

Annexe 1 : Mots clés java

Mots clés	Explications
boolean	Type booléen (valeurs possibles : <code>true</code> ou <code>false</code>).
break	Sort de la boucle en cours ou d'une instruction <code>switch</code> .
case	Clause d'une instruction <code>switch</code> .
catch	Clause d'un bloc <code>try/catch</code> interceptant un type d'exception.
char	Type caractère Unicode (16 bits).
class	Utilisé pour la définition d'une classe et ainsi créer vos propres objets (Type)
continue	Passe à l'itération suivante d'une boucle sans effectuer les instructions situées après ce mot clé de l'itération en cours.
default	Cas par défaut d'une instruction <code>switch</code> .
Do	Point d'entrée d'une boucle <code>do/while</code> (répéter / jusqu'à en algorithmique).
double	Type nombre flottant en double précision (64 bits).
else	Clause d'une instruction <code>if</code> .
enum	Nouveau type de classes depuis le JDK 1.5. Elle permet d'encapsuler une structure d'objet (comme une classe classique) mais aussi une liste d'instance limitant ainsi ses implémentations. Très utile pour des objets servants de paramètres pour des types de fonctionnement. (voir le chapitre consacré).
extends	Indique qu'une classe hérite d'une et une seule autre.
false	Valeur booléenne contraire à <code>true</code> .
final	Appliqué à une variable, ce mot clé la transforme en constante, dont le contenu ne peut alors plus changer. Utilisé sur une méthode, celle-ci ne pourra plus être surchargée ou redéfinie dans une sous-classe. Sur une classe, elle ne pourra plus être dérivée par une autre classe (exemple la classe <code>String</code>).
finally	Clause d'un bloc <code>try/catch</code> exécutée, qu'il y ait eu déclenchement d'une exception ou non.
float	Type nombre flottant en simple précision (32 bits).
for	Boucle dont le nombre de répétitions est connu (correspond à la boucle pour / fin pour en algorithmique).
if	Instruction conditionnelle (correspond au si / fin si en algorithmique).
import	Importe un package complet ou une classe présente dans un package.
instanceof	Permet de tester si un objet est une instance d'une classe.
int	Type entier (32 bits) : -2 147 483 648 à 2 147 483 647.
interface	Comparable à une classe mais de manière plus abstraite, une interface contient des méthodes devant être redéfinies obligatoirement par la classe implémentant l'interface.
long	Type entier (64 bits) : -9 223 372 036 854 775 808 à 9 223 372 036 854 775 807.
native	Indique que le code qui suit est écrit dans un autre langage que Java (solution non portable).
new	Instancie un nouvel objet ou un tableau.
null	Référence nulle : un objet ayant pour valeur <code>null</code> ne pointe vers aucun objet, il n'a aucune valeur.

Annexe 1 : Mots clés java

Mots clés	Explications
package	Ensemble de classes et d'interfaces regroupés dans un ou plusieurs dossiers et sous-dossiers.
private	Utilisé sur une classe, il en restreint l'accès (uniquement utilisable pour les classes internes). Sur une méthode : on ne peut y accéder qu'à partir de la classe dans laquelle elle est définie. Appliqué à une variable : on ne peut y accéder qu'à partir de la classe dans laquelle elle est définie.
protected	Utilisé sur une classe, il en restreint l'accès à la classe contenant et ses classes filles (Uniquement utilisable pour les classes internes). Sur une méthode : on ne peut y accéder qu'à partir de la classe dans laquelle elle est définie, par les classes filles ou le package dans lequel se trouve la classe contenant la méthode. Appliqué à une variable : on ne peut y accéder qu'à partir de la classe dans laquelle il est défini, par les classes filles ou le package dans lequel se trouve la classe contenant l'attribut.
public	Donne l'accessibilité à l'élément signé, méthode, variable ou classe par toutes les classes.
return	Utilisé dans une méthode pour retourner une valeur ou un objet.
short	Type entier (16 bits) : -32768 à 32767.
static	Appliqué à une variable : instancié une seule fois, l'attribut est commun à l'ensemble des instances d'une classe. Appliqué à une méthode : peut être appelée directement à partir du nom de la classe sans utiliser une instance de la classe.
super	Accède à la super-classe ou à son constructeur : permet d'exécuter le code présent dans la classe mère depuis la classe fille.
switch	Alternative à l'instruction <code>if</code> (les différents cas doivent être connus).
synchronized	Assure l'accès à une méthode ou à un bloc de code à un seul thread (les autres threads doivent alors attendre s'ils veulent accéder à cette partie du code).
this	Utilisé dans une classe pour indiquer l'objet courant.
throw	Lance une exception.
throws	Utilisé pour la déclaration des exceptions que peut lancer une méthode.
transient	Déclare une variable non sérialisable.
true	Valeur booléenne contraire à <code>false</code> .
try	Bloc de code susceptible d'intercepter une exception.
void	Utilisé pour les méthodes qui ne renvoient rien.
volatile	Permet de s'assurer que plusieurs <code>Thread</code> concurrent puissent accéder à la dernière valeur de la variable. Donc, une variable déclarée <code>volatile</code> est accessible par plusieurs <code>Thread</code> simultanément !
while	Boucle dans laquelle il est possible de ne jamais entrer (correspond à tant que / fin tant que en algorithmique).