

Programmation Orientée Objet Avancée
– TP 1 –

Exercice 1 :

On dispose de la classe suivante :

```
public class Point {  
    public Point (double x, double y) { this.x=x ; this.y=y ; }  
    public void deplace (double dx, double dy) { x+=dx ; y+=dy ; }  
    public void affiche () {  
        System.out.println ("Point de coordonnees " + x + " " + y) ;  
    }  
    public double getX() { return x ; }  
    public double getY() { return y ; }  
    private double x, y ;  
}
```

On souhaite réaliser une classe Cercle disposant des méthodes suivantes :

- constructeur recevant en argument les coordonnées du centre du cercle et son rayon,
- deplaceCentre pour modifier les coordonnées du centre du cercle,
- changeRayon pour modifier le rayon du cercle,
- getCentre qui fournit en résultat un objet de type Point correspondant au centre du cercle,
- affiche qui affiche les coordonnées du centre du cercle et son rayon.

- Définir la classe Cercle comme possédant un membre de type Point.
- Définir la classe Cercle2 comme classe dérivée de Point.

Dans les deux cas, on écrira un petit programme mettant en jeu les différentes fonctionnalités de la classe Cercle.

Exercice 2 :

- Réaliser une classe Repertoire permettant de gérer un répertoire téléphonique associant un numéro de téléphone (chaîne de caractères) à un nom. Pour faciliter les choses, on prévoira une classe Abonne destinée à représenter un abonné et disposant des fonctionnalités indispensables.

La classe Repertoire devra disposer des fonctionnalités suivantes :

- constructeur recevant un argument de type entier précisant le nombre maximum d'abonnés que pourra contenir le répertoire (cette particularité évite d'avoir à se soucier d'une gestion dynamique du répertoire),
 - méthode addAbonne permettant d'ajouter un nouvel abonné ; elle renverra la valeur false si le répertoire est plein, la valeur true sinon,
 - méthode getNumero fournissant le numéro associé à un nom d'abonné fourni en argument,
 - méthode getNAbonnes qui fournit le nombre d'abonnés figurant dans le répertoire,
 - méthode getAbonne fournissant l'abonné dont le rang est fourni en argument,
 - méthode getAbonnesTries fournissant un tableau des références des différents abonnés, rangés par ordre alphabétique (pour simplifier, on supposera que les noms sont écrits en minuscules, sans caractères accentués).
- Écrire un petit programme de test.