

La boucle while en Java

I - Syntaxe de la boucle while

Le but de la boucle **while** est de répéter un ensemble d'instructions, jusqu'à ce qu'une condition particulière soit vraie. La syntaxe de la boucle **while** est la suivante :

```
while (condition)
{
    instructions
}
```

La *condition* est soit **vraie** soit **fausse**, et sera formulée en utilisant les opérateurs de comparaison vus avec le test if (`==`, `!=`, `<`, `>`, etc.), et les opérateurs logiques [le ET, le OU, le NON].

Le bloc d'instruction, **placé entre accolades** juste après le `while`, sera exécuté **tant que la condition est vraie**.

II - Premier exemple d'utilisation de la boucle while

Imaginons que nous voulions afficher à l'écran tous les nombres entiers compris entre 1 et 20. Une première solution consiste à écrire dans notre programme 20 lignes `System.out.print("5");` dans lesquelles on demanderait d'afficher chaque nombre un par un. Cette solution, pas très ingénieuse, montre ses limites si on voulait afficher les nombres de 1 à 2000 ...

Une autre solution consiste à utiliser une boucle `while`, comme le montre le programme commenté suivant :

```
1  /* début de la classe */
2  class boucle {
3
4      /* début de la fonction main() */
5      public static void main(String[] args) {
6
7          /* Déclaration des variables */
8          int i;
9
10         /* initialisation des variables */
11         i=1;
12
13         /* début de la boucle while : tant que i est inférieur */
14         /* ou égal à 20 on l'affiche et on l'incrémente : */
15         while(i<=20)
16         {
17             System.out.println(i);
18             i=i+1;
19         } /* fin de la boucle while */
20     } /* fin de la fonction main() */
21 } /* fin de la classe */
```

La ligne 14 contenant `i=i+1;` permet d'incrémenter la variable `i`, c'est-à-dire de l'augmenter d'une unité.

III - Un programme complet

Le programme présenté sur la page 2 est la version Java du jeu « **Devinez un nombre** », où l'ordinateur a en mémoire un nombre compris entre 0 et 1000, et où l'utilisateur doit deviner ce nombre. Chaque fois que l'utilisateur propose un nouveau nombre, l'ordinateur répond si le nombre proposé est *plus grand* ou *plus petit* que le nombre secret à trouver. Le but du jeu est de retrouver le nombre secret le plus rapidement possible.

Ce programme Java fait appel aux notions suivantes :

- * Utilisation d'une boucle **while**
- * Utilisation du **if** pour tester les variables
- * Utilisation des **opérateurs de comparaison** [**!=** , **<** et **>**]
- * **Saisie d'une chaîne** de caractères au clavier
- * **Conversion d'une chaîne** de caractères en un nombre réel
- * **Conversion d'un nombre réel** en un nombre entier
- * Initialisation d'une variable avec **un nombre aléatoire** compris entre 0 et 1000

```
1 import java.io.*;
2
3 /*****
4 /* début de la classe */
5 /*****
6
7 class jeux {
8
9     /*****
10    /* début de la fonction main() */
11    /*****
12
13    public static void main(String[] args) {
14
15        /* Déclaration des variables */
16        int nombre_secret,int_nombre_propose;
17        Float reel_nombre_propose;
18        String chaine_nombre_propose;
19        BufferedReader clavier;
20
21        /* initialisation des variables */
22        int_nombre_propose=-1;
23        chaine_nombre_propose=""; /* chaine vide */
24        clavier=new BufferedReader(new InputStreamReader(System.in));
25
26        /* choisi un nombre au hasard entre 0 (compris) et 1000 (nom compris) : */
27        nombre_secret=(int) (Math.random() * 1000);
28
29        /* début de la boucle while */
30        while(int_nombre_propose!=nombre_secret) {
31            System.out.print("Entrez un nombre : ");
32
33            /* saisie une chaine de caractère au clavier */
34            try { chaine_nombre_propose=clavier.readLine(); }
35            catch(IOException e) { }
36
37            /* convertit la chaine chaine_nombre_propose en réel reel_nombre_propose */
38            reel_nombre_propose=Float.valueOf(chaine_nombre_propose);
39
40            /* convertit le réel reel_nombre_propose en entier int_nombre_propose */
41            int_nombre_propose=reel_nombre_propose.intValue();
42
43            /* compare le nombre entier proposé au nombre secret : */
44            if (int_nombre_propose>nombre_secret)
45                System.out.println("Le nombre cherché est plus petit que"
46                +int_nombre_propose);
47
48            if (int_nombre_propose<nombre_secret)
49                System.out.println("Le nombre cherché est plus grand que"
50                +int_nombre_propose);
51
52        } /* fin de la boucle while */
53        System.out.println("Bravo, vous avez trouvé le nombre recherché !");
54    } /* fin de la fonction main() */
55 } /* fin de la classe */
```