

concept	définition	pseudo-code	Java		
affectation	attribution d'une valeur (par exemple 5) à une variable (par exemple l'entier a)	$a \leftarrow 5$	<code>a=5;</code>		
affichage	présentation à l'écran d'une valeur entière, d'un flottant, d'une chaîne de caractères ou d'un booléen	Affiche ("toto")	<code>System.out.println("toto");</code>		
algorithme	suite d'instructions pour résoudre un problème, c'est-à-dire fournir un résultat en sortie à partir de variables en entrée. On peut utiliser un algorithme en faisant un "appel d'algorithme" (appel d' Algo2 dans l'exemple ci-contre) en fournissant les entrées adaptées ($entree1$ et $entree2$ dans l'exemple) et en récupérant le résultat renvoyé en sortie (dans la variable $var1$, dans l'exemple).	Algorithme Algo1 Entrées : $entree1$ de type type1, $entree2$ de type type2. Sortie : variable de type type3 (dépend des valeurs de $entree1$ et $entree2$). Variables : $var1$ de type type3. Début $var1 \leftarrow \text{Algo2}(entree1, entree2) + entree2$ renvoyer $var1$ Fin	<pre>public static type3 Algo1(type1 entree1, type2 entree2){ type3 var1; var1=Algo2(entree1,entree2)+entree2; return var1; }</pre>		
appel d'algorithme	utilisation d'un algorithme sur d'éventuelles données en entrée	Algo1 ($entree1$, $entree2$)	<code>Algo1(entree1, entree2);</code>		
booléen	valeur VRAI ou FAUX	Variable : booléen $resultat$ $resultat \leftarrow \text{VRAI}$ $resultat \leftarrow \text{FAUX}$	<pre>boolean resultat; resultat=true; resultat=false;</pre>		
boucle	structure de programmation qui permet d'exécuter plusieurs fois une suite d'instructions	Tant que ... faire : ... Fin TantQue	<pre>while(...){ ...; }</pre>		
chaîne de caractères	ensemble de caractères consécutifs, toujours notée entourée de guillemets droits	Variable : chaîne de caractères $chaîne$ $chaîne \leftarrow \text{"toto"}$	<pre>String chaîne; chaîne="toto";</pre>		
commentaire	ligne d'un code source qui n'est pas exécutée	//Commentaire	//Commentaire		
compilation	création du code exécutable d'un programme à partir de son code source. On compile du code source Java avec le programme <code>javac</code> .	-	-		
concaténation	opération qui réunit en une seule chaîne de caractères deux chaînes de caractères	Concatène ("toto","1") renvoie "toto1"	"toto"+"1" est la chaîne de caractères "toto1"		
déclaration	définition du type d'une variable	Variations : entier $var1$ et $var2$ de type type2.	<code>int var1; type2 var2;</code>		
égalité	être de même valeur	$1=1$ / " $chaîne$ "=" $chaîne$ "	<code>1==1</code> / " <code>chaîne</code> ". <code>compareTo</code> ("chaîne")==0		
entier	en pseudo-code, "nombre entier"; en Java, "entier compris entre -2 147 483 648 et 2 147 483 647"	Variable : entier n $n \leftarrow 42$	<pre>int n; n=42;</pre>		
flottant	nombre à virgule	Variable : flottant pi $pi \leftarrow 3.14156$	<pre>double pi; pi=3.14156;</pre>		
initialisation	première affectation dans une variable	(voir <i>affectation</i>)	(voir <i>affectation</i>)		
instruction	ordre donné à un ordinateur, élément d'un algorithme	une instruction par ligne	instruction toujours terminée par ";"		
longueur	pour un tableau t , nombre de cases; pour une chaîne de caractères c , nombre de caractères	Longueur (t), Longueur (c)	<code>t.length</code> ; <code>c.length()</code> ;		
modulo	(a modulo b) est le reste dans la division de a par b . On peut utiliser modulo pour tester si un entier n est pair.	Modulo (a,b) //Test de parité de n : Si (Modulo ($n,2$)=0)...	<pre>a%b //Test de parité de n : if(n%2==0)...</pre>		
renvoyer	donner le résultat d'un algorithme	renvoyer 5	<code>return 5;</code>		
tableau	ensemble de variables de même type. Le nombre de valeurs stockées (la <i>longueur</i>) ne peut pas être changé après l'initialisation du tableau. La numérotation des cases en Java commence à 0.	Variable : tableau d'entiers t $t \leftarrow \text{NouveauTableau}(2)$ Case ($t,1$) $\leftarrow 3$ Case ($t,2$) $\leftarrow 6$	<pre>int[] t; t=new int[2]; équivalent à : int[] t={3,6}; t[0]=3; t[1]=6;</pre>		
test	structure de programmation qui permet d'exécuter une instruction de manière conditionnelle	Si ... alors : ... Sinon : ... FinSi	<pre>if(...){ ...; } else { ...; }</pre>		
trace	suite des valeurs prises par chaque variable tout au long de l'algorithme	-	-		
type	ensemble de valeurs possibles pour une variable	entier, flottant (nombre à virgule), chaîne de caractères, booléen, couleur, aucun, tableau d'entiers	<code>int, double, String, boolean, Color, void, int[]</code>		
valeur	contenu d'une variable, par exemple entier 2, chaîne de caractères "2", ou tableau à deux cases numérotées 1 et 2	2 "2" <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>2</td></tr></table>	1	2	2 "2" {1,2}
1	2				
variable	élément ayant un nom fixé et un type fixé, qui permet de stocker une valeur, qui peut changer suite à une affectation	(voir <i>déclaration</i> , <i>initialisation</i> , <i>affectation</i> , <i>type</i> , <i>valeur</i>)	(voir <i>déclaration</i> , <i>initialisation</i> , <i>affectation</i> , <i>type</i> , <i>valeur</i>)		