

```

18
19 public static ArrayList<Integer> liste_des_cachés = new ArrayList<>();
20 public static ArrayList<Integer> liste_des_isolés = new ArrayList<>();
21
22 /* Liste "public" déclarée ailleurs.
23
24 liste_des_cases_vides = ArrayList<Case>;
25
26 */
27
28 static void Recherche_des_Célibataires() {
29
30     ArrayList<Integer> isolés = new ArrayList<>();
31     ArrayList<Integer> cachés = new ArrayList<>();
32
33     for (int candidat : de_1_à_9) {
34         for (Case kase : liste_des_cases_vides) {
35             if (kase.contient(candidat)) {
36
37                 if (Célibataire_isolé_trouvé_dans_la_case(kase, candidat)) {
38
39                     /* Pour l'affichage sur la grille des célibataires isolés. */
40                     liste_des_isolés.add(candidat);
41                     liste_des_isolés.add(kase.numéro);
42
43                     /* Pour l'affichage de la liste des célibataires isolés. */
44                     isolés.add(candidat);
45                     isolés.add(kase.numéro + 1);
46
47                     continue;
48                 }
49
50                 if (Célibataire_caché_trouvé_dans_la_case(kase, candidat)) {
51
52                     /* Pour l'affichage sur la grille des célibataires cachés. */
53                     liste_des_cachés.add(candidat);
54                     liste_des_cachés.add(kase.numéro);
55
56                     /* Pour l'affichage de la liste des célibataires cachés. */
57                     cachés.add(candidat);
58                     cachés.add(kase.numéro + 1);
59                 }
60             }
61         }
62     }
63     Afficher_le_message("célibataires isolés: " + isolés + "\ncélibataires cachés: " + cachés);
64 }
65
66 static boolean Célibataire_isolé_trouvé_dans_la_case(Case kase, int candidat) {
67
68     ArrayList<Integer> liste = new ArrayList<>();
69
70     for (int kan : de_1_à_9) {
71         if (kase.contient(kan)) liste.add(kan);
72     }
73     return liste.size() == 1 && liste.get(0) == candidat;
74 }
75
76 static boolean Célibataire_caché_trouvé_dans_la_case(Case kase, int candidat) {
77
78     /* Si le candidat est seul dans le bloc */
79     if (Compte_du_candidat_dans_le_bloc(kase.bloc, candidat) == 1) return true;
80
81     /* Si le candidat est seul dans la colonne */
82     if (Compte_du_candidat_dans_la_colonne(kase.colonne, candidat) == 1) return true;
83
84     /* Si le candidat est seul dans la ligne */
85     return Compte_du_candidat_dans_la_ligne(kase.ligne, candidat) == 1;
86 }
87

```

```

88
89  static int Compte_du_candidat_dans_le_bloc(int bloc, int candidat) {
90
91      ArrayList<Case> liste = new ArrayList<>();
92      Case case_de_grille;
93
94      for (int case_de_bloc : de_1_à_9) {
95
96          case_de_grille = la_Case(3 * (bloc % 3) + (case_de_bloc - 1) % 3 + 9 * (3 * (bloc / 3)
97              + (case_de_bloc - 1) / 3));
98
99          if (case_de_grille.contient(candidat)) liste.add(case_de_grille);
100      }
101
102      return liste.size();
103  }
104
105  static int Compte_du_candidat_dans_la_colonne(int colonne, int candidat) {
106
107      ArrayList<Case> liste = new ArrayList<>();
108      Case case_de_grille;
109
110      for (int ligne : de_1_à_9) {
111
112          case_de_grille = la_Case(colonne + 9 * (ligne - 1));
113
114          if (case_de_grille.contient(candidat)) liste.add(case_de_grille);
115      }
116
117      return liste.size();
118  }
119
120  static int Compte_du_candidat_dans_la_ligne(int ligne, int candidat) {
121
122      ArrayList<Case> liste = new ArrayList<>();
123      Case case_de_grille;
124
125      for (int colonne : de_1_à_9) {
126
127          case_de_grille = la_Case(colonne - 1 + 9 * ligne);
128
129          if (case_de_grille.contient(candidat)) liste.add(case_de_grille);
130      }
131
132      return liste.size();
133  }
134
135  /* -----*/

```