

# TP 13 Programmes Java

D'abord, démarrer un terminal en utilisant une des méthodes indiquées sur le site ci-dessous, comme au TPs précédents. Le mieux est peut-être d'utiliser la machine virtuelle Antix.

`cedric.cnam.fr/~porumbed/var11/term/`

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 `tp13`.

```
cd                #se placer dans le dossier personnel
mkdir tp13        #créer un nouveau dossier
cd tp13           #se placer dans le nouveau dossier tp13
touch Exo1.java   #créer nouveau fichier
geany Exo1.java& #éditer le fichier, n'oublier pas le '&' sinon le terminal reste bloqué
```

**Exercice 1** Utiliser le code à droite pour écrire un fichier `Exo1.java` qui permet d'afficher le nombre de valeurs positives dans un tableau. Il faut écrire que le `main`. Pour compiler et exécuter, il faut utiliser `javac` et `java`, voir les indications fournies dans les diapos du cours 12 ou les TP précédents.

Modifier le programme pour afficher en plus le nombre de valeurs inférieures ou égales à zéro.

```
class E...{
    public ... void main (String []...){
        int [] tab = {3, 9, -23,1,29,-134}
        int nbPos = ....
        for (int i=0;i<tab.length;i++){
            if( tab[i] ....){
                nbPos = nbPos + ....
            }
        }
        System.out ....
    }
}
```

**Exercice 2** Écrire une classe `Exo2` avec une méthode statique `conclureNote(int note)` qui ne renvoie rien et qui affiche :

- échec si la note est inférieure à 10;
- succès si la note appartient à l'intervalle  $[10, 19]$
- vous êtes en génie pour une note de 20

**Exercice 3** Écrire une classe `Exo3` avec une fonction `valeurAbsolue(double x)` qui renvoie la valeur absolue de `x`. Si `x >= 0`, il faut renvoyer `x`; sinon, il faut renvoyer `-x`.

**Exercice 4** Écrire une classe `Exo4` avec une fonction `carreParfait(int x)` qui renvoie `true` si `x` est un carré parfait (ex,  $64 = 8^2$ ,  $10000 = 100^2$ ) ou `false` sinon. **Indication** vous pouvez faire une boucle `for` pour parcourir les valeurs  $i = 1, 2, 3, \dots, x$ . S'il y a un seul  $i$  tel que  $i^2 = x$ , alors `x` est un carré parfait et on

renvoie `true`. Si la boucle finit sans renvoyer `true`, on renvoie `false` à la fin du programme.

**Exercice 5** Écrire une classe `Exo5` avec une fonction `sommeImpairs(int n)` qui renvoie la somme des premiers  $n$  impairs. Par exemple, pour  $n = 5$  il faut renvoyer  $1 + 3 + 5 + 7 + 9 = 25$ . N'hésitez pas à utiliser le bloc de code ci-après comme exemple.

```
int somme = 0;
for (int i=1;i<n;i=i+2)//attention: i=i+2
    somme = somme + i
```

Une propriété sympa : une telle somme est toujours un carré parfait. Écrire dans le `main` le code qui permet de vérifier que la valeur renvoyée par `sommeImpairs(int n)` est bien en carré parfait pour tout  $n$ ; vous pouvez utiliser la fonction `carreParfait(...)` écrite à l'exercice précédent.

**Exercice 6** Écrire un programme `Exo6.java` avec

une fonction `void compter(int[] t)` qui calcule le nombre de valeurs négatives et le nombre de valeurs positives dans un tableau `t`. La fonction doit stocker ces nombres dans les variables globales (attention statiques!) `nbPosit` et `nbNegat`. Remplir le code ci-dessous et tester votre fonction sur le tableau `tab`.

```
class Exo5{
    static int nbPosit=0;
    static int nbNegat=0;
    public static void compter (...) {
        ...
    }
    public static void main( ... ) {
        int [] tab = {-3, 10, -8, -1, 9, -1, 2, 7, -43};
        ...
    }
}
```

**Exercice 7** Modifier le programme ci-après pour le faire afficher le taux moyen d'imposition pour un revenu saisi par l'utilisateur. Le taux moyen d'imposition est la part que l'impôt représente par rapport au revenu fiscal de référence. C'est le résultat du rapport entre l'impôt et le revenu. Pour un revenu de 20000 euros, il faut payer un impôt de  $(20000-9710)*0.14=1440.6$ . Le taux moyen d'impôt. est  $1440.6/20000=0.72=7.2\%$ .

```
import java.util.*;
class ....{
    static double calculerImpots (....) {
        //coller le code de l'exercice 3 du TP précédent
    }
    public ... void main (String []...) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Entrez votre revenu fiscal de référence:");
        double revenu = scan.nextDouble();
        //à remplir
    }
}
```

**Exercice 8** Écrire une fonction d'en-tête

```
int nbPersonnesImposables(double[] revenus)
```

qui reçoit en entrée un tableau (de) `revenus` de plusieurs personnes et qui renvoie le nombre de personnes imposables. Une personne est imposable si son impôt dépasse 0, vous pouvez utiliser une fonction `calculerImpots()` qu'on a déjà vue plusieurs fois (voir exo précédent). Le programme principal devrait demander à l'utilisateur de saisir 5 valeurs de revenu avec un code comme :

```
double [] revenus = new double [5];
Scanner scan = new Scanner(System.in);
System.out.println("Entrez les revenus de 5 personnes:");
for (int i=0; i<5; i++)
    revenus [i] = scan.nextDouble();
```