

TD n°2

Objets

Exercice 1 Qu'affiche le programme suivant ?

```
public class Ex1 {
    private int a;
    public Ex1() {
        this.a = 0;
    }
    public void f() {
        this.a++ ;
    }
    public void affiche() {
        System.out.println(this.a);
    }
}

public class Test {
    public static void main(String [] args) {
        Ex1 p = new Ex1();
        Ex1 q = new Ex1();
        p.affiche();
        q.affiche();
        p.f();
        p.affiche();
        q.affiche();
        p = q;
        p.f();
        p.affiche();
        q.affiche();
    }
}
```

Exercice 2 Qu'affiche le programme suivant ?

```
public class Ex2 {
    private int a;
    public Ex2() {
        this.a = 0;
    }
    public void f(int a) {
```

```

        System.out.println(a) ;
    }
}

public class Test {
    public static void main(String[] args) {
        Ex2 p = new Ex2();
        p.f(12);
    }
}

```

Dans la fonction `f`, comment faire si on veut également afficher la valeur de la propriété `a` ?

Exercice 3 Ecrire une classe implémentant un *couple* d'entiers :

1. Définir une classe `Couple` dont le constructeur initialise les attributs privés du couple. Définir une méthode `affiche` et une fonction `main` pour tester cette classe.
2. Définir un deuxième constructeur, qui initialisera à 0 les composants du couple.
3. Définir un troisième constructeur, qui initialisera un couple à l'aide d'un autre couple.
4. Définir des fonctions permettant d'accéder et de modifier chaque élément du couple.

5. Que dire du code suivant

```

Couple p,q;
p = new Couple(1,2);
q = new Couple(1,2);
if (p == q) System.out.println("pile");
else System.out.println("face");

```

Définissez une méthode qui permette de nuancer davantage les comparaisons entre ces objets.

6. Enrichir la classe `Couple` d'une méthode définissant quand un couple est inférieur à un autre selon la règle lexicographique suivante :

$$(x_1, y_1) < (x_2, y_2) \quad \text{ssi} \quad (x_1 < x_2) \text{ ou } (x_1 = x_2 \text{ et } y_1 < y_2).$$

7. Définissez une classe `DeuxCouples` qui soit composée de deux couples, ainsi qu'un constructeur naturel.
8. Définissez un ordre lexicographique sur les `DeuxCouples` et une méthode de comparaison.

Exercice 4 Qu'affiche le programme suivant ?

```

public class Ex4 {
    private int a;
    public Ex4 () {
        this.a = 0;
    }
    public static void f(Ex4 o) {
        o.a++;
        o.affiche();
    }
}

```

```

    }
    public void affiche() {
        System.out.println(a);
    }
}

public class Test {
    public static void main(String[] args) {
        Ex4 obj = new Ex4();
        obj.affiche();
        Ex4.f(obj) ;
        obj.affiche();
    }
}

```

Exercice 5 *Devinettes*

On considère la classe suivante :

```

public class Integer {
    public static int n;
    private int m;

    public Integer(int i) {
        n = i;
        this.m = i;
    }

    public void afficher() {
        System.out.println("n = " + n + ", m = " + this.m);
    }
}

```

1. Ecrire les accesseurs/modifieurs pour `m`. Quelles sont les différences entre `n` et `m` ?
2. On considère la classe suivante :

```

public class TD2 {
    public static void main(String[] args) {
        Integer un = new Integer(1);
        un.afficher();
        Integer deux = new Integer(2);
        deux.afficher();
        un.afficher();
    }
}

```

Qu'obtient-on dans le terminal lorsqu'on exécute TD2 ?

3. Dans la méthode `main` précédente, on souhaite modifier la valeur de la propriété de classe `n`. Comment procéder ?
4. On rajoute maintenant les lignes suivantes :

```
int x = 1;
int y = x;
y = 3;
System.out.println(x);
Integer trois = un; trois.setM(3);
un.afficher();
Qu'obtient-on dans le terminal?
```