	L'Institut International de Technologie Sfax (IIT) **Année Universitaire :** 2020-2021 **Module :** POO **Responsable du cours :** Fairouz Fakhfakh **Série N° : 5**

Exercice n°1

Un point est caractérisé par ses deux coordonnées cartésiennes (de type double) **x** et **y.**
Réaliser une classe **Point** disposant des fonctionnalités suivantes :

- un constructeur qui initialise les deux coordonnées,
- une méthode **getX** retournant l'abscisse de l'objet courant
- une méthode **getY** retournant l'ordonnée de l'objet courant
- une méthode **afficher** qui affiche les coordonnées d'un point
- une méthode **deplacer** permettant d'effectuer une translation d'un point.

On souhaite réaliser une classe Cercle, comme classe dérivée de Point, 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.

Ecrire un programme mettant en jeu les différentes fonctionnalités de la classe Cercle.

Exercice n°2

Ecrire les classes nécessaires au fonctionnement du programme suivant (en ne fournissant que les méthodes nécessaires à ce fonctionnement) :

```
public class TestFormes {
        public static void main(String[] argv) {
                Forme[ ] figures = new Forme[3] ;
                figures [ 0 ] = new Carre( 2 ) ; // Création d'un carré de 2 cm de coté
                figures [ 1 ] = new Cercle( 3 ) ; // Création d'un cercle de 3 cm de rayon
                figures [ 2 ] = new Carre( 5.2 ) ; // Création d'un carré de 5,2 cm de coté
                for( int i=0 ; i< figures.length ; i++ )
                        System.out.println( figures[i].toString() +" : surface = "+
                figures[i].getSurface() +"cm2" ) ;
                }
                }
```

Sortie du programme :

Carré (coté 2.0 cm) : surface = 4.0cm2
Cercle (rayon 3.0 cm) : surface = 28.274333882308138cm2
Carré (coté 5.2 cm) : surface = 27.040000000000003cm2

Exercice n°3

Ecrire une classe **Animal** qui dispose d'un attribut entier ***nbPattes***. Cette classe dispose des méthodes suivantes :

- un constructeur, qui prend en argument un entier (le nombre de pattes).
- **String toString(),** qui renvoie une chaîne de caractères contenant le nombre de pattes de l'animal.
- **Affiche()** qui affiche le nombre de pattes de l'animal (appeler la méthode toString()).

1. Ecrire une classe **Poule** qui hérite de la classe Animal
2. Ecrire une classe **Lapin** qui hérite de la classe Animal
3. Ecrire une classe **ProgAnimal** dans laquelle la méthode **main()** crée un Lapin et une Poule.
4. Modifier la classe **ProgAnimal** de façon à conserver la liste des animaux se trouvant dans un zoo et à afficher à la fin le nombre d'animaux ayant un nombre de pattes donné. La nouvelle version de la classe doit utiliser une nouvelle classe **zoo** disposant des méthodes suivantes :
    - un constructeur qui prend en argument la taille du zoo (nombre maximal d'animaux)
    - **Ajouter** qui ajoute un animal dans le zoo
    - **Affiche_nombre** qui affiche le nombre d'animaux ayant un nombre de pattes donné.

Exercice n°4

Soient les trois classes suivantes :

```
class Ellipse {
    double r1, r2;
    Ellipse (double r1, double r2) {this.r1=r1 ; this.r2=r2;  }
    double aire() {.............}
}

class Cercle extends Ellipse {
    Cercle(double r)   {super (r,r);  }
    double getRayon()  {return r1;   }
}
```

```
class Test {
        public static void main(String[] args) {

                Ellipse e = new Ellipse (2,4);
                Cercle c = new Cercle (2);
                System.out.println(e instanceof Cercle); // inst 1
                System.out.println(e instanceof Ellipse); // inst 2
                System.out.println(c instanceof Cercle); // inst 3
                System.out.println(e instanceof Ellipse); // inst 4
                e=c ;  // inst 5
                System.out.println(e instanceof Cercle );  // inst 6
                System.out.println(e instanceof Ellipse); // inst 7
                double r= e.getRayon();// inst 8
                c= e; // inst 9
                double r= ((Cercle) e).getRayon();  // inst 10
                c = (Cercle)e ;   // inst 11
                }
}
```

Donner le résultat de chaque instruction du programme principal en remplissant ce tableau :

Instructions	**Résultats**	**Une erreur est générée**	
		oui	non
inst 1			
inst 2			
inst 3			
inst 4			
inst 5	-		
inst 6			
inst 7			
inst 8			
inst 9			
inst 10			
Inst 11			