

# TP 12 Programmes Java

D'abord, démarrer un terminal

Taper les commandes suivantes pour commencer. Cela pourrait vous aider à vous mieux organiser votre travail ; vous pourriez écrire tous vos fichiers Java dans un dossier tp12.

```
cd                #se placer dans le dossier personnel
mkdir tp12        #créer un nouveau dossier
cd tp12           #se placer dans le nouveau dossier tp12
touch Exo1.java   #créer nouveau fichier
kate Exo1.java&  #editer le fichier, n'oublier pas le '&' sinon le terminal reste bloqué
```

**Exercice 1** Écrire dans le fichier `Exo1.java` un programme Java qui affiche *Ciao Bella*. Il suffit d'ajouter une ligne au code ci-dessous.

```
class Exo1{
    public static void main(String [] args){
        ....
    }
}
```

Pour compiler et exécuter, il faut utiliser `javac` et `java` ainsi :

```
javac Exo1.java
java Exo1
```

**Exercice 2** Modifier le programme pour écrire une classe `Exo2` dans un fichier `Exo2.java`. Ajouter une fonction statique `puissance5(float x)` qui renvoie  $x^5 = x \times x \times x \times x \times x$ . Appeler cette fonction dans le programme principal (c. à. d. dans la fonction `main(...)`) pour afficher  $2^5$ ,  $3^5$ ,  $4^5$  et  $5^5$ . Note :  $5^5 = 3125$ .

**Exercice 3** Écrire une classe `Exo3` dans un fichier `Exo3.java`. Définir une fonction statique

```
static double impots(double revenuFiscalRef)
```

qui renvoie le montant des impôts pour un revenu fiscal référence `revenuFiscalRef`. Faire fonctionner le code ci-après, mais n'hésitez pas à tester le programme pour plusieurs valeurs à la ligne 6. Il s'agit d'un impôt progressif, par tranches, basé sur les règles (simplifiées) suivantes définies par la loi :

- Pas d'impôt pour la tranche 0 – 9710 euros.
- 14% pour la tranche de 9710 à 26818 euros
- 30% pour tout revenu supérieur à 26818 euros.

Attention : l'impôt est progressif en France. Tester votre programme sur les revenus suivants :

- `impots(10000) = 0.14 × (10000 – 9710) = 40.6;`
- `impots(30000) = 0.14 × (26818 – 9710) + 0.3 × (30000 – 26818) = 3349.72.`

```
1 class Exo3{
2     static double impots(double ...) {
3         ...
4     }
5     public static void main(String [] args){
6         double revenuNet = 10000; //valeur écrite en dur dans le code
7         double mesImpots = impots(revenuNet);
8         System.out.println(" Vos_impôts="+mesImpots);
9     }
10 }
```

---

**Exercice 4** Écrire une classe `Exo4` dans un fichier `Exo4.java` pour afficher l'indice de masse corporelle (le poids divisé par la taille en mètres au carré) pour un poids entier et une taille entière écrits en dur dans le code. Faire fonctionner le code ci-après.

```
class Exo4{
    static double imc(int kg, int cm){
        double m = ...
        ... ..
    }
    public ... void main(String [] args){
        int kg=70;
        int cm=170;
        double indice=imc(kg,cm);
        System.out.println("IMC="+indice);
        if(indice < 20)
            System.out.println("maigreur");
        if(indice > 30)
            System.out.println("surpoids");
    }
}
```

**Exercice 5** Écrire une classe `Exo5` dans un fichier `Exo5.java`. Dans ce programme `Exo5` on écrit le salaire en dur dans le code, ex, `int salaire=100000`; Si le salaire est supérieur ou égal à 1000000, il faut afficher « vous avez besoin d'un parachute doré ». Si le salaire est inférieur à ou égal à 10000 euros, il faut afficher « vous ne payez pas d'impôt; sympa ». Pour tout salaire entre 10001 euros et 999999 euros, il faut afficher « je pense que vous allez payer des impôts ».

Il faut écrire que la fonction `main(...)`

**Exercice 6** Écrire une classe `Exo6` dans un fichier `Exo6.java`. Ajouter une fonction qui renvoie la valeur minimale d'un tableau grâce à une boucle `for`. Remplir le code ci-après :

```
class Exo6{
    static int minTab(int [] tab){
        int m = tab[0];
        ... ..
    }
    public ... void main(String [] args){
        int [] z = {1, 2, 3, 4, 0};
        System.out.print("Taille de tab:");
        System.out.println(z.length);
    }
}
```

**Exercice 7** Écrire une classe `Exo7` dans un fichier `Exo7.java`. Ajouter une fonction qui renvoie le produit des valeurs d'un tableau passé comme argument. Il faut utiliser une boucle `for`. Tester le programme.

**Exercice 8** Écrire une classe `Exo8` dans un fichier `Exo8.java`. Ajouter une fonction qui renvoie le nombre de valeur négatives dans un tableau `tab`. Il faut utiliser une boucle `for`. Tester le programme.

**Exercice 9** Écrire une classe `Exo9` dans un fichier `Exo9.java`. Ajouter une fonction qui renvoie le nombre d'apparitions d'une valeur `val` dans un tableau `tab`. Il faut utiliser une boucle `for`. Tester le programme.