

# TP : Boucles

## Remarques :

En TP, lorsqu'un exercice demande d'« écrire une méthode », il faudra aussi la tester en l'incluant dans un programme complet (c'est à dire une méthode `main`) qui l'appelle avec des arguments appropriés.

Ce TP contient un certain nombre d'exercices et pour certains il sera difficile de tous les faire. Il est donc important pour tous d'avoir abordé des exercices des deux sections (Boucles For, Boucle While). N'hésitez pas à passer certains exercices si le temps venait à manquer.

## 1 Rappel Syntaxe

1. En Java, la boucle `pour` a la forme suivante :

```
for (var = b1; var <= b2; var = var +1)
instructions ou bloc d'instructions
```

### Exemple :

```
int i;
for (i=1; i<= n; i= i+1)
```

2. En Java, la boucle `tant que` a la forme suivante :

```
while (condition)
instructions ou bloc d'instructions
```

### Exemple :

```
int n = 0;
while(n * n < K)
n++;
```

## 2 Boucle For

### Exercice 1 : *Affichage*

1. Écrire une méthode `static void ligne(int n)` qui affiche une ligne contenant `n` symboles `*`.
2. Écrire une méthode `static void carre_plein(int n)` qui affiche un carré plein de `n` par `n`.
3. Écrire une méthode `static void carre_vide(int n)` qui affiche un carré vide de `n` par `n`.
4. Écrire une méthode `static int triangle(int n)` qui affiche un triangle plein de hauteur `n` et de base `n` (cf. exemple).

<code>carre_plein(10)</code>	<code>carre_vide(10)</code>	<code>triangle(10)</code>
*****	*****	*
*****	*      *	**
*****	*      *	***
*****	*      *	****
*****	*      *	*****
*****	*      *	*****
*****	*      *	*****
*****	*      *	*****
*****	*      *	*****
*****	*****	*****

### Exercice 2 : Puissance d'un nombre

1. Écrire une méthode `static double puissance (double nb, int n)` qui calcule la puissance  $n^{\text{ème}}$  du nombre `nb`, où `n` est un entier positif. Par exemple, `puissance(2.5, 3)` vaut 15,625.
2. Modifier la méthode précédente pour traiter le cas où `n` est un entier quelconque. Par exemple, `puissance(2.5, -3)` vaut 0,064; `puissance(2.5, 0)` vaut 1.
3. Tester la méthode `puissance` en écrivant une méthode `main`.

### Exercice 3 : Factorielle

On rappelle que la fonction factorielle est définie sur les entiers positifs de la façon suivante :

$$\begin{aligned} \text{factorielle}(0) &= 1 \\ \text{factorielle}(n) &= n \times (n-1) \times \dots \times 2 \times 1 \quad \text{si } n \geq 1 \end{aligned}$$

Écrire une méthode `static int factorielle (int n)` qui calcule la factorielle d'un entier `n`.

### Exercice 4 : Une suite

Écrire une méthode `static int sommeCarresImpairs (int n)` qui calcule la somme des carrés des `n` premiers entiers impairs. Par exemple, `sommeCarresImpairs(5)` =  $1^2 + 3^2 + 5^2 + 7^2 + 9^2 = 165$ .

### Exercice 5 : Deux suites

On définit une suite double :

$$\begin{aligned} u_0 &= 1 & u_{n+1} &= (u_n + v_n)/2 \\ v_0 &= 2 & v_{n+1} &= \sqrt{u_{n+1}v_n} \end{aligned}$$

On admet que les suites  $(u_n)$  et  $(v_n)$  sont adjacentes de limite  $\sqrt{27}/\pi$ .

1. Écrire une méthode `static double approximationDePi (int n)` qui calcule l'approximation du nombre  $\pi$  obtenue à partir de  $v_n$ .
2. Écrire la méthode `main` permettant de tester la méthode précédemment écrite

Rappel : la méthode `Math.sqrt` permet de calculer la racine carrée d'un nombre.

### Exercice 6 : Affichage (Le retour)

Écrire une méthode `static int triangle2(int n)` qui affiche un triangle comme celui dessiné ci-dessous lorsque `n = 5`.

```
      *
     ***
    *****
   ********
  *********
 *****
```

## 3 Boucle While

### Exercice 7 : Deviner un nombre

Un jeu très amusant consiste à choisir un nombre au hasard et le faire deviner à l'adversaire en lui disant « plus grand » ou « plus petit ». Écrire un programme qui devine un nombre entre 1 et 100 choisi par l'utilisateur. Le programme utilisera au moins 2 variables de type `int` : `inf` et `sup` initialisées respectivement à 0 et 100. Elles seront mises à jour en maintenant le fait que la valeur à devinée sera toujours comprise entre `inf` et `sup`.

Exemple de dialogue :

```
> java Devin  
Est-ce 50 ? plus  
Est-ce 75 ? plus  
Est-ce 87 ? moins  
Est-ce 81 ? plus  
Est-ce 84 ? moins  
Est-ce 82 ? oui
```