

Feuille de TD/TP n° 2

Classes et objets - 1

Exercice 1 : Qu'affiche le programme suivant ?

```
public class Ex1 {
    int a = 0 ;

    public void f() {
        a++ ;
    }
    public String toString() {
        return String.valueOf(a) ;
    }
    public static void main(String[] args) {
        Ex1 p = new Ex1() ;
        Ex1 q = new Ex1() ;
        System.out.println(p.toString()) ;
        System.out.println(q.toString()) ;
        p.f() ;
        System.out.println(p.toString()) ;
        System.out.println(q.toString()) ;
        p = q ;
        p.f() ;
        System.out.println(p.toString()) ;
        System.out.println(q.toString()) ;
    }
}
```

Exercice 2 : Qu'affiche le programme suivant ?

```
public class Ex3 {
    int a = 0 ;

    public void f(int a) {
        System.out.println(a) ;
    }
    public static void main(String[] args) {
        Ex3 p = new Ex3() ;
        p.f(12) ;
    }
}
```

Dans la méthode `f`, comment faire si l'on veut également afficher la valeur de l'attribut `a` ?

Exercice 3 : Concevoir une classe `Rationnel` permettant de manipuler des nombres rationnels.

- Ajouter des méthodes permettant de modifier la valeur du numérateur et du dénominateur.

- Ajouter une méthode `afficheFrac` permettant d'afficher le rationnel sous forme de fraction, et une méthode `afficheDec` pour l'afficher sous forme décimale.
- Ajouter une méthode de prototype `Rationnel addition(Rationnel r)` additionnant deux rationnels, et une méthode `void ajouter(Rationnel r)` ajoutant le rationnel donné à celui de l'objet.
- Ajouter une méthode de prototype `Rationnel multiplier(Rationnel r)` multipliant deux rationnels.
- Comment serait-il possible de garantir que les rationnels représentés correspondent toujours à des fractions irréductibles ?

Exercice 4 :

1. Ajouter à la classe `Point` la méthode `origine`, de type de retour `boolean` qui teste si les coordonnées du point sont nulles.
2. Ajouter une méthode `egale` telle que `p.egale(q)` renvoie `true` si et seulement si les abscisses et ordonnées des points `p` et `q` sont égales.
3. Ajouter une méthode `symetrie` telle que `p.symetrie()` renvoie un nouvel objet `Point` qui représente le symétrique du point `p`, dans une symétrie centrale par rapport à l'origine du repère.
4. Ajouter une méthode `distance` qui calcule la distance entre le point et un autre point passé en paramètre. Vous utiliserez la méthode `Math.sqrt` (voir l'API).

Exercice 5 : Définir une classe `Segment` qui représente un segment de droite défini par deux objets de la classe `Point` représentant les extrémités du segment.

- Ajouter deux méthodes permettant de mettre à jour les points-extrémités.
- Ajouter une méthode `longueur` qui renvoie la longueur du segment.
- Ajouter une méthode `egale` telle que `s1.egale(s2)` renvoie `true` si et seulement si les deux extrémités sont égales.
- Ajouter une méthode `affiche` qui affiche la définition du segment.