

## Complément TP

### Remise à niveau Java

Le but de ce TP est de continuer à rappeler très rapidement les principes de base du langage Java. Pour ceux qui débutent réellement en Java, il est évident que ces quelques exercices ne suffiront pas. Aussi, il vous est conseillé de travailler par vous même pour vous familiariser avec la programmation en Java, qui est un langage orienté objet dérivé du C, mais plus simple que le C++.

1) Coller les deux morceaux de code suivants dans deux fichiers, compilez-les et corrigez-les.

```
public class TestAdd {
    static final int NOMBRE = 10;
    public static void main(String args[]) {
        Add f = new Add();
        int i = 0;
        for (int i = 0; i < NOMBRE; i++) {
            System.out.print(i + " ");
            ajouter(f, i);
            afficher(f);
            System.out.print("\n");
        }
    }
}

public class Add {
    private int valeur = 0;
    public void ajouter(int valeur) {
        valeur += valeur;
    }
    public void afficher() {
        System.out.println("Valeur = " + valeur);
    }
}
```

2) Coller le code ci-dessous dans un fichier, puis compilez-le

```
public class TestArray {
    public static void main(String args[]) {
        String jourSemaine[];
        jourSemaine = new String[7];
        jourSemaine[0] = "dimanche";
        jourSemaine[1] = "lundi";
        jourSemaine[2] = "mardi";
        jourSemaine[3] = "mercredi";
        jourSemaine[4] = "jeudi";
        jourSemaine[5] = "vendredi";
        jourSemaine[6] = "samedi";
        System.out.println("Moi, je prefere le " + jourSemaine[3]);
    }
}
```

Modifiez le code pour que le jour de la semaine soit dépendant du premier argument de la ligne de commande `args[0]`. Pour cela, utilisez la classe *Integer* avec l'idée de faire passer une chaîne de caractères vers une valeur entière.

## 3) Voici deux prototypes de classes :

Classes	<i>Point</i>	<i>Rectangle</i>
Données membres	<i>double x ;</i> <i>double y ;</i>	<i>Point[] points ;</i>
Méthodes membres	<i>Point (double, double) ;</i> <i>void setXY (double, double) ;</i> <i>double getX() ;</i> <i>double getY() ;</i> <i>double distance(Point) ;</i>	<i>Rectangle(Point, Point, Point, Point) ;</i> <i>double grandCote() ;</i> <i>double petitCote() ;</i> <i>double surface() ;</i> <i>double perimetre() ;</i> <i>void deplacer(double, double) ;</i> <i>void afficher() ;</i>

Ecrivez les classes Java correspondantes (un fichier par classe), l'ordre des points n'est pas précisé et la garantie que les quatre points forment un rectangle n'est pas à vérifier dans cette première étape.

Voici le code de la méthode `afficher()` de la classe `Rectangle` :

```
public void afficher(String s){
    System.out.println("Surface de " + s + " = " + surface());
    System.out.println("Perimetre de " + s + " = " + perimetre());
    System.out.println("Petit cote de " + s + " = " + petitCote());
    System.out.println("Grand cote de " + s + " = " + grandCote());
}
```

Tester le résultat obtenu avec le programme suivant :

```
public class TestUML {
    public static void main(String args[]) {
        Point p1 = new Point(1,1);
        Point p2 = new Point(3,1);
        Point p3 = new Point(3,5);
        Point p4 = new Point(1,5);
        Rectangle r1 = new Rectangle(p1, p2, p3, p4);
        System.out.println("Rectangle r1 : ");
        r1.afficher();
        p3.setXY(3,10);
        p4.setXY(1,10);
        System.out.println("Rectangle r1 : ");
        r1.afficher();
        System.out.println("J'aimerais bien que r1 n ait pas changé");
        Rectangle r2 = new Rectangle(p1, p2, p3, p4);
        System.out.println("Rectangle r2 : ");
        r2.afficher();

        r2.deplacer(3,1);
        System.out.println("Rectangle r1 : ");
        r1.afficher();
        System.out.println("Rectangle r2 : ");
        r2.afficher();
    }
}
```