JAVA Interface

Interface

- Une interface regroupe des méthodes vues comme des services proposés par une classe
- Mot clé interface à la place de Class
- Deux interfaces peuvent hériter l'une de l'autre

Interface: contenu

- Une interface est une classe entièrement faite de **membres publics** qui sont
  - des **méthodes abstraites,**
  - **variables statiques finales** (c'est-a-dire des constantes de classe).
- pas besoin d'écrire les qualifieurs public et abstract devant les méthodes, ni public, static et final devant les variables.
- Le corps de ces méthodes n'est **pas défini dans l'interface juste signature +type**

Implémentation d'une interface

- Une interface est une spécification : elle fixe la liste des méthodes qu'on est certain de trouver dans toute classe qui déclare être conforme à cette spécification.
- Une classe *implémente* une interface lorsqu'elle offre toutes les méthodes publiques définies dans l'interface, elle est alors obligée de définir le corps de chacune des méthodes dont l'en-tête figure dans l'interface.
- Mot clé implements
- Une classe peut implémenter plusieurs interfaces
- Une classe qui implémente une interface peut aussi avoir d'autres méthodes
- les méthodes des interfaces sont toujours publiques (implicitement) ; par conséquent, leurs définitions dans des sous-classes doivent être explicitement qualifiées public, sinon une erreur sera signalée.

Interface ≈ héritage

- une interface est aussi une sorte d'héritage.
- Cet héritage est multiple : une classe peut implémenter plusieurs interfaces distinctes

Interface: exemple (1)


```
public interface CalculGeometrique {
    public double calculAire();
    public double calculPerimetre(); }
```

Interface : exemple (2)


```
public class Polygone implements CalculGeometrique {
    …
     public double calculPerimetre() {
          …
    }
     public double calculAire() {
          …
    }
    …
}
```

Interface : exemple (3)


```
public class Cercle implements CalculGeometrique {
    …
     public double calculAire(){
          return Math.PI*rayon*rayon;
    }

    public double calculPerimetre(){
          return 2*Math.PI*rayon;
    }
    …
}
```

Variable de type interface

- On peut déclarer un objet de type interface
- Quand on met une variable dans un objet de type interface on restreint ses messages aux méthodes de l'interface
- Ceci permet de faire des tableaux d'éléments de type interface ou des fonctions de paramètres de type interface

Variable de type interface: exemple


```
public static CalculGeometrique findLargest(CalculGeometrique o1,
CalculGeometrique o2)
{
    if (o1.calculPerimetre()>o2.calculPerimetre())
        return o1;
    else
        return o2;
}
```

Variable de type interface: exemple (2)


```
Point point1 = new Point(1., 1.);
Point point2 = new Point(2., 3);
Point point3 = new Point(-1., -2.);
Cercle C1 = new Cercle(3, point2);
Cercle C2 = new Cercle(1, point1);
Polygone P1 = new Polygone();
P1.addPoint(point1);
P1.addPoint(point2);
P1.addPoint(point3);
CalculGeometrique maxiPerimetre = findLargest(C1, P1);
System.out.println("L'objet de pus grand perimetre est " + maxiPerimetre);
```

Dans la fonction main

Variable de type interface: exemple


```
// L'objet de plus grande surface
CalculGeometrique[] tab = new CalculGeometrique[3];
tab[0] = C1; tab[1] = P1;
tab[2] = C2;
double surface = 0;
CalculGeometrique maxiSurface = null;
for (int i=0; i<tab.length; i++)
    if (surface < tab[i].calculAire()) {
        surface = tab[i].calculAire();
        maxiSurface = tab[i];
    }
System.out.println("L'objet de pus grande surface est " + maxiSurface);
```

Dans la fonction main