TD1
Programmation Orientée Objet

Exercice 1

Soit la classe *Personne* suivante :

```
Public class Personne
{
	private String nom , prenom ;

	public Personne( String n , String p)
	{
		nom =n;
		prenom = p ;
	}

	Public void afficher ()
	{
		System .out. print ("Je suis : " + nom + " " + prenom );
	}
}
```

Ecrire une classe application permettant **d'instancier** une personne, **afficher** la personne.

Exercice 2

1. Réaliser une classe Point permettant de représenter un point sur un axe. Le point est caractérisé par un nom (de type char) et une abscisse (de type double). On prévoit :
   - un constructeur recevant en arguments le nom et l'abscisse du point,
   - une méthode affiche imprimant le nom du point et son abscisse,
   - une méthode translate effectuant une translation définie par la valeur de son argument.

2. Écrire le programme principal utilisant cette classe pour créer un point, en afficher les caractéristiques, le déplacer et en afficher à nouveau les caractéristiques.

Exercices à domicile (semaines 3 & 4)

Exercice 3 (Méthodes d'accès aux champs privés)

Soit le programme suivant comportant la définition d'une classe nommée Point1 et son utilisation :

```
class Point1
{
	private double x ; // abscisse
	private double y ; // ordonnee

	public Point1(int abs, int ord)
	{
		x = abs ;
		y = ord ;
	}

	public void deplace (int dx, int dy)
	{
		x += dx ;
		y += dy ;
	}

	public void affiche ()
	{
		System.out.println ("Je suis un point de coordonnees " + x + " " + y) ;
	}

}
public class Exo3
{
	public static void main (String[] args)
	{
		Point1 a ;
		a = new Point1(3,5) ;
		a.affiche() ;
		a.deplace(2, 0) ;
		a.affiche() ;
		Point1 b = new Point1(6, 8) ;
		b.affiche() ;
	}
}
```

- Modifier la définition de la classe Point1 en supprimant la méthode affiche et en introduisant deux méthodes d'accès nommées abscisse et ordonnee fournissant respectivement l'abscisse et l'ordonnée d'un point.
- Adapter la méthode main en conséquence.

Exercice 4 (Conversions d'arguments)

On suppose qu'on dispose de la classe A ainsi définie :

```
class A
{
	void f (int n, float x) { ..... }
	void g (byte b) { ..... }
	.....

}
```

Soit ces déclarations :

```
A a ; int n ; byte b ; float x ; double y ;
```

Dire si les appels suivants sont corrects et sinon pourquoi.

```
a.f (n, x) ;
a.f (b+3, x) ;
a.f (b, x) ;
a.f (n, y) ;
a.f (n, (float)y) ;
a.f (n, 2*x) ;
a.f (n+5, x+0.5) ;
a.g (b) ;
a.g (b+1) ;
a.g (b++) ;
a.g (3) ;
```

Exercice 5 (Surdéfinition de méthodes)

Quelles erreurs figurent dans la définition de classe suivante ?

```
class Surdef
{
        public void f (int n) { ..... }
        public int f (int p) { ..... }
        public void g (float x) { ..... }
        public void g (final double y) { ..... }
        public void h (long n) { ..... }
        public int h (final long p) { ..... }
}
```

Exercice 6 (Recherche d'une méthode surdéfinie)

Soit la définition de classe suivante :

```
class A
{
        public void f (int n) { ..... }
        public void f (int n, int q) { ..... }
        public void f (int n, double y) { ..... }
}
```

Avec ces déclarations :

```
A a ; byte b ; short p ; int n ; long q ; float x ; double y ;
```

Quelles sont les instructions correctes et, dans ce cas, quelles sont les méthodes appelées et les éventuelles conversions mises en jeu ?

```
a.f(n);
a.f(n, q) ;
a.f(q) ;
a.f(p, n) ;
a.f(b, x) ;
a.f(q, x) ;
```