

## Laboratoire 6 - Les méthodes - introduction

### Première partie

1. Écrivez une méthode qui affiche un menu à l'écran. Voici ce menu :

1 : Ajouter  
2 : Soustraire  
3 : Multiplier  
4 : Diviser  
0 : Ce menu

Autrement dit, votre méthode, qui s'appellera `afficherMenu` devra afficher les 5 lignes précédentes à l'écran.

Ensuite, testez votre méthode en l'appelant de votre méthode `main`.

2. Utilisez le code que vous avez écrit la semaine précédente. Vous deviez écrire un petit programme java qui saisit un nombre entier positif ( $> 0$ ) (disons  $n$ ) et qui calcule le  $n^{\text{ième}}$  nombre harmonique  $H_n$  défini comme suit :

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

Voici quelques valeurs

$n$	$H_n$
1	1
2	1.5
3	1.833333333333...
4	2.083333333333...
5	2.283333333333...
6	2.45
7	2.59285714...

En utilisant la structure de boucle de votre choix (`while`, `do ... while`, `for`), faites une méthode appelée `harmonique`, qui prend en paramètre un entier (disons  $n$ ) et qui calculera puis affichera la valeur du  $n^{\text{ième}}$  nombre harmonique  $H_n$ . Si la valeur de  $n$  est inférieure ou égale à 0, la méthode devra afficher le message "valeur de n non valide". La méthode ne retourne pas de valeur (`void`).

Ensuite, testez votre méthode `harmonique` dans la méthode `main`. Votre classe contiendra donc deux méthodes soit `harmonique` et `main`.

3. Écrivez un petit programme dans lequel il y aura la méthode `harmonique` et qui affiche une table des dix premiers nombres harmoniques.

## Deuxième partie (supplément facultatif)

Nous avons fait en classe les méthodes suivantes :

```
public static void afficherNfois ( int nbFois, char carac ) { ...  
public static void afficherRectangle ( int largeur, int hauteur ) { ...
```

En vous inspirant des méthodes faites au cours,

1. Écrire la méthode `carre` qui affiche un carré de côté `cote`. Puisqu'un carré est un cas particulier de rectangle, votre méthode devra utiliser la méthode `afficherRectangle`.

```
public static void afficherCarre ( int cote ) { ...
```

2. Écrire la méthode `afficherTriangleIsoGauche` qui prend en paramètre la hauteur et qui affiche, toujours avec le caractère `*`, un triangle droit de hauteur et de base hauteur. Par exemple, l'appel `afficherTriangleIsoGauche ( 5 )` affichera

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

3. Écrire la méthode `afficherTriangleIsoDroit` qui prend en paramètre la hauteur et qui affiche, toujours avec le caractère `*`, un triangle droit de hauteur et de base hauteur. Par exemple, l'appel `afficherTriangleIsoDroit ( 5 )` affichera

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

4. Écrire la méthode `afficherTriangleEqui` qui prend en paramètre la hauteur et qui affiche un triangle équilatéral de hauteur hauteur. Par exemple, l'appel `afficherTriangleEqui ( 5 )` affichera

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