

# TD 1

## Programmation Java

Y. ALJ

11 février 2020

1 Réorganiser les morceaux suivants pour obtenir le résultat (output) ci-dessous :

```
class Shuffle1 {  
    public static void main(String [] args) {
```

```
        if (x == 1) {  
            System.out.print("d");  
            x = x - 1;  
        }
```

```
        if (x > 2) {  
            System.out.print("a");  
        }
```

```
        int x = 3;
```

```
        x = x - 1;  
        System.out.print("-");
```

```
        while (x > 0) {
```

```
            if (x == 2) {  
                System.out.print("b c");  
            }
```

Output:

```
File Edit Window Help Sleep  
* java Shuffle1  
a-b c-d
```

## 2 Devenez compilateur et dites ce qui ne va pas des les morceaux de code suivants :

<pre>1 class Exercice2a { 2     public static void main(String [] args) { 3         int x = 1; 4         while ( x &lt; 10 ) { 5             if ( x &gt; 3) { 6                 System.out.println("big x"); 7             } 8         } 9     } 10 }</pre>	<pre>1 public static void main(String [] args) { 2     int x = 5; 3     while ( x &gt; 1 ) { 4         x = x - 1; 5         if ( x &lt; 3) { 6             System.out.println("small x"); 7         } 8     } 9 }</pre>
---	---

```
1 class Exercice2c {
2     int x = 5;
3     while ( x > 1 ) {
4         x = x - 1;
5         if ( x < 3) {
6             System.out.println("small x");
7         }
8     }
9 }
```

### 3 Deviner ce que fait chaque ligne du programme :

1	int size = 27;	1	.....
2		2	.....
3	String nom_du_chien = "Bobby";	3	.....
4		4	.....
5	Chien mon_chien = new Chien(nom_du_chien, size);	5	.....
6		6	.....
7		7	.....
8	x = size - 5;	8	.....
9		9	.....
10		10	.....
11		11	.....
12	if (x < 15) mon_chien.aboyer(8);	12	.....
13		13	.....
14		14	.....
15		15	.....
16	while (x > 3) {	16	.....
17	mon_chien.play();	17	.....
18	}	18	.....
19		19	.....
20		20	.....
21	int[] numList = {2,4,6,8};	21	.....
22		22	.....
23		23	.....
24	System.out.print("Salut");	24	.....
25		25	.....
26	System.out.print("Chien " + nom_du_chien);	26	.....
27		27	.....
28	String num = "8";	28	.....
29		29	.....
30	int z = Integer.parseInt(num);	30	.....
31		31	.....
32	try {	32	.....
33	readTheFile("mon_fichier.txt");	33	.....
34	}	34	.....
35	catch(FileNotFoundException ex) {	35	.....
36	System.out.print("Fichier introuvable");	36	.....
37	}	37	.....

1	<code>int size = 27;</code>	1	<code>// declarer une variable de type entier appelee "size" et lui donner la valeur 27</code>
2		2	
3	<code>String nom_du_chien = "Bobby";</code>	3	<code>// declarer une variable de type String appelee "nom_du_chien" et lui donner la valeur Bobby</code>
4		4	
5	<code>Chien mon_chien = new Chien(nom_du_chien, size);</code>	5	<code>// declarer une nouvelle variable de type Chien appelee "mon_chien" et</code>
6		6	
7		7	<code>// creer mon chien a partir de "nom_du_chien" et "taille"</code>
8	<code>x = size - 5;</code>	8	
9		9	
10		10	<code>// soustraire 5 de 27 et l assigner a la variable x</code>
11		11	
12	<code>if (x &lt; 15) mon_chien.aboyer(8);</code>	12	<code>// si x est inferieur a 15 aboyer 8 fois</code>
13		13	<code>// faire quelque chose en boucle tant que x est superieure a 3</code>
14		14	<code>// demander au chien de jouer</code>
15		15	<code>// fin de la boucle while. Ce qui est entre {...}</code>
16		16	<code>// sera fait en boucle.</code>
17		17	
18		18	<code>// declarer une liste d entiers appelee numList et mettre</code>
19		19	<code>// les valeurs 2, 4, 6 et 8 dedans</code>
20		20	
21	<code>int[] numList = {2,4,6,8};</code>	21	<code>// afficher "Salut"</code>
22		22	
23		23	<code>// afficher "Salut Bobby"</code>
24	<code>System.out.print("Salut");</code>	24	
25		25	<code>// declarer une variable de type String qui a comme nom 'num'</code>
26	<code>System.out.print("Chien " + nom_du_chien);</code>	26	<code>// et lui donner la valeur "8"</code>
27		27	
28	<code>String num = "8";</code>	28	<code>// convertir la chaine de caractere "8" en une</code>
29		29	<code>// valeur numerique</code>
30	<code>int z = Integer.parseInt(num);</code>	30	
31		31	<code>// Essayer de faire qqch. Peut etre ce que l on cherche</code>
32	<code>try {</code>	32	<code>// faire ne va pas marcher</code>
33	<code>readTheFile("mon_fichier.txt");</code>	33	
34	<code>}</code>	34	<code>// Lire un fichier texte appele "mon_fichier.txt"</code>
35	<code>catch(FileNotFoundException ex) {</code>	35	<code>// (On essaie de le lire en tout cas!)</code>
36	<code>System.out.print("Fichier introuvable");</code>	36	<code>// Fin de ce que lon essaie de faire {}</code>
37	<code>}</code>	37	
		38	<code>// Ce qui est mis dans ce bloc {...} va etre execute</code>
		39	<code>// si jamais ce qu on essaie de faire ne marche pas.</code>
		40	
		41	<code>// Si ce qu on essaie de faire ne marche pas afficher</code>
		42	<code>// fichier introuvable</code>
		43	<code>// Fin du traitement effectue au cas ou ce qu on essaie</code>
		44	<code>// essaie de faire ne marche pas.</code>