

Opérations sur les variables numériques en Java

I - Les 4 opérations de base sur les variables numériques

Opération	Symbole dans un programme Java
somme	+
différence	-
produit	*
quotient	/

II - Exemples de programmes utilisant des variables numériques

Le programme suivant affiche la somme, le produit, la différence et le quotient de 2 nombres a et b :

```

1  class calcul {
2      public static void main(String[] args) {
3
4          /* déclaration des variables de type double : */
5          double a,b,somme,difference,produit,quotient;
6
7          /* initialisation des variables a et b : */
8          a=5;
9          b=8;
10
11         /* calcule les 4 opérations avec les nombres a et b
12         et met les résultats dans les variables somme,
13         difference, produit et quotient : */
14         somme=a+b;
15         difference=a-b;
16         produit=a*b;
17         quotient=a/b;
18
19         /* affichage des résultats à l'écran : */
20         System.out.print("Le nombre a vaut ");
21         System.out.println(a);
22
23         System.out.print("Le nombre b vaut ");
24         System.out.println(b);
25
26         System.out.print("a+b = ");
27         System.out.println(somme);
28
29         System.out.print("a-b = ");
30         System.out.println(difference);
31
32         System.out.print("a*b = ");
33         System.out.println(produit);
34
35         System.out.print("a/b = ");
36         System.out.println(quotient);
37     }
38 }

```

Code source 1

Le programme suivant affiche le diamètre, la circonférence et l'aire d'un cercle, en fonction de son rayon :

```

1  class cercle {
2      public static void main(String[] args) {
3
4          /* déclaration des variables de type double : */
5          double pi,rayon,diametre,circonference,aire;

```

```

5      /* initialisation des variables : */
6      pi=3.14159265358979;
7      rayon=5;

8      /* calcul des 3 grandeurs du cercle */
9      diametre=2*rayon;
10     circonference=2*pi*rayon;
11     aire=pi*rayon*rayon;

12     /* affichage des résultats à l'écran : */
13     System.out.print("Le rayon du cercle vaut ");
14     System.out.println(rayon);

15     System.out.print("Le diamètre du cercle vaut ");
16     System.out.println(diametre);

17     System.out.print("La circonférence du cercle vaut ");
18     System.out.println(circonference);

19     System.out.print("L'aire du cercle vaut ");
20     System.out.println(aire);
21 }
22 }

```

Code source 2

Une fois exécuté, l'affichage de ce programme est le suivant :

```

Le rayon du cercle vaut 5.0
Le diamètre du cercle vaut 10.0
La circonférence du cercle vaut 31.4159265358979
L'aire du cercle vaut 78.53981633974475

```

Remarque : dans le *code source 2*, on aurait pu afficher chaque résultat avec une seule instruction `System.out.println`, à condition de réparer la chaîne de caractères et la variable à afficher par un signe `+` :

```

/* affichage des résultats à l'écran : */
System.out.println("Le rayon du cercle vaut "+rayon);

System.out.println("Le diamètre du cercle vaut "+diametre);

System.out.println("La circonférence du cercle vaut "+circonference);

System.out.println("L'aire du cercle vaut "+aire);

```

Exercice 1 : Réalisez un programme qui affiche le périmètre et l'aire d'un rectangle, en fonction de sa longueur et de sa largeur. L'affichage pourra ressembler à cela :

```

La longueur du rectangle est : 5 cm
La largeur du rectangle est : 2 cm
Le périmètre du rectangle est : 14 cm
L'aire du rectangle est : 10 cm carré

```

III - Les chaînes de caractères

Qu'est-ce qu'un chaîne de caractères ?

Comme son nom l'indique, une chaîne de caractères est un ensemble de caractères mis bout à bout. Les expressions suivantes sont des chaînes de caractères :

```

"Bonjour"
"J'aime le JAVA."
"Trois plus deux égale cinq"
"123456"
"Où ? Ici ! C'est pas vrai ?"
"Et oui, on peut mettre n'importe quel caractères dans une chaîne : , ; $ * # "

```

Remarques concernant les chaînes de caractères en JAVA :

- * Une chaîne de caractère peut contenir n'importe quels caractères :
 - * Des lettres minuscules
 - * Des lettres majuscules
 - * Des chiffres
 - * Des caractères de ponctuation
 - * Des caractères spéciaux
- * Une chaîne de caractère est toujours placée entre deux caractères **double côte** " [obtenu par la touche 3 du clavier en minuscule].
- * Une variable contenant une chaîne de caractères est une variable de type **String** [avec un S majuscule].

Premier exemple de programme utilisant des chaînes de caractères :

```

1 class calcul {
2     public static void main(String[] args) {
3
4         /* déclaration d'une variable de type String : */
5         String ma_chaine;
6
7         /* initialisation de la chaîne de caractères : */
8         ma_chaine="Aujourd'hui, je programme en Java !";
9
10        /* affichage de la chaîne de caractères à l'écran : */
11        System.out.println("Voici le contenu de la variable ma_chaine :");
12        System.out.println(ma_chaine);
13    }
14 }

```

Code source 3

Analyse du *code source 3* :

- * Sur la **ligne 4**, on déclare une variable de type **String**. Cette variable est nommée **ma_chaine**, et pourra contenir une chaîne de caractères.
- * Sur la **ligne 6**, on donne la valeur **"Aujourd'hui, je programme en Java !"** à la variable **ma_chaine**.
- * Et sur la **ligne 8**, on affiche la chaîne de caractères **ma_chaine** à l'écran.

Deuxième exemple de programme utilisant des chaînes de caractères :

```

1 class calcul {
2     public static void main(String[] args) {
3
4         /* déclaration des variables de type String : */
5         String sujet,verbe,complement,espace;
6
7         /* initialisation des 4 chaînes de caractères : */
8         sujet="Toto";
9         verbe="programme";
10        complement="en Java sous Linux";
11        espace=" ";
12
13        /* affichage des chaînes de caractères à l'écran : */
14        System.out.println("La phrase suivante est constituée de 5 chaînes :");
15        System.out.println(sujet+espace+verbe+espace+complement);
16    }
17 }

```

Code source 4

On peut remarquer, sur la ligne 11 de ce *code source 4*, que **pour afficher plusieurs chaînes de caractères avec l'instruction System.out.println, il faut séparer les différentes chaînes de caractères par un signe +**.

Si on exécute le *code source 4*, on obtient l'affichage suivant à l'écran :

```

La phrase suivante est constituée de 5 chaînes :
Toto programme en Java sous Linux

```

Troisième exemple de programme utilisant des chaînes de caractères :

```

1 class calcul {
2     public static void main(String[] args) {
3
4         /* déclaration des variables de type String : */
5         String sujet,verbe,complement,espace,phrase;
6
7         /* initialisation des 4 chaînes de caractères : */
8         sujet="Toto";
9         verbe="programme";
10        complement="en Java sous Linux";
11        espace=" ";
12        phrase=sujet+espace+verbe+espace+complement;
13
14        /* affichage des chaînes de caractères à l'écran : */
15        System.out.println("Voici ma phrase :");
16        System.out.println(phrase);
17    }
18 }

```

Code source 5

Dans ce dernier programme, la chaîne de caractères *phrase* a été initialisée en mettant « bout à bout » les 5 autres chaînes de caractères. Voici l'interprétation de la **ligne 9** du *code source 5* :

```

phrase = sujet+espace+verbe+espace+complement;
⇔ phrase = "Toto" + " " + "programme" + " " + "en Java sous Linux"
⇔ phrase = "Toto programme en Java sous Linux"

```

Après la **ligne 9**, la variable *phrase* vaut donc **Toto programme en Java sous Linux**. On dit que la chaîne de caractère *phrase* est la **concaténation** [c'est-à-dire la mise bout à bout] des chaînes *sujet*, *espace*, *verbe*, *espace* et *complément*. Voici la définition exacte du mot *concaténation*, d'après le dictionnaire : **Enchaînement des éléments constitutifs de plusieurs chaînes de caractères mises bout à bout.**

Si on exécute le *code source 5*, on obtient l'affichage suivant à l'écran :

```

Voici ma phrase :
Toto programme en Java sous Linux

```

Opérations possibles sur les chaîne de caractères :

Les opérations possibles sur les chaînes de caractères, dont nous allons voir une exemple de chaque, sont :

- * Le calcul de la longueur de la chaîne
- * La concaténation de deux chaînes
- * L'extraction d'une sous-chaîne
- * Le calcul de la position d'une sous-chaîne
- * Le remplacement d'une sous-chaîne par une autre
- * La conversion d'une chaîne en majuscule
- * La conversion d'une chaîne en minuscule
- * La conversion d'une chaîne en valeur numérique

Longueur de la chaîne :

Exemple de code source	Contenu des variables après chaque ligne de code	
	s	i
String s; int i;	/* s est de type String */	/* i est de type int */
s="Bonjour"; i=s.length();	Bonjour Bonjour	7
s="Au revoir !"; i=s.length();	Au revoir ! Au revoir !	7 11

Concaténation :

La concaténation consiste à mettre bout à bout plusieurs chaînes de caractères, pour n'en former qu'une seule. En Java, pour concaténer 2 chaînes, on utilise le signe **+** comme le montre les deux exemples suivants :

Exemple de code source	Contenu des variables après chaque ligne de code		
	s1	s2	s3
<pre>String s1,s2,s3; s1="Vive "; s2="Linux !"; s3=s1+s2;</pre>	Vive Vive Vive	Linux ! Linux !	Vive Linux !

Exemple de code source	Contenu de la variable après chaque ligne de code
	s
<pre>String s; s="Je "; s=s+"ne "; s=s+"peux "; s=s+"plus "; s=s+"vivre "; s=s+"sans "; s=s+"Linux !";</pre>	Je Je ne Je ne peux Je ne peux pas Je ne peux pas vivre Je ne peux pas vivre sans Je ne peux pas vivre sans Linux !

Extraction d'une sous-chaîne

Exemple de code source	Contenu des variables après chaque ligne de code	
	s1	s2
<pre>String s1,s2; s1="concaténation"; s2=s1.substring(3,9);</pre>	/* s1 est de type String */ concaténation concaténation	/* s2 est de type String */ caténa

Dans une chaîne de caractères, les caractères sont numérotés à partir de zéro. Voici par exemple la numérotation des caractères de la chaîne "concaténation" :

caractère :	c	o	n	c	a	t	é	n	a	t	i	o	n
n° du caractère :	0	1	2	3	4	5	6	7	8	9	10	11	12

Dans l'exemple précédent, la ligne `s2=s1.substring(3,9);` veut dire « **donner à la variable s2 la valeur de la sous-chaîne constituée des caractères n°3 à n°9 de la chaîne s1** ». On constate que le caractère n°3 est compris dans la sous-chaîne extraite, mais que le caractère n°9 n'est pas compris.

La ligne de code `chaîne2=chaîne1.substring(i,j)` veut donc dire : « **extraire de la chaîne1 la sous-chaîne constituée des caractères n°i [compris] au caractère n°j [non compris], et mettre le résultat dans la chaîne2** »

Autre exemple d'extraction d'une sous-chaîne :

Exemple de code source	Contenu des variables après chaque ligne de code	
	s1	s2
<pre>String s1,s2; s1="Ceci est une chaîne"; s2=s1.substring(13,19);</pre>	/* s1 est de type String */ Ceci est une chaîne Ceci est une chaîne	/* s2 est de type String */ chaîne

Position d'une sous-chaîne (indexOf)

Exemple de code source	Contenu des variables après chaque ligne de code	
	s	i
<pre>String s int i; s="Ceci est une chaîne"; i=s.indexOf("est"); i=s.indexOf("une"); i=s.indexOf("Ceci"); i=s.indexOf("wagon"); i=s.indexOf("une chaîne");</pre>	/* s est de type String */ Ceci est une chaîne Ceci est une chaîne Ceci est une chaîne Ceci est une chaîne Ceci est une chaîne	/* i est de type int */ 5 9 0 -1 9

D'après la valeur de la variable `i`, on peut en déduire que, dans la chaîne `s` :

- * La sous chaîne "est" commence au caractère n°5
- * La sous chaîne "une" commence au caractère n°9
- * La sous chaîne "Ceci" commence au caractère n°0 [c'est-à-dire au début de la chaîne]
- * La sous chaîne "wagon" n'existe pas dans la chaîne `a` [car `b = -1`]
- * La sous chaîne "une chaîne" commence au caractère n°9 [comme la sous-chaîne "une"]

Grâce à cette fonction `indexOf`, on peut facilement savoir si une chaîne contient ou pas une sous-chaîne.

Remplacement d'une sous-chaîne par une autre

Exemple de code source	Contenu des variables après chaque ligne	
	s1	s2
<pre>String s1,s2; s1="Toto programme en Java"; s2=s1.replaceAll("Toto","Titi"); s2=s1.replaceAll("programme en","danse la");</pre>	<pre>Toto programme en Java Toto programme en Java Toto programme en Java</pre>	<pre>Titi programme en Java Titi programme en Java Toto danse la Java</pre>

La ligne `s2=s1.replaceAll("Toto","Titi");` signifie : « Dans la chaîne `s1`, remplacer la sous-chaîne "Toto" par la sous-chaîne "Titi", et mettre le résultat dans la chaîne `s2` »

Grâce à cette fonction `replaceAll`, on peut facilement modifier une partie d'une chaîne de caractères.

Conversion d'une chaîne en majuscules ou en minuscules

Exemple de code source	Contenu des variables après chaque ligne	
	s1	s2
<pre>String s1,s2; s1="Toto programme en Java"; s2=s1.toUpperCase(); s2=s1.toLowerCase();</pre>	<pre>Toto programme en Java Toto programme en Java Toto programme en Java</pre>	<pre>TOTO PROGRAMME EN JAVA toto programme en java</pre>

La fonction `toUpperCase` permet de convertir une chaîne de caractères en lettres **MAJUSCULES**, alors que la fonction `toLowerCase` permet de convertir une chaîne en **minuscules**. Dans l'exemple ci-dessus, le résultat est placé dans la variable `s2`, et le chaîne de départ `s1` n'est jamais modifiée.

Conversion d'une chaîne en valeur numérique

Prenons par exemple la chaîne de caractères "842" : il s'agit simplement des 3 caractères 8, 4 et 2 mis bout à bout, mais cette chaîne de caractères n'a rien à voir avec une valeur numérique [pas plus que la chaîne "Toto"]. Convertir la chaîne "842" [lire *la chaîne huit quatre deux*] en une valeur numérique consiste à la convertir en un nombre réel de type `Float`, et de valeur numérique 842 [lire *huit cent quarante deux*].

Comme le montre le programme suivant, c'est la fonction `valueOf` qui permet de convertir une chaîne de type `String` en un réel de type `Float` :

Exemple de code source	Contenu des variables après chaque ligne	
	s	f
<pre>String s; Float f; s="842"; f=Float.valueOf(s);</pre>	<pre>"842" "842" "842"</pre>	<pre>842.0</pre>

Exercice 2 : Réalisez un programme qui calcule le diamètre, la circonférence, et l'aire d'un cercle, à partir de son rayon. Toutes les variables numériques utilisées seront de type **double**. Le programme invite l'utilisateur à entrer le rayon du cercle, puis affichera à l'écran les 3 renseignements. L'affichage pourra ressembler à cela :

```
Entrez le rayon du cercle en cm, puis appuyez sur la touche Entrée : 4.3
Le diamètre du cercle est : 8.6 cm
La circonférence du cercle est : 27.017696820872194 cm
L'aire du cercle est : 58.0880481704222171 cm carré
```