

Éléments de correction de TP – NFA031

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

Décembre 2015

1 TP4 Nouvelle version (appelé “TP3” dans le sujet PDF)

1.1 Exercice 1

1.1.1 Q1.

Pour Multiples10.java, le programme n’ayant pas besoin d’entrées et affichant toujours le même résultat, on n’a juste besoin de vérifier que la sortie est celle attendue.

Pour Puissances2N.java, un jeu de test doit inclure N=0 (une valeur inférieure à la valeur initiale de power, qui est de 1), N=1 et une ou plusieurs valeurs au dessus de 1. On choisit de préférence un assortiment, des valeurs faibles comme 17 ou 33, mais aussi des plus grandes comme 256 ou 512 (pour lequel on doit avoir la liste (1, 2, 4, 8, 16, 32, 64, 128, 256, 512)).

1.1.2 Q2.

Solution possible :

```
1 public class Multiples10_2{
2     public static void main(String[] args){
3         int a=10;
4         // boucle
5         while (a<=90){
6             System.out.print(a+",");
7             a=a+10;
8         }
9         System.out.print(a);
10    }
11 }
```

1.1.3 Q3.

Solution possible :

```
1 public class Multiples10_3{
2     public static void main(String[] args){
3         // avec une boucle for
4         for(int i=1;i<=10;i++){
5             System.out.print(10*i);
6             if(i!=10){
7                 System.out.print(",");
8             }
9         }
10    }
11 }
```

1.1.4 Q4.

Solution possible :

```
1 public class PuissancesDe2_4 {
2     public static void main(String[] args) {
3
4         // lecture
5         Terminal.ecrireString("N ?");
6         int N = Terminal.lireInt();
7
8         int power = 1;    // i-eme puissance de 2
9
10        for (int i = 0; i <= N; i++) {
11            System.out.println(power);
12            power = power *2;
13            if( power > N ) {
14                break;
15            }
16        }
17    }
18 }
```

1.2 Exercice 2 Maximum des nombres lus

1.2.1 Q1.

```
1 public class maxNombres {
2     public static void main(String[] args) {
3         int max = 0;
4         int nb = 0;
5         for(int i=0; i<10;i++){
6             Terminal.ecrireString("nb ?");
7             nb = Terminal.lireInt();
8             if(nb > max){
9                 max = nb;
10            }
11        }
12        Terminal.ecrireString("le max est :"+max);
13    }
14 }
```

1.2.2 Q2.

```
1 public class maxNombres2 {
2
3     public static void main(String[] args) {
4         int max = 0;
5         int nb =-1;
6         while(nb!=0){
7             Terminal.ecrireString("Tapez un nombre ou 0 pour sortir :");
8             nb = Terminal.lireInt();
9
10            if(nb>max){
11                max=nb;
12            }
13        }
14        if(max!=0){
15            Terminal.ecrireStringln("Le max est "+max);
16        }else{
17            Terminal.ecrireStringln("le premier nombre saisi était 0, recommencez !");
18        }
19        Terminal.ecrireStringln("Au revoir");
20    }
21
22 }
```

1.3 Exercice 3

1.3.1 Q1.1

```
1  /**
2  * Ce programme lit les bornes inf, sup d'un intervalle que l'on suppose non vide,
3  * à savoir, tel que inf <=sup.
4  * On lit ensuite un entier x jusqu'à que sa valeur soit comprise dans l'intervalle
5  * [inf, sup]. Plus précisément, jusqu'à ce que inf <=x <= sup.
6  */
7  public class DansIntervalleValide{
8      public static void main(String[] args){
9          int inf, sup;
10         int x = 0;
11         boolean dedans = false; // pour contrôler la boucle
12         // lecture non sécurisée de l'intervalle
13         Terminal.ecrireString("Entrez la borne inférieure de l'intervalle: ");
14         inf= Terminal.lireInt();
15         Terminal.ecrireString("Entrez la borne supérieure de l'intervalle: ");
16         sup= Terminal.lireInt();
17         // Au départ dedans vaut false, on n'a pas encore lu x
18         while (!dedans){
19             Terminal.ecrireString("Entrez un entier compris dans [\"+inf+\",\"+sup+\"]: ");
20             x = Terminal.lireInt();
21             if (x < inf || x > sup){ // si en dehors, message d'erreur
22                 Terminal.ecrireStringln(x+\" est en dehors de l'intervalle. Recommencez.\");
23             } else { // sinon, on change dedans pour sortir de la boucle
24                 dedans=true;
25             }
26         }
27         Terminal.ecrireString("La valeur lue "+x+\" est dans [\"+inf+\",\"+sup+\"]: ");
28     }
```

Testez avec :

- (12,45), 23
- (12,45), 3
- (12,45), 53
- (0,5), 3
- (-12,-5), -8
- (-12,-5), -3

1.3.2 Q1.2

Si inf est supérieur à sup, strictement, alors dans la conditionnelle, x est considéré comme n'appartenant pas à l'intervalle, systématiquement. En effet, soit $x < \text{inf}$, soit $x > \text{sup}$ est vrai, donc $x < \text{inf} \parallel x > \text{sup}$ est toujours vrai. On est alors dans une situation où la boucle est "infinie". Exemple avec (12,45) et $x=3,23$ ou 53.

L'idée de cet exercice est de montrer que parfois on fait des hypothèses implicites lorsqu'on écrit du code ($\text{inf} \leq \text{sup}$), et si on veut des programmes robustes on doit assurer qu'elles sont validées.

1.3.3 Q2

```
1 public class DansIntervalleValide{
2     public static void main(String[] args){
3         int x = 0;
4         boolean dedans = false; // pour contrôler la boucle
5         int inf,sup;
6
7         // lecture sécurisée de l'intervalle
8         do{
9             Terminal.ecrireStringln("Entrez un intervalle valide : ");
10
11             Terminal.ecrireString("Entrez la borne inférieure de l'intervalle: ");
12             inf= Terminal.lireInt();
13             Terminal.ecrireString("Entrez la borne supérieure de l'intervalle: ");
14             sup= Terminal.lireInt();
15
16             if(inf>sup){
17                 Terminal.ecrireStringln("La borne inf est supérieure à la borne sup,
18                 recommencez !");
19             }
20
21         }while(inf>sup);
22         // maintenant, on est sûrs que inf<=sup
23
24         // Au départ dedans vaut false, on n'a pas encore lu x
25         while (!dedans){
26             Terminal.ecrireString("Entrez un entier compris dans [\"+inf+\",\"+sup+\"]: ");
27             x = Terminal.lireInt();
28             if (x < inf || x > sup){ // si en dehors, message d'erreur
29                 dedans = false;
30                 Terminal.ecrireStringln(x+" est en dehors de l'intervalle. Recommencez.");
31             } else { // sinon, on change dedans pour sortir de la boucle
32                 dedans=true;
33             }
34         }
35         Terminal.ecrireString("La valeur lue "+x+" est dans [\"+inf+\",\"+sup+\"]: ");
36     }
37 }
```
