

```

14
15 public class Case extends Fragment {
16
17     /*
18     candidats = new boolean[9][9][9];
19
20     Matrice "public" déclarée et mise-à-jour ailleurs.
21     Cette matrice contient l'état des candidats de chacune des 81 cases de la grille.
22     */
23
24     public static Case[] case_de = new Case[81];
25     public int numéro, ligne, colonne, bloc;
26
27     public Case() {
28
29     }
30
31     public Case(int _numéro) {
32
33         numéro = _numéro;           // numéro = {0,...,80}; numéro = colonne + 9*ligne.
34         ligne = _numéro / 9;        // ligne = {0,...,8}.
35         colonne = _numéro % 9;      // colonne = {0,...,8}.
36         bloc = colonne/3 + 3 * (ligne/3); // bloc = {0,...,8}.
37
38     }
39
40     public static Case la_case(int numéro) {
41         return case_de[numéro];
42     }
43
44     public boolean contient_le(int candidat) { // candidat = {1,...,9}.
45         return candidats[this.ligne][this.colonne][candidat - 1];
46     }
47
48     public int getCouple() {
49         int couple = 0;
50         for (int candidat : de_1_à_9) { // L'indice candidat varie de 1 à 9.
51             if (this.contient_le(candidat)) {
52                 couple = 10 * couple + candidat;
53             }
54         }
55         if (couple > 10 && couple < 99) {
56             return couple;
57         } else {
58             return 0;
59         }
60     }
61
62     public int getTriplet() {
63         int triplet = 0;
64         for (int candidat : de_1_à_9) { // L'indice candidat varie de 1 à 9.
65             if (this.contient_le(candidat)) {
66                 triplet = 10 * triplet + candidat;
67             }
68         }
69         if (triplet > 111 && triplet < 999) {
70             return triplet;
71         } else {
72             return 0;
73         }
74     }
75
76     /* -----*/
77

```