

PRG1 – TP4 – Classe SmallSet

On désire écrire une classe `SmallSet` permettant de déclarer et de manipuler des ensembles définis sur le domaine $0 \dots 255$. On choisit de représenter chaque ensemble E par un tableau **boolean** `[256] tab` avec la convention $i \in E \Leftrightarrow E.tab[i]$ vaut vrai. Compléter la classe `SmallSet` suivante :

```
1 public class SmallSet {
2
3     private boolean[] tab = new boolean[256];
4
5     public SmallSet () {
6         for (int i = 0; i <= 255; ++i) {
7             tab[i] = false;
8         }
9     }
10
11     public SmallSet (boolean[] t) {
12         for (int i = 0; i <= 255; ++i) {
13             tab[i] = t[i];
14         }
15     }
16
17     /**
18      * @return      nombre de valeurs appartenant à l'ensemble
19      */
20     public int size () { ... }
21
22     /**
23      * @param x      valeur à tester
24      * @pre          0 <= x <= 255
25      * @return      true, si l'entier x appartient à l'ensemble,
26      *              false sinon
27      */
28     public boolean contains (int x) { ... }
29
30
31 }
```

```
32  /**
33   * @return      true, si l'ensemble est vide, false sinon
34   */
35  public boolean isEmpty () { ... }
36
37  /**
38   * Ajoute x à l'ensemble (sans effet si x déjà présent)
39   *
40   * @param  x      valeur à ajouter
41   * @pre      0 <= x <= 255
42   */
43  public void add (int x) { ... }
44
45  /**
46   * Retire x de l'ensemble (sans effet si x n'est pas présent)
47   *
48   * @param  x      valeur à supprimer
49   * @pre      0 <= x <= 255
50   */
51  public void remove (int x) { ... }
52
53  /**
54   * Ajoute à l'ensemble les valeurs deb, deb+1, deb+2, ..., fin.
55   *
56   * @param begin    début de l'intervalle
57   * @param end      fin de l'intervalle
58   * @pre      0 <= begin <= end <= 255
59   */
60  public void addInterval (int deb, int fin) { ... }
61
62  /**
63   * Retire de l'ensemble les valeurs deb, deb+1, deb+2, ..., fin.
64   *
65   * @param begin    début de l'intervalle
66   * @param end      fin de l'intervalle
67   * @pre      0 <= begin <= end <= 255
68   */
69  public void removeInterval (int deb, int fin) { ... }
70
71  /**
72   * Réalise l'opération this ← this ∪ set2.
73   *
74   * @param set2      second ensemble
75   */
76  public void union (SmallSet set2) { ... }
```

```
77
78  /**
79   * Réalise l'opération this  $\leftarrow$  this  $\cap$  set2.
80   *
81   * @param set2      second ensemble
82   */
83  public void intersection (SmallSet set2) { ... }
84
85  /**
86   * Réalise l'opération this  $\leftarrow$  this  $\setminus$  set2.
87   *
88   * @param set2      second ensemble
89   */
90  public void difference (SmallSet set2) { ... }
91
92  /**
93   * Réalise l'opération this  $\leftarrow$  this  $\Delta$  set2.
94   *
95   * @param set2      second ensemble
96   */
97  public void symmetricDifference (SmallSet set2) { ... }
98
99  /**
100   * Réalise l'opération this  $\leftarrow$   $\overline{\text{this}}$ .
101   */
102  public void complement () { ... }
103
104  /**
105   * Réalise l'opération this  $\leftarrow$   $\emptyset$ .
106   */
107  public void clear () { ... }
108
109  /**
110   * @param set2      second ensemble
111   * @return          true, si this  $\subseteq$  set2, false sinon
112   */
113  public boolean isIncludedIn (SmallSet set2) { ... }
114
115  /**
116   * @return          copie de this
117   */
118  public SmallSet clone () { ... }
119      return new SmallSet(tab);
120  }
121
```

```
122  /**
123   * @return      true, si this et set2 sont égaux, false sinon
124   */
125  @Override
126  public boolean equals (Object set2) { ... }
127
128  @Override
129  public String toString() {
130      String s = "éléments présents : ";
131      for (int i = 0; i <= 255; ++i) {
132          if (tab[i]) {
133              s = s + i + " ";
134          }
135      }
136      return s;
137  }
138 }
```

Pour tester vos méthodes, vous disposez dans le répertoire

G:\l3miage\prg1\tp4 (sous Windows)

/share/l3miage/prg1/tp4 (sous Linux) :

- du squelette de la classe SmallSet
- de la classe TestSmallSet qui gère 5 ensembles déclarés par
private static SmallSet[] E = **new** SmallSet[5]
et qui itère sur un menu proposant les méthodes SmallSet.