

EYA SAHBENI
TAKWA RABEI
2EME EEA

TP2 Les abstract & les interface

Application 1 : Interfaces

```
import java.util.*;

interface Dbervices {
    void addDb();

    void deleteDb();
}

abstract class Forme{
    int color;
    Forme(int color){this.color = color;}

    abstract float surf();
}

class Rect extends Forme implements Dbervices{
    float l, L;
    float surf(){ return L*l;}
    public Rect(float l, float L, int color){
        super(color);  this.l=l; this.L=L;
    }
    public void addDb(){
        System.out.println("Ajouter avec succes du "+"Rect [color], long.[L] et largeur [l]");
    }
    public void deleteDb(){
        System.out.println("Supp. avec succes du "+"Rect [color], long.[L] et largeur [l]");
    }
}

class Cercle extends Forme implements Dbervices{
    float r;
    float surf(){ return (float)(3.14*r*r);}
    public Cercle(float r, int color){
        super(color);  this.r=r;
    }
    public void addDb(){
        System.out.println("Ajout avec succès du"+" Cercle [color] du rayon [r]");
    }
    public void deleteDb(){
```

```

        System.out.println("Supp. avec succès du"+" Cercle [color] du rayon [r]");
    }
}
public class TestForme{

    public static void main(String []args){

        // Forme x = new Form(); // ERR
        //x.surf();

        Vector<Forme> dessin = new Vector<Forme>();
        dessin.add(new Rect(100,50,1));
        dessin.add(new Cercle(10 ,3));
        dessin.add(new Rect(70,20 ,4));
        dessin.add(new Rect(100,50,1));
        dessin.add(new Cercle(15 ,7));

        for (Forme F : dessin)
            ((DbServices)F).addDb();

    }
}

```

Ajouter avec succes du Rect [color], long.[L] et largeur [l]
Ajout avec succès du Cercle [color] du rayon [r]
Ajouter avec succes du Rect [color], long.[L] et largeur [l]
Ajouter avec succes du Rect [color], long.[L] et largeur [l]
Ajout avec succès du Cercle [color] du rayon [r]

Application 2 : Gestion polymorphique

```

import java.util.Vector;
interface Emprutable{
    void emprunter();
    void retourner();
    void listerEmprunts();
}

// Sous-classe Livre
class Livre implements Emprutable {
    int nbpages;
    String titre;
    String auteur;

    public Livre(int nbpages, String titre, String auteur) {
        this.nbpages = nbpages;
        this.titre = titre;
        this.auteur = auteur;
    }

    @Override
    public void emprunter() {

```

```

@Override
public void retourner() {

}

@Override //implémentation de la méthode abstract afficherInfo
public void listerEmprunts(){
    System.out.println("Livres : "+titre+", Auteurs : "+auteur); {

}
}
}
class Mag implements Empruntable {
    int num;
    String titre;
    String auteur;

    public Mag(String titre, int num) {
        this.titre = titre;
        this.num = num;
    }

    @Override
    public void emprunter() {

    }

    @Override
    public void retourner() {

    }

    @Override //implémentation de la méthode abstract afficherInfo
    public void listerEmprunts(){
        System.out.println("Mag : "+titre+", Auteurs : "+auteur); {

    }
}
}
// Programme principal
public class Main {
    public static void main(String[] args) {
        Vector<Empruntable>bib = new Vector<Empruntable>();
        bib.add(new Livre(100, "Le Petit Prince", "Antoine de Saint-Exupéry"));
        bib.add(new Mag("Science & Vie", 123));

        for (Empruntable e : bib)
            if (e instanceof Livre)
                e.listerEmprunts();
    }
}

```

}
}

Livre : Le Petit Prince, Auteur : Antoine de Saint-Exupéry