

## TP 2 : Un peu de syntaxe java

Veillez consulter les vidéos sur ma chaine YouTube :

[https://www.youtube.com/watch?v=iW8M3nFb-es&list=PLQ4KaU9Gz0\\_G3jJTsq6ddAU0Bp8-gvSzA](https://www.youtube.com/watch?v=iW8M3nFb-es&list=PLQ4KaU9Gz0_G3jJTsq6ddAU0Bp8-gvSzA)

### Objectifs :

Le but de ce TP est d'acquérir les notions fondamentales relatives au langage java à savoir, les types de données, les structures conditionnelles ainsi que les structures répétitives.

### Consignes :

Lire le manuel NetBeans mis sur la plateforme. Les activités à faire avec le responsable du TP. L'étudiant doit préparer sa série avant d'assister à la séance de TP. Les étudiants ayant achevé les activités avant les délais peuvent reprendre sur machine tous les exercices résolus de la série de TD.

Les exercices utilisant des boucles, doivent être repris avec tous les types de boucles et vérifiés.

**Durée de réalisation** : 2 Heures

### Activité 1

Ecrire un programme en langage java qui affiche les informations fournies par les méthodes suivantes (N est saisi au clavier) :

- Une méthode qui calcule la somme des N premiers entiers pairs (exemple : pour N= 5, on aura  $2+4+6+8+10=30$ )
- Une méthode qui calcule qui affiche les dix nombres suivants le nombre N. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

### Activité 2

Ecrire un programme en langage java qui calcule la somme des éléments impairs d'un tableau de 20 entiers (introduits en donnée).

### Activité 3

- Compiler le fichier **ProgObjet1.java** puis le faire exécuter.
- Noter le résultat de son exécution :

```
class ObjetTableau { // début de la classe
static int nbobjets = 0;
int n ;
int[] t ;
ObjetTableau(int a) { // le constructeur
n = a; // n prend la valeur a
t = new int[n] ; // on réserve la place pour le tableau t
for (int i=0;i<n;i++) { // remplissage du tableau
t[i] = i+1 ;
}
nbobjets++;
```

```

} // fin du constructeur
void affiche() {
System.out.println("affichage d'un objet");
System.out.println("n vaut "+n);
for (int i = 0; i < n; i++){
System.out.print(t[i]+" * ");
}
System.out.println("");
}
static void nombreObjets() {
System.out.println("Nombre d'objets : " + nbobjets);
}
} // fin de la classe
public class ProgObjet1 { // début de classe
public static void main (String args[ ]) { // le main
ObjetTableau.nombreObjets();
ObjetTableau obj1 = new ObjetTableau(6);
obj1.affiche();
ObjetTableau.nombreObjets();
ObjetTableau obj2 = new ObjetTableau(3);
obj2.affiche();
ObjetTableau.nombreObjets();
} // fin du main
} // fin de la classe

```

Noter dans le tableau suivant si la méthode appelée est une méthode de classe ou d'instance, et préciser quel est l'affichage correspondant :

instruction	méthode : classe / instance	affichage
ObjetTableau.nombreObjets();		
ObjetTableau obj1 = new ObjetTableau(6);		
obj1.affiche();		
ObjetTableau.nombreObjets();		
ObjetTableau obj2 = new ObjetTableau(3);		
obj2.affiche();		
ObjetTableau.nombreObjets();		

#### **Activité 4**

Soit le programme suivant :

```

public class Birthday {
private Date date;
private String name;
public Birthday(Date date, String name) {
this.date = date;
this.name = name;
}
}

```

```
}  
public String toString() {  
return "[ Anniversaire de " + name + " le " + date + " ]";  
}  
public static void main(String arg[]) {  
Date d = new Date(29, 6, 2009);  
Birthday e = new Birthday(d, "Aridj");  
System.out.println(e);  
d.setMonth(6);  
System.out.println(e);  
}  
}
```

Qu'affiche ce programme ? Respecte-t-il le principe d'encapsulation des données ? Si non, comment le corriger ? Ajouter une méthode pour comparer l'âge de deux personnes.

### **Activité 5**

Ecrire un programme qui saisit une année et qui vérifie si celle-ci est bissextile.