

Éléments de correction de TP – NFA031

Raphaël Fournier-S'niehotta, fournier@cnam.fr

Décembre 2014

1 TP8

1.1 Exercice 1

```
1  class TP8Exo1{
2
3      public static int sommeTableau(int[] table){
4          int somme = 0;
5          for (int i = 0; i < table.length; i++) {
6              somme += table[i];
7          }
8          return somme;
9      }
10     public static void main(String[] args){
11         int [] monTableau = {12, 245, 3};
12         int k = sommeTableau(monTableau);
13         Terminal.ecrireIntln(k);
14     }
15 }
```

1.2 Exercice 2

```
1  class TP8Exo2{
2
3      public static boolean verifierPresenceTableau(char c, int[] table) {
4          boolean isPresent=false;
5          for (int i = 0; i < table.length; i++) {
6              if(c==table[i]){
7                  isPresent=true;
8                  break;
9              }
10         }
11         return isPresent;
12     }
13
14     public static void main(String[] args){
15         int [] tableau = {'a','z','e','r','t','y'};
16         // on utilise maintenant notre procédure de vérification,
17         // en stockant son résultat dans un booléen
18         boolean b = verifierPresenceTableau('w',tableau);
19         Terminal.ecrireString("Est-ce que "+ "w" + " est dans le tableau ? Réponse : ");
20         Terminal.ecrireBooleanln(b); // false
21         // ou alors, sans utiliser de booléen intermédiaire :
22         Terminal.ecrireString("Est-ce que "+ "e" + " est dans le tableau ? Réponse :");
23         Terminal.ecrireBoolean(verifierPresenceTableau('e',tableau));
24     }
25 }
```

1.3 Exercice 3

```
1  class TP8Exo3{
2
3      public static boolean t1_dans_t2(char[] t1, char[] t2) {
4          boolean inclus=true;
5          for (int i = 0; i < t1.length; i++) { // pour chaq char dans t1
6              boolean estDansT2=false;
7              for (int j = 0; j < t2.length; j++) { // on parcourt t2
8                  if(t1[i]==t2[j]){
9                      estDansT2=true;
10                 }
11             }
12             if(!estDansT2){
13                 //vous pouvez décommenter la ligne suivante pour afficher la lettre qui
14                 // est dans le 1er tableau mais pas dans le 2e
15                 //Terminal.ecrireString(t1[i]+" n'est pas dans t2 !");
16                 inclus=false;
17                 // inutile d'aller plus loin, une fois qu'on a trouvé un char de t1 non inclus dans t2
18                 break;
19             }
20         }
21     }
22     }
23     return inclus;
24 }
25
26 public static void main(String[] args){
27     char [] t1 = {'a','z','e','r','t','y'};
28     char [] t2 = {'a','z','e','r','t','y', 'v','b','n'};
29     char [] t3 = {'a','z','e','f','t','y', 'v','b','n'};
30
31     boolean b = t1_dans_t2(t1,t2);
32     Terminal.ecrireString("Est-ce que t1 est inclus dans t2 ? Réponse : ");
33     Terminal.ecrireBooleanln(b);
34
35     // comme en q2, sans booléen intermédiaire :
36     Terminal.ecrireString("Est-ce que t2 est inclus dans t3 ? Réponse : ");
37     Terminal.ecrireBooleanln(t1_dans_t2(t2, t3));
38 }
39 }
```

1.4 Exercice 4

```
1  class TP8Exo4{
2
3      public static boolean estUneSousChaine(char [] t1, char[] t2){
4          boolean isSC=false;
5          // pos sert à savoir où on en est de la chaîne recherchée
6          int pos=0;
7          // une sous chaîne ne peut pas commencer trop près de la fin de t2
8          // d'où le "-t1.length"
9          for (int j = 0; j < t2.length-t1.length; j++) {
10             // pour la position courante de t2, on regarde s'il y a égalité
11             // avec le caractère courant de la chaîne recherchée
12             if(t2[j]==t1[pos]){
13                 // s'il y a égalité, on incrémente, en vue de tester
14                 // le caractère suivant de la chaîne cherchée
15                 pos++;
16                 // si on arrive alors à pos==t1.length, c'est que tout est bon :
17                 // c'est une sous-chaîne
18                 if(pos==t1.length){
19                     isSC=true;
20                     Terminal.ecrireStringln("SC de "+(j-t1.length)+"à "+j);
21                     break;
22                 }
23             }else{
24                 // s'il n'y a pas égalité
25                 // il faut que l'on réinitialise notre recherche :
26                 // on revient en arrière dans l'étude des caractères de t2,
27                 // et pos est remis à zéro
28                 j-=pos;
29                 pos=0;
30             }
31         }
32         return isSC;
33     }
34
35     public static void main(String[] args){
36         char [] t1 = {'c','h','a','t'};
37         char [] t2 = {'c','h','a','t','c','h','i','e','n'};
38         char [] t3 = {'t','c','h'};
39         char [] t4 = {'z','z','z'};
40
41         Terminal.ecrireString("Est-ce que "+String.valueOf(t1)+" est une SC de "+String.valueOf(t2)+" ? Réponse : ");
42         Terminal.ecrireBooleanln(estUneSousChaine(t1, t2));
43         Terminal.ecrireStringln("****");
44         Terminal.ecrireString("Est-ce que "+String.valueOf(t3)+" est une SC de "+String.valueOf(t2)+" ? Réponse : ");
45         Terminal.ecrireBooleanln(estUneSousChaine(t3, t2));
46         Terminal.ecrireStringln("****");
47         Terminal.ecrireString("Est-ce que "+String.valueOf(t4)+" est une SC de "+String.valueOf(t2)+" ? Réponse : ");
48         Terminal.ecrireBooleanln(estUneSousChaine(t4, t2));
49     }
50 }
```
