amU Faculté des sciences Aix Marseille Université

Année 2025-26

Domaine Sciences et Technologies
Master informatique
Fiabilité logicielle : TD 2
Codes UE : SINB34A
Tests unitaires et couverture

1 Tests unitaires et couverture

Un outil de couverture de tests indique que la ligne `6` dans le code du programme ci-dessous n'est pas couverte.

```
1 public class PartialCoverage {
2   public static void main(String[] args) {
3     int x = 1;
4     int y = x>0 ? -1 : 0;
5     for (int i = x; x < y ; i++) {
6       System.out.println(i);
7     }
8   }
9 }
```

Question 1.1 : Expliquer ce qui se passe.

Question 1.2 : Peut-on réécrire ce code (en un code équivalent) pour supprimer ce phénomène ?

2 Graphes de flot de contrôle (1/2)

On considère le programme ci-dessous.

```
1  void proc (int n) {
2    if (n <= 0) {
3      n = 1-n;
4    }
5    if (n%2 == 0) {
6      n = n/2;
7    }
8    else {
9      n = 3*n + 1;
10   }
11   System.out.println(n);
12 }
```

Question 2.1 : Donner le graphe de flot de contrôle.

Question 2.2 : Donner une suite de tests la plus courte possible pour chacune des couvertures suivantes :

- la couverture de tous les nœuds.
- la couverture de toutes les arêtes.
- la couverture de tous les chemins simples

3 Graphes de flot de contrôle (2/2)

On considère le programme ci-dessous.

```
1  public int foo(int a, int b){
2    if (a > 0 && a != b){
3      a = b;
4      if (b == 1|| a > 0) {
5        b = 0;
6      }
7      else {
8        if (b < 0){
9           a = 0;
10       }
11     }
12   }
13   return a;
14 }
```

Question 3.1 : Donner le graphe de flot de contrôle de ce programme.

Question 3.2 : Donner une suite de tests permettant de tester

- toutes les conditions.
- tous les chemins (simples)

Question 3.3 : Donner le diagramme de flot de données. Que conclure ?

4 Chemins indépendants et flot de données

```
1  int gcd (int a, int b) {
2    if (a&lt;=b){              // l1
3      temp=a;
4      a=b;
5      b=temp;               // l2
6    }
```

```
7     while (a % b != 0) {    // 13
8       temp=b;
9       b=a % b;
10      a=temp;               // 14
11    }
12    return b;               // 15
13  }
```

Question 4.1 : Donner le graphe de flot de contrôle.

Question 4.2 : Donner une expression régulière pour les chemins.

Question 4.3 : Donner une suite de tests testant tous les chemins indépendants.

Question 4.4 : Donner le graphe de flot de données.

5 Exercices supplémentaires

```
1  int fib(int n) {
2      int f = 0;
3      int f0 = 1;
4      int f1 = 1;
5      while (n > 1) {
6          n--;
7          f = f0 + f1;
8          f0 = f1;
9          f1 = f;
10     }
11     return f;
12 }
```

Question 5.1 : Donner le graphe de flot de contrôle.

Question 5.2 : Donner le graphe de flot de données.

Question 5.3 : Ce programme censé calculer la suite de Fibonacci définie par $\mathsf{fib}_0 = 1$, $\mathsf{fib}_1 = 1$, $\mathsf{fib}_{n+2} = \mathsf{fib}_{n+1} + \mathsf{fib}_n$ si $n \geq 0$, est incorrect. Peut-on le voir à partir des CFG et DFG ?

Question 5.4 : Peut-on donner une suite de tests permettant de tester tous les chemins ? Tous les chemins simples ?

Question 5.5 : Donner une suite de tests qui évalue toutes les décisions.

6 Couverture des conditions et des décisions

On donne l'expression booléenne `(a && b) || (c && d)` qui est la condition d'un branchement conditionnel dans un programme de contrôle en avionique.

Question 6.1 : Donner une suite de tests minimale pour la métrique couverture des décisions, c'est-à-dire l'évaluation de la décision à vrai ou faux.

Question 6.2 : Donner une suite de tests minimale pour la métrique couverture des conditions, c'est-à-dire l'évaluation de chaque condition à vrai ou faux.

Question 6.3 : Donner une suite de tests minimale pour la métrique couverture des conditions/décisions modifiées.

Question 6.4 : Peut-on avoir une suite de tests pour les conditions qui n'est pas une suite de tests pour les décisions ? Réciproquement ?

7 Couverture des chemins

Question 7.1 : Donner un exemple de programme pour lequel il n'est pas possible d'avoir une suite de tests couvrant tous les chemins.

Question 7.2 : Peut-on avoir un programme pour lequel il existe une suite de tests minimale qui teste toutes les décisions, mais pas tous les chemins ? La réciproque est-elle vraie ?

8 Exercice bonus

Pour la méthode donnée ci-après répondre aux questions suivantes.

```
1 public int [][] delRangee(int [][] tab, int firstRangee, int lastRangee){
2     int [][] result=null;
3     int nbRangee =tab.length;
4     if ((firstRangee>=nbRangee)||(firstRangee <0)){
```

```
5           System.out.println("premiere rangee non valide");
6       }
7       else if((lastRangee>=nbRangee)||(lastRangee <0)){
8           System.out.println("derniere rangee non valide");
9       }
10      else if (lastRangee<firstRangee){
11          System.out.println("domaine d'indice invalide");
12      } else {
13          int nbNouvRangee=nbRangee -(lastRangee-firstRangee+1);
14          result=new int [nbNouvRangee][tab[0].length];
15          int offset=0;
16          for (int rangee=0; rangee <nbRangee; rangee++) {
17              if ((rangee>=firstRangee)&&(rangee <=lastRangee)) {
18                  offset++;
19              }
20              else {
21                result[rangee-offset]=tab[rangee];
22              }
23          }
24      }
25      return result;
26  }
```

Question 8.1 : Donner le graphe de flot de contrôle de la méthode.

Question 8.2 : Que fait cette méthode ? Rédiger un commentaire Java décrivant le comportement de cette méthode.

Question 8.3 : Donner une suite de tests pour :

- tester tous les chemins,
- tester toutes les décisions,
- tester toutes les conditions booléennes (élémentaire).

Question 8.4 : Donner le graphe de flot de données.