: 1);
        f.transition( input: 3);
        f.transition( input: 3);
        f.transition( input: 3);
        assertFalse(f.isAcc());

}
```



Avec les transitions 1 3 3 3 ce cas de test ne mène pas à une defaillance

Question 7: le cas de test notAlphabet

```java
public class NotInAlphabetException extends Exception {
    public  NotInAlphabetException(){
        super("not in alphabet ");
    }

}
```

```java
public class FSM {
    private int state=0;
    public boolean isAcc(){
        return state==4 ;
    }
    public void transition(int input) throws NotInAlphabetException{
        switch(state){
            case 0:
                if (input==1){
                    state=1;
                } else throw new NotInAlphabetException();
                break;
            case 1:
                if (input==3){
                    state=2;
                }else throw new NotInAlphabetException();
                break;
            case 2:
                if (input==3){
                    state=3;
                }else throw new NotInAlphabetException();
                break;
            case 3:
                if (input==7){
                    state=4;
                }else  throw new NotInAlphabetException();
                break;
        }
    }
}
```

Après creation de l’exception et modification de code FSM pour qu’elle soit capable de déclancher l’exception

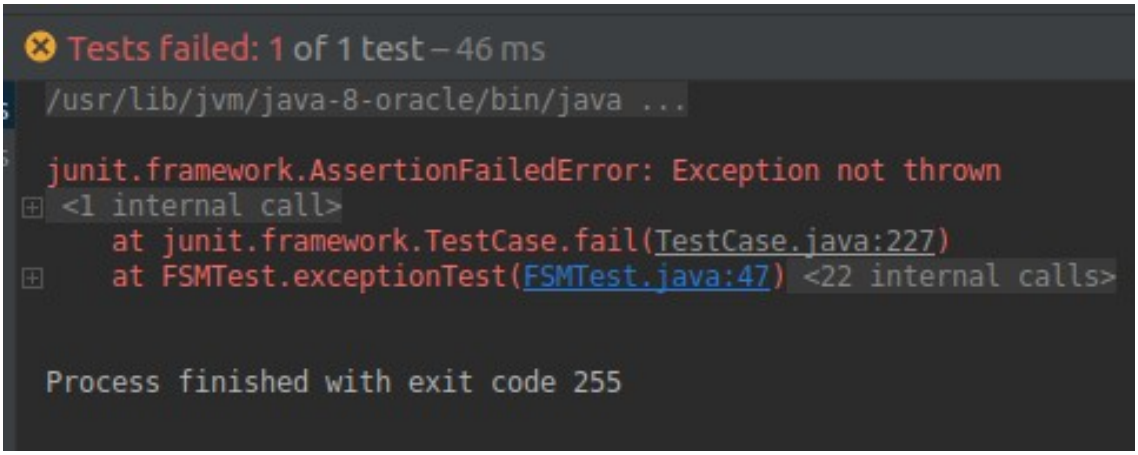
```
@Test
public void exceptionTest(){
    FSM f=new FSM();
    try {
        f.transition( input: 1);
        f.transition( input: 3);
        f.transition( input: 5);//
        f.transition( input: 7);
        fail("Exception not thrown");

    }
    catch (NotInAlphabetException e){
        System.out.println(e.getMessage());
    }
}
```

```
✔ Tests passed: 1 of 1 test – 7 ms
/usr/lib/jvm/java-8-oracle/bin/java ...
not in alphabet

Process finished with exit code 0
```

NB: 5 n’appartient pas à l’alphabet et il n’y a aucune transition avec 5

Si le cas de test catch une exception “notInAlphabetException” de test sera passé avec succés

```java
@Test
public void exceptionTest(){
    FSM f=new FSM();
    try {
        f.transition( input: 1);
        f.transition( input: 3);
        f.transition( input: 3);//
        f.transition( input: 7);
        fail("Exception not thrown");

    }
    catch (NotInAlphabetException e){
        System.out.println(e.getMessage());
    }
}
```



Si le test de catche pas une exception ,il sera echoué