



## TP N° 3

### Exercice 1

Soient les classes suivantes :

<pre>class A {     int a;     A(int a) { a=a; } }</pre>	<pre>class B extends A {     int b;     B(int b) { b=b; } }</pre>	<pre>class C extends B {     int c; }</pre>
---	---	---

Corrigez les erreurs **sur les constructeurs** des classes A, B et C.

### Exercice 2

Ecrire en java chacune des classes décrites ci-dessous :

1. Une **Personne** a un nom (String) et un nombre d'enfants (int).
2. Un **Salarié** est une personne qui perçoit un salaire (double). Tout salarié, a droit à une allocation familiale "AF" (int) de 300 DA pour chacun de ses enfants. "primeAF ()" permet de calculer le total des AFs que perçoit le salarié. "afficher()" permet d'afficher son nom et sa prime.
3. Le programme **TestAF** permet de créer un salarié et d'afficher son nom et sa prime.

### Exercice 3

Soit la classe **Point** ainsi définie :

```
class Point {
    private double x ;
    private double y ;
    public Point (double abs, double ord) { x = abs ; y = ord ; }
    public void affiche () { System.out.print (" (" + x + "," + y + ")") ; }
    public boolean identique (Point a) { return ( (a.x==x) && (a.y==y) ) ; }
}
```

- 
- 1- Réaliser une classe **PointNom**, dérivée de Point
    - a. Définir un attribut nom (char) permettant de manipuler des points avec leurs noms
    - b. Ecrire un constructeur PointNom(double x, double y, char nom)
    - c. Redéfinir (compléter) la méthode affiche() afin qu'elle affiche en plus des coordonnées le nom du point, par exemple : M(5,2)
    - d. Dans la classe PointNom, redéfinir la méthode identique fournissant la valeur true lorsque les deux points concernés ont à la fois mêmes coordonnées et même nom

2- Soit la classe **TestPoint**. Complétez et répondez aux commentaires du code suivant :

```
public class TestPoint{
    public static void main (String args[]){
        Point p1, p2;
        p2= new PointNom(2,5,'A') ;
        // Affichez seulement les coordonnes de p2 "(2, 5)", sans le nom.

        p1=p2;
        System.out.println(p2.identique(p1));        // expliquez l'affichage

        //changez le nom de p1 par 'B'

        System.out.println(p2.identique(p1));        // expliquez l'affichage

        p1 = new PointNom(2,5,'A');
        System.out.println(p2.identique(p1)+ " "+p1+" "+p2);
        // Expliquez l'affichage.
        //Que faut-il faire pour que « p2. identique(p1) » retourne true ?
    }
}
```