Université de Nantes
Master 1 ALMA
Vérification et tests

Nom: _____
Numéro: _____

Contrôle continu – Première session

– Documents autorisés –
– Dispositifs électroniques (téléphones, calculatrices, etc.) interdits –

Répondez aux questions sur cette feuille. Si vous n'avez pas assez de place pour finir votre réponse, utilisez une feuille additionnelle.
Ce contrôle a 4 questions, pour un total de 20 points.

Question:	1	2	3	4	Total
Points:	7	6	7	0	20
Score:					

```
1  import java.util.List;
2  import java.util.ArrayList;
3
4  public class SortedIntegerSet {
5      protected List<Integer> list;
6
7      public SortedIntegerSet()      { list  = new ArrayList<Integer>();}
8      public int get(int index)      {return list.get(index);}
9      public int size()              {return list.size();}
10     public void add(int value)     { list.add(value);}
11
12     public boolean containsAll(SortedIntegerSet other) {
13         if (other.size() > this.size())
14          return false;
15         int i = 0;
16         int j = 0;
17         while (i < other.size()) {
18             if (this.get(j) < other.get(i)) {
19                 j++;
20                 if (j ≥ this.size())
21                   return true;
22             } else if (other.get(i) == this.get(j))
23                 i++;
24             else
25                 return false;
26         }
27         return true;
28     }
29 }
```

FIGURE 1 – Code de la classe SortedIntegerSet

1. Test structurel

   La classe SortedIntegerSet (voir Figure 1) encapsule une liste d'entiers triés et un ensemble d'opérations pour manipuler cette liste. La méthode containsAll rend `true` si la liste courante contient tous les éléments de la liste `other` passée en paramètre.

(2) (a) Donnez le graphe de flot de contrôle de containsAll. Vous pouvez utiliser les numéros de ligne pour nommer les sommets.

Nom: _____
Numéro: _____

(2) (b) Combien de 1-chemins y a-t-il dans ce graphe ?

(2) (c) Donnez des données de test pour couvrir tous les 1-chemins de containsAll() et précisez le résultat attendu pour chacune de ces données. Vous n'avez pas à écrire le code JUnit correspondant, et vous pouvez simplement présenter vos tests sous la forme d'un tableau de la manière suivante :

d_x	d_y	...	Résultat attendu
a_1	b_1	...	r_1
a_2	b_2	...	r_2
...	...	...	...

avec d_x, d_y les paramètres de la méthode, a_i, b_j les données d'entrées de chaque test et r_1, r_2 les résultats attendus. S'il n'y a pas de résultat attendu valide, mettez un point d'interrogation dans la dernière colonne.

Nom: _____
Numéro: _____

(1) (d) Y a-t-il un bug dans cette méthode ? Si oui, lequel et à quelle ligne ?

_____

Nom: _____
Numéro: _____

Remplacer	par
`>`	`>=`
`true`	`false`
`this`	`other`

TABLE 1 – Opérateurs de mutation

2. **Analyse de la mutation**
Considérez les opérateurs de mutation suivants:

(2) (a) La Table 1 contient trois opérateurs de mutation. Appliqués sur containsAll, combien de mutants seront produits?

(2) (b) En considérant les opérateurs de mutation de la Table 1, calculez le score de mutation des données de test que vous avez produites lors de la question 1; détaillez la méthode que vous aurez suivie et vos résultats.

(2) (c) Qu'observez vous parmi les mutants ?

Nom: _____
Numéro: _____

3. **Test fonctionnel**
   Considérez la méthode suivante:

```
1  /**
2  * This operation defines the decision procedure of the home automation system,
3  * ie when should shutters be opened or closed depending on the sensors.
4  *
5  * This operation is stateless/functionnal: it only takes inputs and returns
6  * a value, it does NOT change or use the state of the object.
7  *
8  * The requirements concerning the shutters are the following:
9  * - If luminosity if less than 400, we close the shutters for the night.
10 * - If luminosity is between 400 and 100 000, we open the shutters.
11 * - If luminosity is more than 100 000, we close to avoid the heat.
12 * - If the wind direction is WEST, and the wind speed is more than 40, we open
13 * the shutters so that they don't break (because the windows mostly face west).
14 * - In all other cases, we do nothing.
15 *
16 * @param windSpeed the wind speed to consider
17 * @param windDirection the wind direction to consider
18 * @param luminosity the luminosity to consider
19 * @return Open is shutters should be opened, Close if they should be closed,
20 *             Nothing if nothing must be done.
21 */
22 public PlannedAction analyze(int windSpeed, Direction windDirection, int luminosity);
```

(1) (a) Identifiez les variables sur lesquelles jouer en entrée (les données de test), ainsi que les valeurs observables en résultat (Oracle).

(1) (b) **En considérant pour cette question que Direction n'a que deux valeurs possibles (`East` et `West`)**, réalisez pour chacune de ces valeurs une analyse partitionnelle, afin d'en déduire des classes d'équivalences.

(2) (c) Etablissez une table de décision décrivant le comportement de la méthode.

Nom: _____
Numéro: _____

(2) (d) A partir de cette table, déduisez un ensemble fini minimum de cas de test pertinents (données de test et oracle) pour la méthode.

(1) (e) Voyez vous un problème dans la spécification de l'opération analyze ? Si oui, lequel ?

(2½ (bonus)) 4. Pour chacune de ces déclarations, indiquez soit **vrai**, soit **faux**.

(a) Le test statique peut produire des verdicts erronés.
A. Vrai B. Faux

(b) Le test dynamique ne manque aucune erreur.
A. Vrai B. Faux

(c) Une suite de test qui couvre tous les chemins trouve toutes les erreurs.
A. Vrai B. Faux

(d) Le test par mutation permet de simuler la couverture des instructions.
A. Vrai B. Faux

(e) Il existe des algorithmes qui garantissent l'ordre optimal pour le test d'intégration.
A. Vrai B. Faux