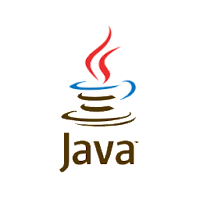
***[](https://1.bp.blogspot.com/-xhlL9DLIuGo/XnkaGiAVjbI/AAAAAAAABJ4/kMJqHT_ld6cGyGxfNwGMxpz8g7JpLHg_ACLcBGAsYHQ/s1600/java.png)***

***Les éléments de base en Java***

**1)  Introduction**  
**==> Programmer c’est quoi ????**  
Programmer signifie réaliser des « programmes informatiques  ». Les programmes  demandent  à l’ordinateur d'effectuer desactions.    
**==> Votre ordinateur est  rempli  de programmes  comme:**

\* la calculatrice est  un programme

\* votre traitement  de texte est  un programme

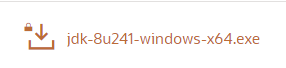
\* votre logiciel  de « chat  » est  un programme

\* les  jeux  vidéo sont  des  programmes

**==> De quels  logiciels  a-t-on besoin pour programmer ?**

**\* Installer les outilles nécessaires: JDK   et  Eclipse IDE**

**JDK :**  
Commencez par télécharger l'environnement Java sur [**le site d'Oracle**](https://www.oracle.com/java/technologies/javase-downloads.html), comme le montre la figure suivante. Choisissez la dernière version stable

[](https://1.bp.blogspot.com/-5cEk3sqD3Xk/XnjS22n2WPI/AAAAAAAABDU/LZCCMKi4fLgKYJqctw5FZeqOVc2oU74PQCLcBGAsYHQ/s1600/2.PNG)

Le Java Development Kit (JDK) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java.  
 Pour L’installation de  **jdk**  est  très simple comme n’importe quel logiciel.  
  
**Eclipse ID :**

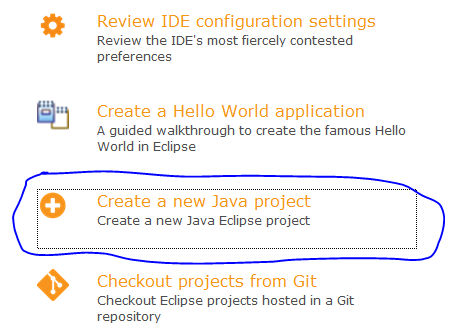
Avant toute chose, quelques mots sur le projet Eclipse. « Eclipse IDE » est un environnement de développement libre permettant de créer des programmes dans de nombreux langages de programmation (Java, C++, PHP…). C'est l'outil que nous allons  utiliser pour programmer.

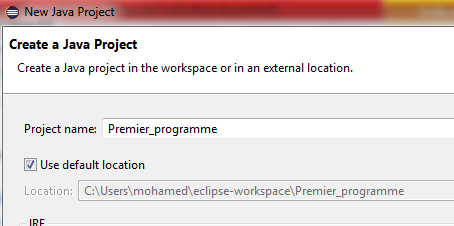
[](https://1.bp.blogspot.com/-0S2uGfiPIGM/XnjTrM8_n1I/AAAAAAAABDg/hZAG-vR1Kwk3R13TUmGeih5JQ7uWGOnEgCEwYBhgL/s1600/Eclipse.png)

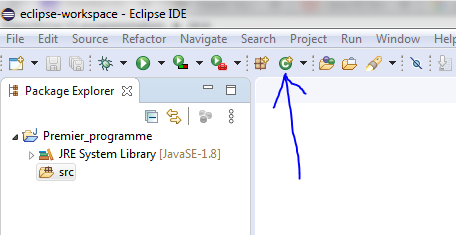
**==> Les étapes pour installer Eclipse**

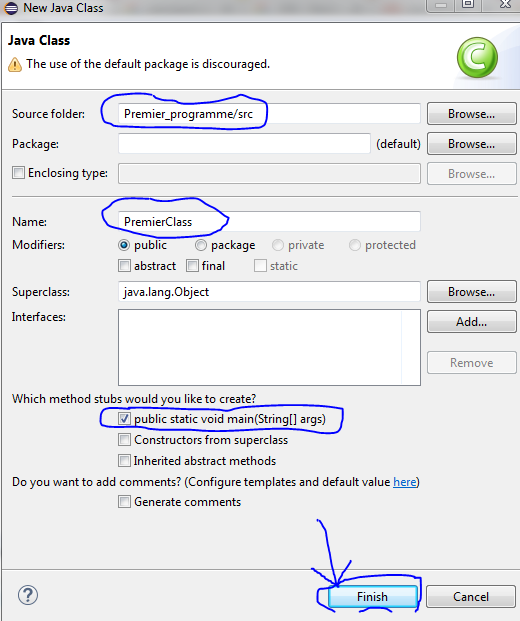
Je vous invite donc à **[téléchargerEclipse IDE](https://www.eclipse.org/" \t "_blank)**. Après  choisissez la version d'Eclipse correspondant à votre OS (*Operating System* = système d'exploitation),  et lors de  l’installation choisissez  **Eclipse IDE for Java Developers**, comme indiqué à la figure suivante.

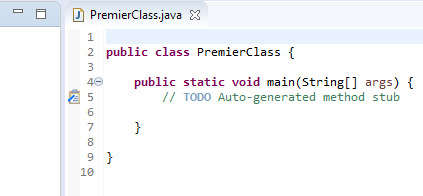
[](https://1.bp.blogspot.com/-Cxv7ADkHkb4/XnjVW4Cin5I/AAAAAAAABDo/zRMH7JFCZOwMmxyONl76Kk6a6oHfV8RPwCLcBGAsYHQ/s1600/3.PNG)

[](https://1.bp.blogspot.com/-8oA2v7KettE/XnjVf8w8PII/AAAAAAAABDs/n056Nim0qFY6-F53btTjV7i3Us6Z7q2AgCLcBGAsYHQ/s1600/4.PNG)

[](https://1.bp.blogspot.com/-tDWALevc0U4/XnjVq9YZL7I/AAAAAAAABDw/aA6esYDy4QIvno3pBoJ4HuBb4UaiQmZ9QCLcBGAsYHQ/s1600/5.PNG)

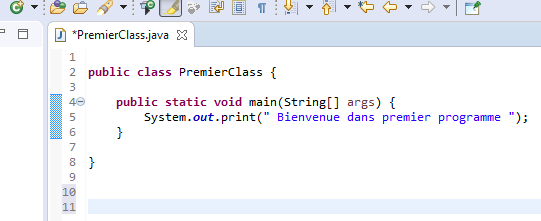
[](https://1.bp.blogspot.com/-F2mE-i0WuQI/XnjVzLIOE_I/AAAAAAAABD0/pD8JNqwSF-oARspwJXSTTGFOGT_-DWg3ACLcBGAsYHQ/s1600/6.PNG)

[](https://1.bp.blogspot.com/--7xl23MlnC0/XnjV5hQfAII/AAAAAAAABD4/OKzHneBwVzUf-NH35aaPUXab8BzYCLr1wCLcBGAsYHQ/s1600/7.PNG)

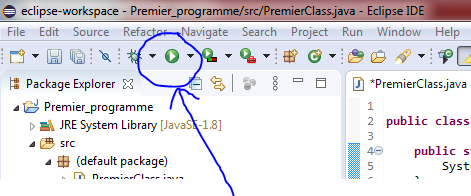
[](https://1.bp.blogspot.com/-vFgzNhJbS3o/XnjWDWJnohI/AAAAAAAABEE/Ngg392BJWp0xWRRRBU70VGvaUFxanaYfwCLcBGAsYHQ/s1600/8.PNG)

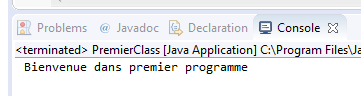
Pour le moment, ce que vous devez retenir, c'est que votre classe est définie par un mot clé (class), qu'elle a un nom (ici, « **PremierClass** ») et que son contenu est délimité par des accolades **{ }**. Nous écrirons nos codes sources entre les accolades de la méthode main. La syntaxe de cette méthode est toujours la même :

**==> Le premier code en java**

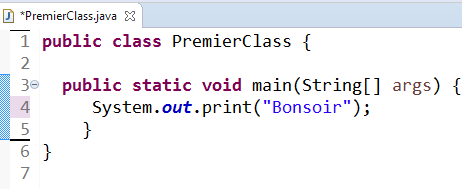
[](https://1.bp.blogspot.com/-3A3LAy1QigE/XnjWMSstqCI/AAAAAAAABEQ/dVymYf_i63gSLhHzbuVqzAB3nF1zTKC2ACEwYBhgL/s1600/9.PNG)

**Pour exécuter ce programme**

[](https://1.bp.blogspot.com/-d21ZI-dXzyY/XnjXu8DHEsI/AAAAAAAABEY/4IWAwq3-ED4DMe4YauYgF45ZBf_GH-6vQCLcBGAsYHQ/s1600/11.PNG)

[](https://1.bp.blogspot.com/-wTDPsa9CLNc/XnjX-qzx8qI/AAAAAAAABEg/sd2V-CXRac49Qx6uJ1G3hd4i_zTMOK4JACLcBGAsYHQ/s1600/10.PNG)

 \*  **public  static void   main ( String []  args )  {   }**  
La fonction ***main***  se trouve dans la classe principale. C'est  un nom  de fonction particulier qui  signifie « principal  ». main est  la fonction principale de votre programme, c'est toujours  par la fonction **main** que le programme  commence.  
Une fonction a  un début  et  une fin, délimités  par des  accolades  { et  }. Toute la fonction main se trouve donc entre ces accolades.  
\* **System.out.print(" Bonjour ") ;**  
Cette ligne à l'intérieur d'une fonction **main** s’appelle instruction, Chaque instruction est une  commande à l'ordinateur de faire quelque chose. Dans ce cas-là  cette ligne demande par l’ordinateur  d’afficher le message Bonjour à l'écran.  
  
**!!!** ***Très  important***: toute instruction se termine obligatoirement par un point-virgule « ; ». Si  vous  oubliez  de mettre un point -virgule à la fin d'une  instruction, votre  programme ne compilera pas !   
  
**==> Écrire  un message**"Bonsoir" **à l'écran**  
À partir de maintenant, on va modifier nous -mêmes  le code de programme précédent. Qui  permet d’afficher le message « Bonsoir » à l'écran 

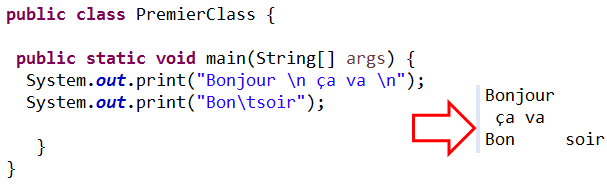
[](https://1.bp.blogspot.com/-HjeEkCsB4JE/XnjaSu_05-I/AAAAAAAABEs/D5W6t1N79KIs-FNjFA8vnDlAziGL3vf8ACEwYBhgL/s1600/16.PNG)

 \* **Les  caractères  spéciaux**  
Les  caractères  spéciaux  sont  des  lettres  spéciales  qui  permettent  d'indiquer qu'on veut aller à la ligne, faire une tabulation, etc.   
Ils  sont  faciles  à reconnaître : c'est  un ensemble de deux  caractères. Le premier     d'entre eux  est  toujours  un anti -slash ( \ ), et  le second un nombre ou une lettre. Voici  deux caractères  spéciaux  courants  que vous  aurez  probablement  besoin d'utiliser, ainsi que  leur signification :

**\n** :  retour à la ligne « Entrée » ;

**\t** :   tabulation

Dans  notre cas, pour faire une entrée, il  suffit  de taper **\n** pour créer un retour à la ligne. Si  je veux  donc faire une tabulation, je devrais  taper **\t**.

[](https://1.bp.blogspot.com/-8z8VR-BVc0U/XnjekF3VblI/AAAAAAAABE4/iVh008KxhAovvJhaRF1Ss0VA3tiDAD_qgCLcBGAsYHQ/s1600/24.PNG)

**==>** **Les  commentaires, c'est très  utile!**  
Lorsqu’un programme devient long, il est conseillé d’ajouter des lignes de commentaires dans le programme, c’est-à-dire des lignes qui ont pour but de donner des indications sur les instructions effectuées et d’expliquer le fonctionnement de programme  sans que le  compilateur ne les prenne en compte.

On va voir  deux  types de commentaires :

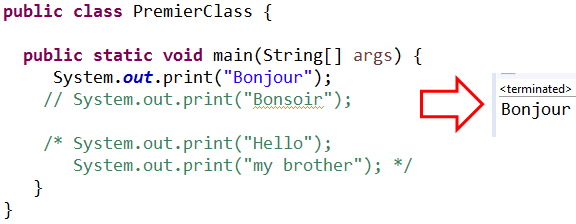
Commentaire sur une ligne :

[https://1.bp.blogspot.com/-EcHERNzH6DY/YKf4786QA4I/AAAAAAAACMo/K6toK3yTFKUZvknXd6nNX8p6rvllWZJMQCLcBGAsYHQ/s0/com.PNG](https://1.bp.blogspot.com/-EcHERNzH6DY/YKf4786QA4I/AAAAAAAACMo/K6toK3yTFKUZvknXd6nNX8p6rvllWZJMQCLcBGAsYHQ/s242/com.PNG)

Commentaire sur plusieurs lignes :

[](https://1.bp.blogspot.com/-HMIkxdd0Dk0/XnjgAby69rI/AAAAAAAABFI/O1g_YAbr4b8nOKDo4FgoKaw6Gl6l61OeQCLcBGAsYHQ/s1600/comment2.PNG)

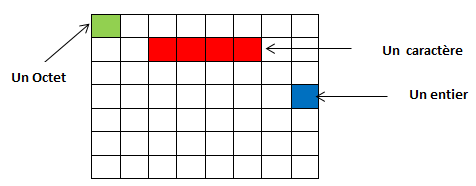
**Remarque :** Parfois on utilise les commentaires pour annuler l’action de quelques  
instructions dans  un programme au lieu de les effacer  comme dans cet exemple:

[](https://1.bp.blogspot.com/-W5qE_ihZZ8A/XnjjGdiP2EI/AAAAAAAABFk/m9EtOWt8VlQAqtgv9G4WyfkCfCe2zpPGACLcBGAsYHQ/s1600/19.PNG)

 Le programme  ignore  l’instruction qui se trouve  entre  le commentaire  et affiche seulement ***Bonjour.***

**2) Les variables et les constantes**

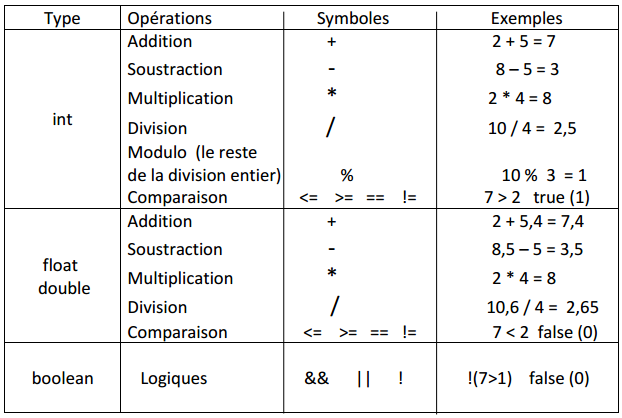
***2-1) Notion de variable***   
Les données  ainsi que les résultats des calculs intermédiaires ou finaux, sont rangés dans des cases mémoires qui correspondent à des variables, Ainsi, une variable  est rangée dans un emplacement mémoire nommé, de taille fixe (ou non) prenant au cours du déroulement de programme un nombre indéfini de valeurs différentes.

[](https://1.bp.blogspot.com/-Njnq_eKqqHk/XnjiTBLaL-I/AAAAAAAABFc/CNaNM4zE_LMWzCIFcRBf62u0fPkH1-JTwCLcBGAsYHQ/s1600/var.PNG)

***2-2) déclaration des variables***  
La partie déclaration consiste à énumérer toutes les variables dont on aura besoin  au cours de programme. Chaque déclaration doit comporter le nom de variable (identificateur) et son type.  
  
**Syntaxe :     type    identificateur   ;**  
 **Identificateur :** Un identificateur est le nom donné à une variable. Ce  nome doit obligatoirement commencer par une lettre suivie d’une suite de  lettres et les chiffres et il  ne doit pas contenir d’espace.  
**Types de données :**  
Le type de variable est l’ensemble des valeurs qu’elle peut prendre.  
**int :** sert à manipuler les nombres entiers positifs ou négatifs. Par exemple :  5   -12    
**float :**  sert à manipuler les nombres à virgule. Par exemple :    5  ,  2.1f  ,  -1.276f  
**double :**  sert à manipuler les nombres à virgule. Par exemple : 5.76543   -1.278 ….  
**char :**  sert à manipuler des caractères, Par exemple :   'A'   ,  '6'   
**boolean:**utilise les expressions logiques. Il  n’y a que deux valeurs booléennes : **true  /  false**  
**String :**  sert à manipuler des chaînes de caractères, Par exemple :   'bonjour'   ,  "hi  cv"   
**Exemple :**

int N  ;          float  e , r  ;             boolean     proposition ;             char   letter ;

***==> Les opérations sur des variables :***

[](https://1.bp.blogspot.com/-ymDMXIBtVDw/XnjoGUaMEuI/AAAAAAAABFw/abwB7Ss3wiAsXqK3lQDlh58_egzN0QkHwCLcBGAsYHQ/s1600/operations.PNG)

**Remarque:**

**&&** signifie :  et

**||**   signifie :  ou

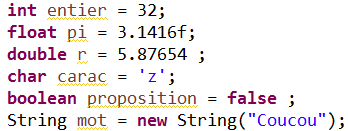
**!**  signifie :   non  (négation)

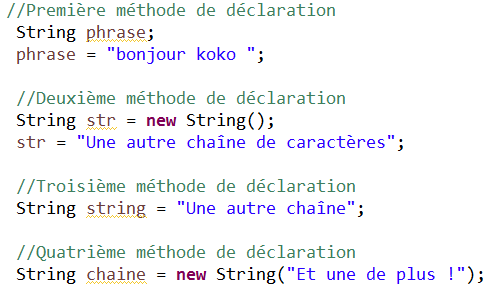
**==**  signifie :  égal

**!=**  signifie :    **≠**  
  
**\***Lorsque l’on écrit directement une valeur flottante (**float**) dans le code, elle est considérée par défaut comme un double. Si on veut réduire sa précision pour qu’elle représente un float, il faut explicitement lui rajouter un  **F**  
**\***Le type **String**  permet de gérer les chaînes de caractères, c'est-à-dire le stockage         de texte.  
Il s'agit d'une variable d'un type plus complexe que l'on appelle *objet*. Vous verrez que   celle-ci  s'utilise un peu différemment des variables précédentes  
***2-3) Les constantes***  
Comme une variable, il existe une  constante  correspond un emplacement mémoire  réservé auquel on accède par le nom qui lui a été attribué, mais dont la valeur stockée ne  sera jamais modifiée au cours du programme.   
  
**Syntaxe :**     **final  Nom\_Constante = valeur ;**  
Exemple :    final   float   pi = 3.14 ;

**3) Les instructions de base**

Une instruction est une  action élémentaire commandant à la machine un calcul, ou une  communication avec l’un de ses périphériques d’entrées  ou de sorties. Les instructions de base sont:   
***3-1) L’instruction d’affectation :***  
L’affectation permet d’affecter une valeur à une variable. Elle est symbolisée  par « = »      
**Syntaxe** :      Variable   =  Expression **;**  
 ***Exemple :***

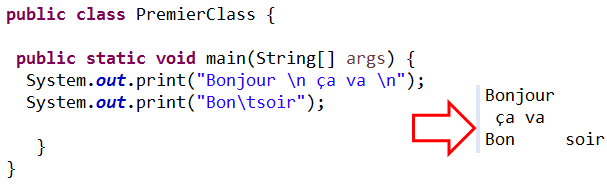
[](https://1.bp.blogspot.com/-zR5dAuKQVYo/XnjqmMJ5KvI/AAAAAAAABF8/UkBrtMFMbDgYytiRjz6GDjP45RhKKaE_ACLcBGAsYHQ/s1600/26.PNG)

[](https://1.bp.blogspot.com/-VksVslP5X34/Xnjq1cCA0rI/AAAAAAAABGA/84IRUjIU4l84cOHo9DwRW-Wbnf8x_PXZwCLcBGAsYHQ/s1600/25.PNG)

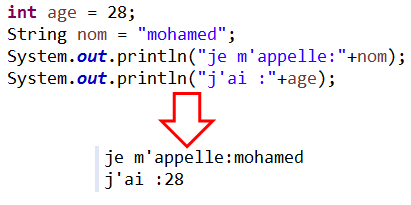
 ***3-2) L’instruction de sortie :***  
Avant de lire une variable, il est conseillé d’écrire un message à l’écran, afin de prévenir l’utilisateur de ce qu’il doit taper. L’instruction de sortie (d’écriture) permet d’afficher des informations  à l’écran.   
**Syntaxe :**  
***System.out.print (" message à l'écran ") ;***  ou

***System.out.print ("  message " + valeur) ;***

**\* System**: ceci correspond à l'appel d'une classe qui se nomme « System ». C'est une classe utilitaire qui permet surtout d'utiliser l'entrée et la sortie standard, c'est- à-dire la saisie clavier et l'affichage à l'écran.  
\* **out**: objet de la classe System qui gère la sortie standard.  
\* **print**: méthode qui écrit dans la console le texte passé en paramètre (entre les parenthèses).  
**Exemple 1:**  
***System.out.print(" Bonjour ! ");***  
***System.out.print(" je m’appelle ");***  
***System.out.print("mohamed");***  
Lorsque vous l'exécutez, vous devriez voir des chaînes de caractères qui se suivent sans saut de ligne. Autrement dit, ceci s'affichera dans votre console comme ceci :  
***Bonjour ! je m’appelle mohamed***  
cela, vous avez plusieurs solutions :  
\* soit vous utilisez un caractère d'échappement, ici **\n** ;  
\* soit vous utilisez la méthode **println()** à la place de la méthode **print().**

[](https://1.bp.blogspot.com/-koOWAqlQp-E/Xnjs2NcQS7I/AAAAAAAABGQ/zCy_lXuozQE5wJdoZCbSoUfJ1UTsoWWGACLcBGAsYHQ/s1600/24.PNG)

**Exemple 2:**

**[](https://1.bp.blogspot.com/-uT2ctqbFKhI/XnjtYY9WMII/AAAAAAAABGY/WZb61AQisTsuHRetk9HOplLtgzh9e3abQCLcBGAsYHQ/s1600/56.PNG)**

***3-3) Les conversions  ou  < cast >***  
En programmation, on est souvent amené à convertir lestypes,  c'est-à-dire passer d'un type numérique à une chaîne de caractères ou vice-versa  Nous allons voir un truc super important en Java. Ne vous en déplaise, vous serez amenés à convertir des variables.  via la syntaxe suivante :    ***variable  = (type) valeur***

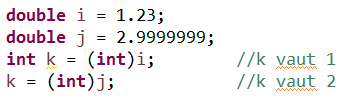
***D'un type int en type float :***

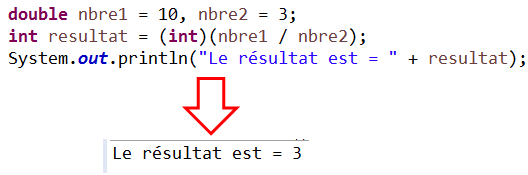
***[https://1.bp.blogspot.com/-ReNviMdK5c0/XnjuMWHjOaI/AAAAAAAABGk/62d-YSSBK3oKI9STEEsfvrXzOKZrc_2tQCLcBGAsYHQ/s1600/27.PNG](https://1.bp.blogspot.com/-ReNviMdK5c0/XnjuMWHjOaI/AAAAAAAABGk/62d-YSSBK3oKI9STEEsfvrXzOKZrc_2tQCLcBGAsYHQ/s1600/27.PNG)***

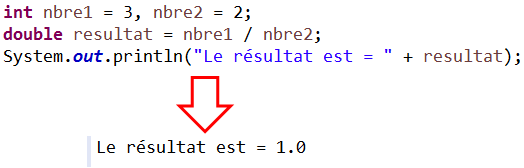
***D'un type double en type int :***

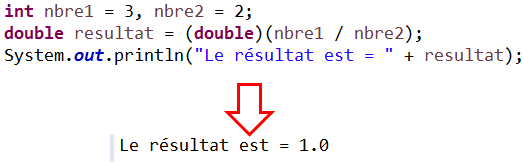
[https://1.bp.blogspot.com/-obG9b4_DIU0/XnjuYA7Lj_I/AAAAAAAABGo/hATk8Q38BPUZrevemres0KZvqIuEAjFcQCLcBGAsYHQ/s1600/28.PNG](https://1.bp.blogspot.com/-obG9b4_DIU0/XnjuYA7Lj_I/AAAAAAAABGo/hATk8Q38BPUZrevemres0KZvqIuEAjFcQCLcBGAsYHQ/s1600/28.PNG)

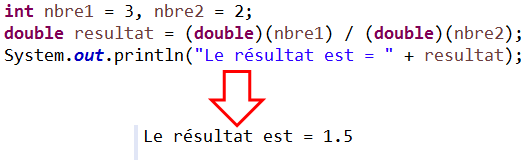
***Exemples:***

[](https://1.bp.blogspot.com/-ivhboglLM-Y/XnjuoIJyv-I/AAAAAAAABGs/8TOfq82SQsYuvfz4QNEHGINDNuBgVEFFwCLcBGAsYHQ/s1600/29.PNG)

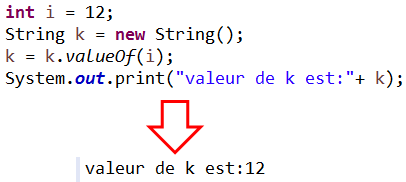
[](https://1.bp.blogspot.com/-iydqgYkNUTA/XnjvJWrh80I/AAAAAAAABG8/JKSwIcyW_UotFFCdUfo0lyCXTmmcnB8mQCLcBGAsYHQ/s1600/32.PNG)

[](https://1.bp.blogspot.com/-7sw5_4A3-EA/XnjvhW8AuHI/AAAAAAAABHI/AVsea_PnATIjjtGFyt07L_GpxoWyQv2QACLcBGAsYHQ/s1600/35.PNG)

[](https://1.bp.blogspot.com/-apEPS-vaFL0/Xnjvsw6QzkI/AAAAAAAABHQ/iD_sFRX_g5MdSl6z4M4OtYFJxHRpb_sEQCLcBGAsYHQ/s1600/38.PNG)

[](https://1.bp.blogspot.com/-Xomz5_1bbuI/XnjwAugxNnI/AAAAAAAABHc/DtlzTamT5LMhvp3zwnOU3bEkdMulfJt6wCLcBGAsYHQ/s1600/41.PNG)

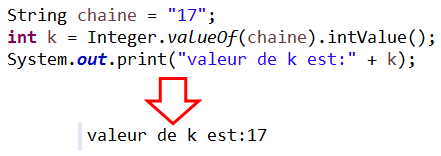
**Remarque**  
**==>**pour convertir n’importe quel type de données en une chaîne, on utilise la fonction ***valueof*** 

[](https://1.bp.blogspot.com/-8mzvsT6KwJ0/XnjyNTg2MZI/AAAAAAAABHw/rcuUIch-r1QCXpdIkCV6l1GFe5ktQUZOgCLcBGAsYHQ/s1600/50.PNG)

**!!!** La valeur 12 est considérée  comme une chaîne, ce n’est pas une valeur numérique

**==>** pour convertir une chaîne  en  un type de données numérique, on utilise les fonctions  ***intValue()  ,     floatValue() ,     doubleValue()***

**Exemple :**

**[](https://1.bp.blogspot.com/-JhFaTy194UA/Xnjyb_ivCLI/AAAAAAAABH0/QalquYculiAoNPX4q1DsVr2rp7gt_CLSwCLcBGAsYHQ/s1600/53.PNG)**

**!!!** La valeur 17 de variable k  est considérée  comme une valeur numérique, ce n’est pas  une chaîne.

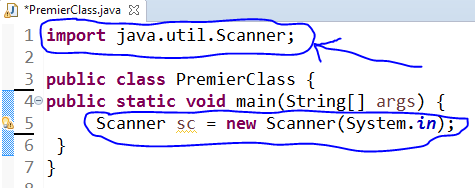
***3-4) Les instructions d’entrée :***

Les instructions d’entrée ou de lecture donnent  la main à l’utilisateur pour saisir une donnée au clavier. La valeur saisie sera affectée à une variable. Ces instructions sont  effectuées via un objet de la class **Scanner.**

***Scanner sc = new Scanner(System.in) ;***

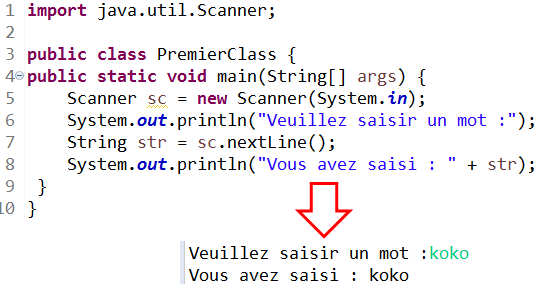
cette instruction ne fonctionne  pas pourquoi ????

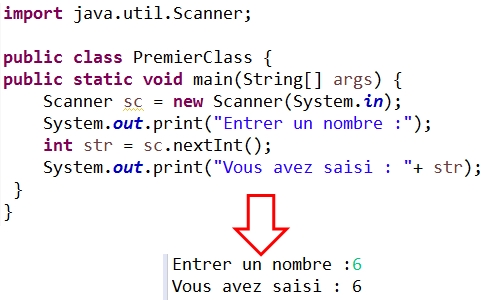
il faut importer la classe Scanner  comme indiquée à la figure suivante :

[](https://1.bp.blogspot.com/-05m7cOVRMNg/XnjzFfqIfCI/AAAAAAAABH4/30G2uL3AS38WlF4ZlGW7xgfRT9gF0dpjACLcBGAsYHQ/s1600/58.PNG)

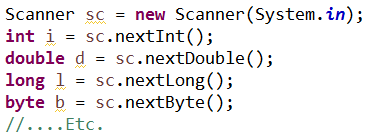
**\* Récupérer  ce que vous tapez**

Voici l'instruction pour permettre à Java de récupérer ce que vous avez saisi pour ensuite l'afficher :

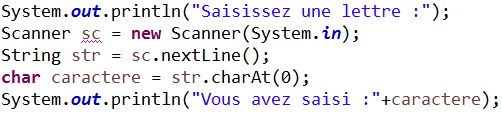
[](https://1.bp.blogspot.com/-ps2edsLOtbI/Xnj11V4PfmI/AAAAAAAABIM/PYbP0g7tMSEkWdVemkms4dB8T2B5syIvgCLcBGAsYHQ/s1600/61.PNG)

[](https://1.bp.blogspot.com/-5jpldFUC1Wo/XnkPSZ8FYaI/AAAAAAAABIY/2_mllGCVluc7rfXMGD0fm2MhFNqzpEk8ACEwYBhgL/s1600/64.PNG)

 Vous savez maintenant que pour lire un int, vous devez utiliser nextInt(). De façon       générale,dites-vous que pour récupérer un type de variable, il vous suffit d'appeler **next<Type de variable commençant par une majuscule>**

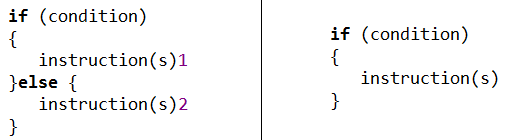
[](https://1.bp.blogspot.com/-nan81Zf1_2A/XnkQIOp5ghI/AAAAAAAABIg/IJN6dvw0fjk1VcMqs1auZE-MUKuZCkongCLcBGAsYHQ/s1600/65.PNG)

***!!! Il y a un type de variables primitives qui n'est pas pris en compte par la classe Scanner : il s'agit du type char.***  
Voici comment on pourrait  récupérer un caractère :

[](https://1.bp.blogspot.com/-e0ZlV_W9ong/XnkRrfbsQBI/AAAAAAAABIs/N01ZGDMdd60L-bynsAG0BVETsViAbByxgCLcBGAsYHQ/s1600/char.PNG)

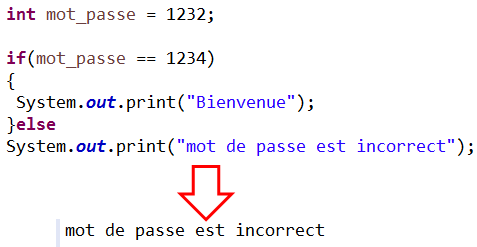
**Chapitre 2 : Les  structures alternatives et répétitives**

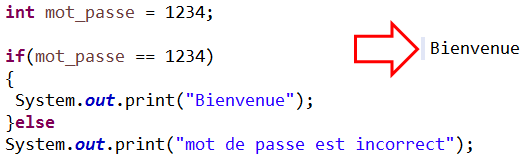
**1) Les structures alternatives**  
***1-1) Introduction :***  
**Syntaxe :**

[](https://1.bp.blogspot.com/-GiOR3vUfbT0/XnkSnLYcFeI/AAAAAAAABI0/QS8oSuxlesgHz0JQy9mxk0wNVt_n8pPfACLcBGAsYHQ/s1600/70.PNG)

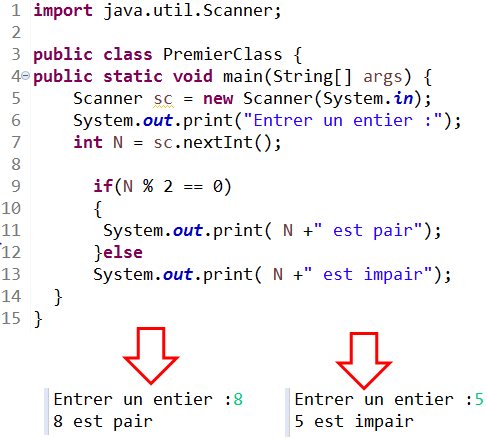
 Une condition est une expression  logique  ou une variable logique évaluée à  Vrai ou faux.  
La condition est évaluée. Si  elle est vraie, la série d’instruction(s)1 est exécutée et l’ensemble d’instruction(s) 2 est ignoré, la machine sautera directement à la première instruction située après cette structure.   
De  même, au cas où la condition  était fausse la machine saute directement à la première  
ligne située après le  **else** et exécute l’ensemble d’instruction2.

**Exemple :**

[](https://1.bp.blogspot.com/-W3qcpnBnxMU/XnkWHby6hxI/AAAAAAAABJA/AdZyVkRVakY_l8skDQJNCjDhMWwlCDUAACLcBGAsYHQ/s1600/71.PNG)

[](https://1.bp.blogspot.com/-e9ZZpRI3yx4/XnkWZqqBh4I/AAAAAAAABJE/ccg2Qg0a3IYs6mOX0JnEAn8AvD1EVzumACLcBGAsYHQ/s1600/72.PNG)

**Exercice d’application 1**  
Ecrire un programme en java qui affiche si un nombre entier  saisi au clavier est pair ou impair.

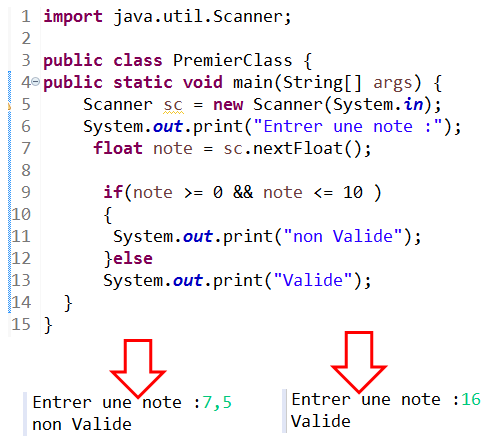
[](https://1.bp.blogspot.com/-3SJ9oPs-NZo/XnkW1Q3wgfI/AAAAAAAABJI/y6ISQPBg7-I-GF7bqa2QzqtXH-ITgNEkACLcBGAsYHQ/s1600/76.PNG)

**Remarque :** il existe aussi un autre type de condition c’est la condition composées.

Certains problèmes exigent  de formuler des conditions qui ne peuvent être exprimées sous la forme simple, par exemple la condition  de  note  de devoir doit être inclus dans  l’intervalle     [0, 20],  cette  condition est composée de deux conditions simples qui  sont  note ≥0    et     note ≤ 20

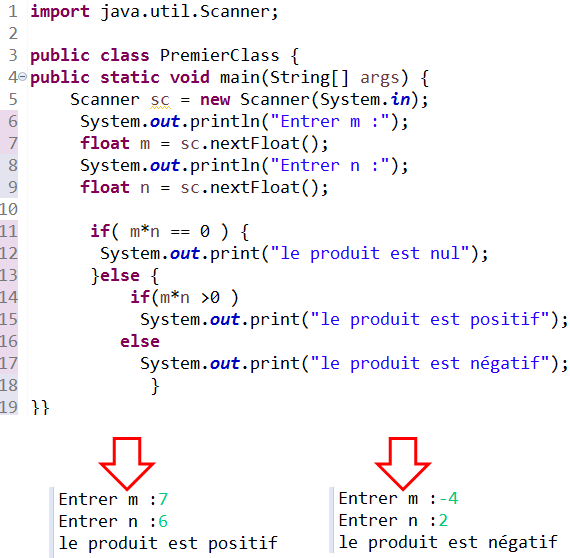
**Exercice d’application 2**

Ecrire un programme en java qui permet de faire  une remarque d’une note saisi au clavier ( si la note  supérieur  à 10  alors  affiche  validé sinon non validé  (NB : la note comprise entre 0 et 20 ! )**.**

[](https://1.bp.blogspot.com/-l3Lyg2N8jfY/XnkXnHUZnVI/AAAAAAAABJc/NhGYkwg1_agBg4pY8GU7H4LE3BdEa6uIQCLcBGAsYHQ/s1600/80.PNG)

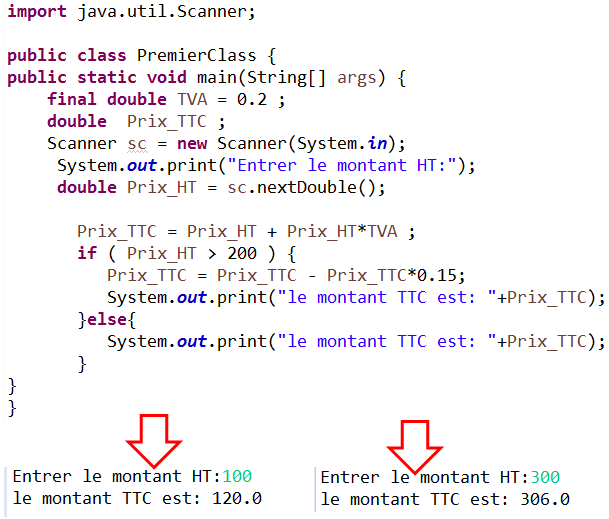
 **Exercice 1:**

Ecrire un programme en java qui demande deux nombres m et n à  l’utilisateur et l’informe ensuite si le produit  de ces deux nombres est positif ou négatif. On inclut   dans le programme  le cas  où le produit peut être nul.

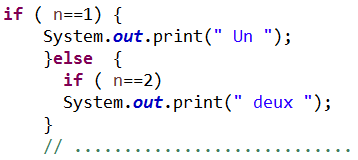
[](https://1.bp.blogspot.com/-9-Cm_yF1YRI/XnkYIhIfa5I/AAAAAAAABJk/CyRYsqtMk18f5pmDlaeJvXReS4uSNuvKwCLcBGAsYHQ/s1600/85.PNG)

**Exercice 2:**

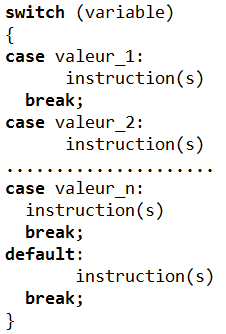
Une boutique propose à ces clients, une réduction de 15% pour les montants d’achat supérieurs à 200 dh. Ecrire un programme en java permettant de saisir le prix total HT et de calculer le montant TTC en prenant en compte la réduction et la TVA=20%

[](https://1.bp.blogspot.com/-2Ey5OjIPpd4/XnkYjOgvb3I/AAAAAAAABJo/3ntJVBopKdIYJhQZLia9l1Wmq6mWCthYACLcBGAsYHQ/s1600/89.PNG)

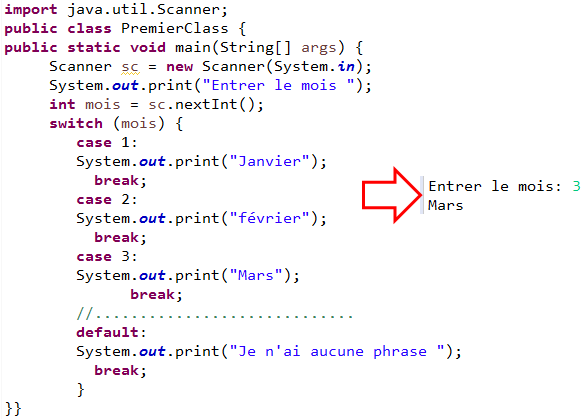
***1-3)  Structure à choix multiples   ( Switch )***  
La condition  **if... else**  que l'on vient de voir est le type de condition le plus souvent utilisé. Toutefois, le  **if... else**  peut s'avérer quelque peu… répétitif : Prenons cet exemple :

[](https://1.bp.blogspot.com/-pZvOkmAlpf4/XnkcHT7SdLI/AAAAAAAABKE/aEI6UFlCbfUUlcqe0MhkVvVjbzeVpr9IACLcBGAsYHQ/s1600/90.PNG)

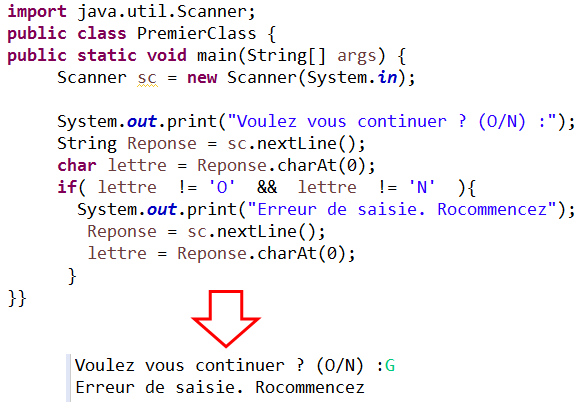
Pour résoudre ce problème de la répétition de  **if  else   if  else** …. plusieurs fois. On utilise une nouvelle structure  à choix multiples***.*** Cette  structure conditionnelle permet de choisir le traitement à effectuer en fonction  de la valeur ou de l’intervalle de valeurs d’une variable  ou d’une expression.  
**Syntaxe :**

[](https://1.bp.blogspot.com/-WIYhF14Zis0/XnkdCsTj4TI/AAAAAAAABKM/nvce3S_dCc0PHTJPu8TvGITBDK-NlOLvgCLcBGAsYHQ/s1600/switch.PNG)

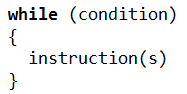
 Lorsque l’ordinateur rencontre cette instruction, il vérifie la valeur de la variable, et il la  compare  aux différentes valeurs.   
Les valeurs sont évaluées  dans l’ordre, les unes après les autres, et une fois la valeur de variable est vérifiée l’action associée  est exécutée. On peut utiliser une instruction **default**   (facultative), dont l’action sera exécutée si aucune des valeurs évaluées n’a pas été remplie.  
  
**Exercice d’application :**  
Ecrire un programme en java  permettant d’afficher le mois en lettre selon le  numéro saisi  au clavier       (janvier=1  ,  février =2  …….. décembre = 12).

[](https://1.bp.blogspot.com/-7aqRlodqfK8/Xnkd-cE6tdI/AAAAAAAABKY/zch00Q965XY-lmwztQ8aQuoytvxFi2S6QCLcBGAsYHQ/s1600/93.PNG)

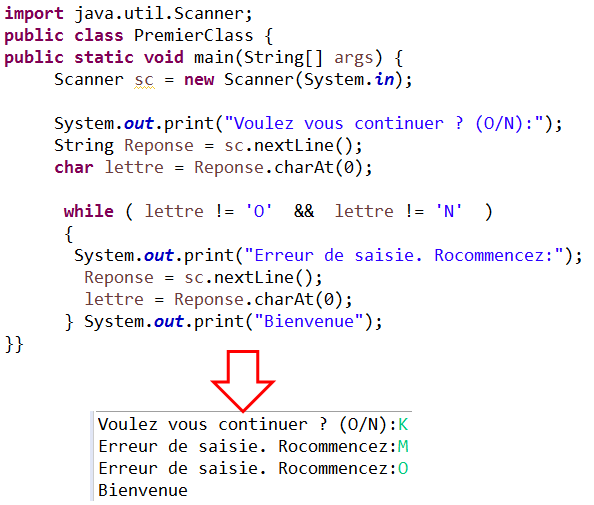
**Remarque :** Vous devez mettre une instruction **break ;**   à la fin de chaque cas. Pour ignorer les instructions en dessous ! L'instruction **break ;**  commande en fait à l'ordinateur de «sortir » des accolades. Et  instruction  **defaulf ;**   sera exécutée si aucune des valeurs  évaluées n’a pas été remplie.  
**2) Les Structures  répétitives**  
**2-1) Introduction**  
Prenons d’une saisie au clavier, par exemple, on pose une question à laquelle doit répondre  par « oui » ou « non ».  
L’utilisateur risque de taper autre chose (une autre lettre), le programme peut soit planter  par une erreur d’exécution soit dérouler normalement jusqu’au bout, mais en produisant  des résultats fantaisistes.   
Pour éviter ce problème, on peut mettre en place un contrôle de saisie pour vérifier que les  données entrées au clavier correspondent bien à celles attendues par le programme.

[](https://1.bp.blogspot.com/-qATQShdrMIc/Xnkewof_cvI/AAAAAAAABKg/Jhi-sMGNZAkUhY5hc0ZMObfeAJqVx2DIwCLcBGAsYHQ/s1600/96.PNG)

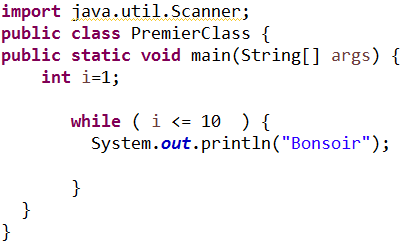
**!!!**Le programme  ci-dessus résout le problème si on se trompe qu’une  seule fois, et on fait entrer une valeur correcte à la deuxième demande. Sinon en cas de deuxième  
 erreur, il faudrait rajouter un  «  **if**  ». Et ainsi de suite, on peut rajouter des  centaines  de «  **if**  »  
**==>** La solution à ce problème consiste à utiliser une structure répétitive.   
**Une structure répétitive**, encore appelée **boucle**, est utilisée quand une instruction ou une liste d’instruction, doit être répétée plusieurs fois. La répétition est soumise à une condition.  
**2-2) La boucle while**  
La boucle **while**  permet de répéter un traitement tant que la condition est vraie.  
**Syntaxe:**

[](https://1.bp.blogspot.com/-q7Jn0zpIQX4/Xnkhh_K93hI/AAAAAAAABKs/u3pTSxkAXO4GOwS1OzwiRIMVrGhqjpX4wCLcBGAsYHQ/s1600/b.PNG)

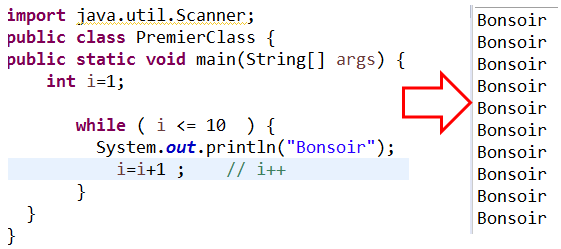
 \* L’exécution de la boule dépend de la valeur de la condition. Si est vrai, le programme exécute les instructions qui suivent, jusqu’à ce qu’il  rencontre l’accolade **}**. Il retourne ensuite sur la ligne du **while**, procède au même examen, et ainsi de suite  
  
\* La boucle ne s’arrête que lorsque prend la valeur fausse, et dans ce cas le programme poursuit son exécution après  la boucle  « sortir des accolades ».  
  
**Exemple:**

[](https://1.bp.blogspot.com/-2mgw4F5qb50/XnkjGT6XqjI/AAAAAAAABK4/MXL2YkEuCeMG1zQ7IHNz4pO8D_jrfOPtwCLcBGAsYHQ/s1600/99.PNG)

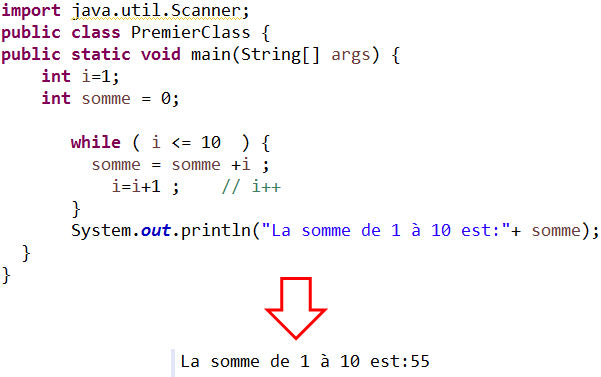
**Remarque :**  
Si la  structure   **while**   contient la condition ne devient jamais fausse. Le programme tourne dans une boucle infinie et n’en sort plus.  
**Exemple :**

[](https://1.bp.blogspot.com/-TFIjtqS9rFs/Xnkj1T1PpUI/AAAAAAAABLA/mn2yZxXf0kEAkoc_X7FUz1IK0ljW5V6SwCLcBGAsYHQ/s1600/100.PNG)

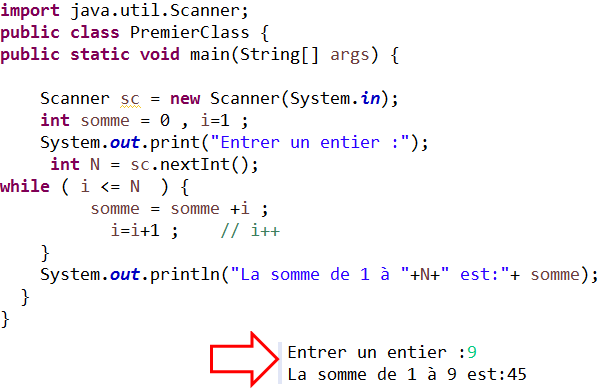
  Dans cet exemple nous avons  une boucle infinie. L’ordinateur ne s’arrêtera jamais  d’afficher le message ***Bonsoir*** car la variable **i** qui est testée dans la condition n’est jamais incrémenter. Alors pour  résoudre  le problème de boucle infinie dans ce cas-là, on  incrémente la variable i à chaque tour de boucle  pour afficher   ***Bonsoir*** 10 fois  exactement.

[](https://1.bp.blogspot.com/-KrA2OHH18L4/XnknsgqbveI/AAAAAAAABLM/qun0pgPnctwahcv4SMKhzqWB-JyE0RZ_QCLcBGAsYHQ/s1600/103.PNG)

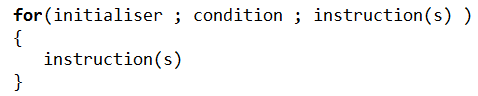
**Exercice 1**  
Ecrire un programme en java  qui calcule la somme  S = 1+2+3+4+……..+ 10. Utilisant la boucle while.

[](https://1.bp.blogspot.com/-XfahQprgaSc/XnkoBpIluOI/AAAAAAAABLU/bBp4EzzYv_svXmVrstOUgcpPLkohAzDeACLcBGAsYHQ/s1600/106.PNG)

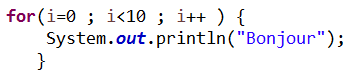
**Exercice 2**  
Ecrire un programme en java qui calcule la somme   S= 1+2+3+……..+ N,  où N saisi par l’utilisateur.

[](https://1.bp.blogspot.com/-PPEF2kErGPE/Xnkoab8acEI/AAAAAAAABLc/9MyKCOvy94YBXWauHQiH-zJ0gTzN5PM-gCLcBGAsYHQ/s1600/109.PNG)

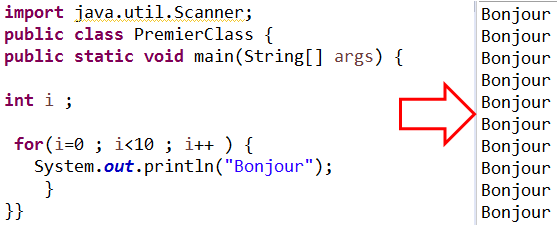
**2-3)  La boucle  for**  
La boucle  ***for***  permet de répéter une liste d’instructions un nombre connu de fois.  
**Syntaxe :**

[](https://1.bp.blogspot.com/-fU0v1bfjOJY/XoZY41gYFXI/AAAAAAAABkA/KAiLwJSaTr4tglF9o2oCYZos6z3X2oOyQCLcBGAsYHQ/s1600/01.PNG)

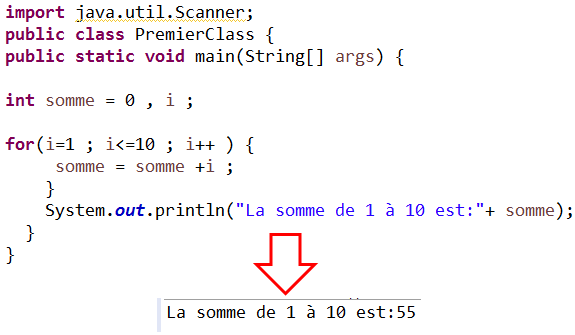
**Exemple:**

[](https://1.bp.blogspot.com/-xvHevZodkjE/XoZZB-ChuyI/AAAAAAAABkE/XSuqeiZnzUgXPT2ZwyJAXQiyG_mizgBRQCLcBGAsYHQ/s1600/02.PNG)

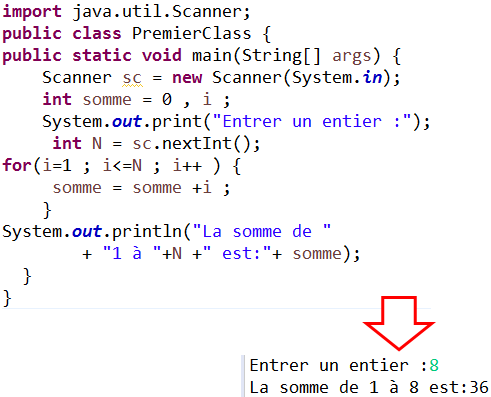
Cet exemple  permet d’afficher le message bonjour 10  fois, et tant que la condition est vraie la boucle répète l’exécution des instructions qui se trouvent  entre les accolades   
**==>** La partie d’**initialisation**  est de type entier. Elle est initialisée par la valeur initiale, la partie **instruction(s)** augmente cette valeur par  1 automatiquement à  chaque tour de boucle  jusqu’à la valeur finale.  
**==>**Lorsque la valeur compteur vaut la valeur finale, le traitement est exécuté une seule fois puis le programme sort de la boucle.  
  
**Exemple:**

[](https://1.bp.blogspot.com/-NjY0KDzsdjg/XoZZ4kG77dI/AAAAAAAABkM/XkCUcGniWCUSsHzYDf911vi124sSa5zTACLcBGAsYHQ/s1600/113.PNG)

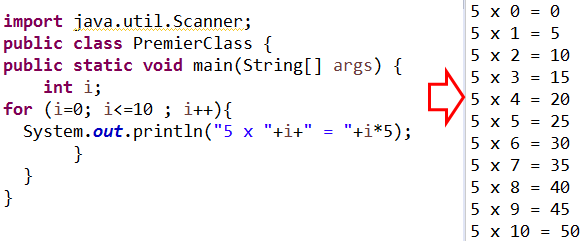
**Exercice 1**   
Ecrire un programme en java  qui calcule   S= 1+2+3+4+……..+ 10. Utilisant la boucle for.

[](https://1.bp.blogspot.com/-y9lprHnryzk/XoZaOdKgGZI/AAAAAAAABkU/sDIxKwg4S5QWpWpWaOKiVQhVvq6mlmyRwCLcBGAsYHQ/s1600/116.PNG)

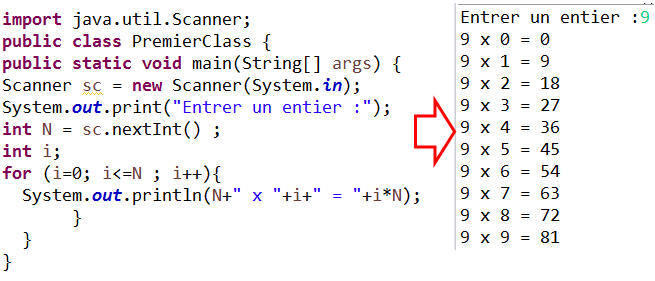
**Exercice 2**   
Ecrire un programme en java qui calcule  S= 1+2+3+4+……..+ N. Utilisant la boucle for.

[](https://1.bp.blogspot.com/-Oui6vBW4Fyw/XoZaiqCk9CI/AAAAAAAABkg/-a5lCbjIoQ4Vc6X9pKNuWI-qRQJF37OYQCLcBGAsYHQ/s1600/119.PNG)

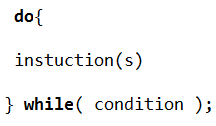
**Exercice 3**  
Ecrire un programme en java qui affiche la table de multiplication de 5. Utilisant la boucle for.

[](https://1.bp.blogspot.com/-FQum7cDGM0A/XoZa3FBIaKI/AAAAAAAABko/mvUbkZNgUyYjnGVpMV4W1QJwLT8gzFatwCLcBGAsYHQ/s1600/122.PNG)

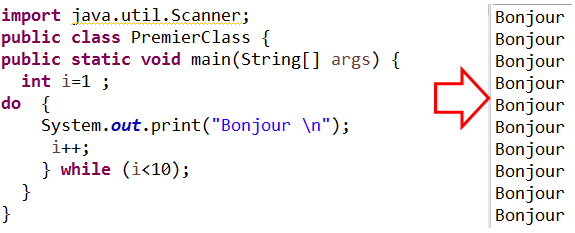
**Exercice 4**  
Ecrire un programme en java qui affiche la table de multiplication d’un  entier saisie par l’utilisateur, Utilisant la boucle for

[](https://1.bp.blogspot.com/-AwFds62RZkc/XoZbQC6VOKI/AAAAAAAABkw/ZEQF5ZBiX6gveH-qCAl2xmW2daLmjt7_ACLcBGAsYHQ/s1600/125.PNG)

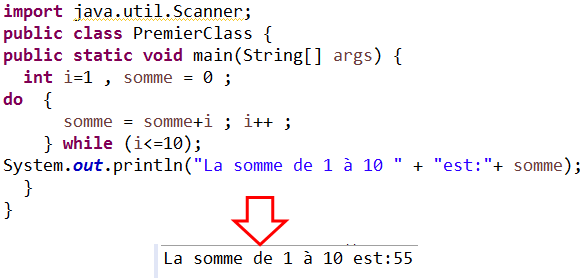
**2-4) La  boucle   do ….While**  
Cette boucle permet de répéter les instructions qui se trouvent entre les accolades tant que la condition est vraie.  
**Remarque :**  
Cette boucle ne s’utilise en général que pour des menus, elle est dangereuse  car il n’y a pas de vérification de la condition avant  d’y entrer   
Ce type de boucle est  très  similaire à while, bien qu’un peu moins  utilisé en général. La seule chose qui  change en fait  par rapport  à while, c'est  la position de la condition.  Au li eu d'être au début  de la boucle, la condition est  à la fin:  
**Syntaxe :**

[](https://1.bp.blogspot.com/-PJ--0L5KjVo/XoZdFn9b5jI/AAAAAAAABk8/gOlPOVgU_BEHS8xWpWZiFCJ06kFfVJVOgCLcBGAsYHQ/s1600/03.PNG)

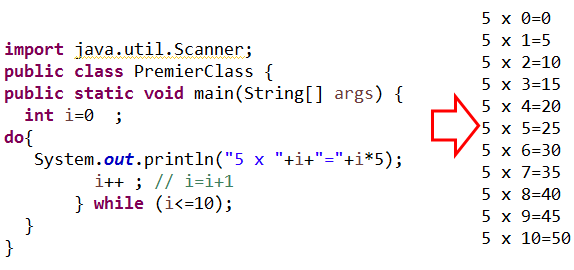
la boucle **while**  pourrait  très  bien ne jamais  être exécutée si  la condition est  fausse dès  le départ. Par exemple, si la condition est fausse  dès  le début  et  on ne serait  jamais  rentré  dans  la boucle. et Pour la boucle **do… while**, c'est  différent: cette boucle s'exécutera toujours  au moins  une fois même si la condition est fausse.  
  
Il  est  donc  parfois  utile de faire des  boucles  de ce type, pour s'assurer que l'on rentre au moins  une fois  dans  la  boucle.  
  
**!!!** N'oubliez  pas  de mettre un point-virgule  après  le while, si non votre programme plantera à la compilation !  
  
**Exemple:**  
En utilisant  la boucle do…..while, on écrit  un programme  qui affiche ***Bonjour*** 10 fois.

[](https://1.bp.blogspot.com/-ux3aUjY9bR8/XoZeFSO6zCI/AAAAAAAABlE/vhj5eQwOtasUBYro_KKkl4JoNdmVqD-WwCLcBGAsYHQ/s1600/127.PNG)

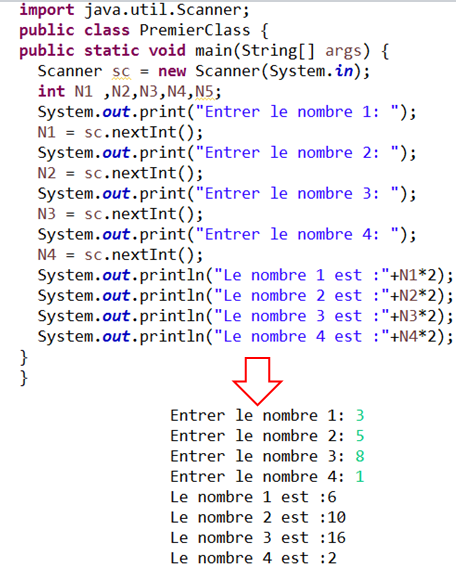
   
 **Exercice 1**  
Ecrire un programme en java qui calcule la somme  S= 1+2+3+…+ 10. Utilisant la boucle  do…..while.

[](https://1.bp.blogspot.com/-uHzdKOlXVyc/XoZgDU7J5GI/AAAAAAAABlQ/xxQrsJKqigsDOxU-Ay2SpK_1RWx3WxJDgCLcBGAsYHQ/s1600/130.PNG)

**Exercice 2**  
Ecrire un programme en java qui affiche la table de multiplication de 5. Utilisant la boucle  do ….While.

[](https://1.bp.blogspot.com/-MA1PYrRjs3g/XoZgeF8iNTI/AAAAAAAABlY/QmqqQcYrC0QTuhhUwF8xyNcGR149rTVjwCLcBGAsYHQ/s1600/133.PNG)

                                           **Chapitre 3 : Les  Tableaux**  
 **1) Introduction**  
Imaginons que dans un algorithme, nous avons besoin d’un grand nombre de variables, il devient difficile de donner un nom pour chaque variable.  
**Exemple :**  
Ecrire un programme  permettant de saisir  quatre notes et de les afficher après avoir multiplié toutes les notes par 2.

[](https://1.bp.blogspot.com/-kHYRI2xexe8/XoZsz88l5VI/AAAAAAAABlk/o57cIIVkQ-s8hBpHHjmUerSaXWqNkOXGwCLcBGAsYHQ/s1600/136.PNG)

**Remarque :**  
  
  
***la suite  sera disponible plus-tard  !!!***