

TP2 : Les abstract & les interface

Exercice 1 :

code :

```
import java.util.*;

// Interface des services de base de données
interface DbServices {
    void addDB();
    void deleteDB();
}

// Classe abstraite Forme
abstract class Forme {
    int color;
    Forme(int color) {
        this.color = color;
    }
    abstract float surf();
}

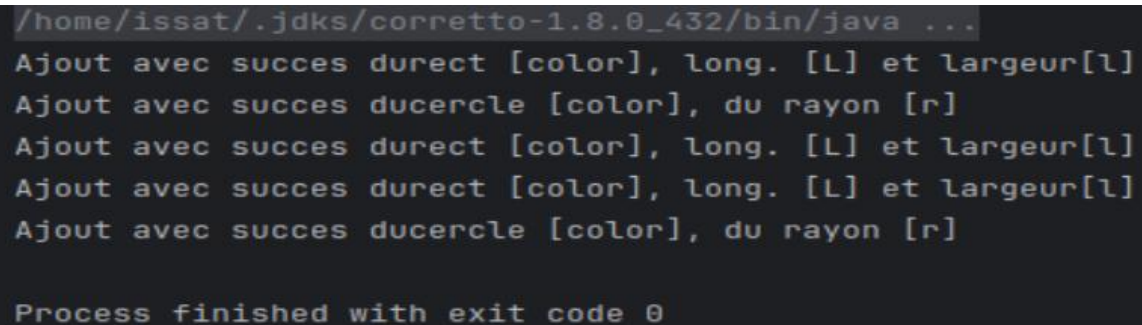
// Classe Rectangle
class Rect extends Forme implements DbServices {
    float l, L;
    public Rect(float l, float L, int color) {
        super(color);
        this.l = l;
        this.L = L;
    }
    @Override
    float surf() {
        return L * l;
    }
    @Override
    public void addDB() {
        System.out.println("Ajout avec succès du rectangle - Couleur: " + color +
```

```
        ", Longueur: " + L + ", Largeur: " + l);
    }
    @Override
    public void deleteDB() {
        System.out.println("Suppression avec succès du rectangle - Couleur: " + color +
            ", Longueur: " + L + ", Largeur: " + l);
    }
}
// Classe Cercle
class Cercle extends Forme implements DbServices {
    float r;
    public Cercle(float r, int color) {
        super(color);
        this.r = r;
    }
    @Override
    float surf() {
        return (float) (3.14 * r * r);
    }
    @Override
    public void addDB() {
        System.out.println("Ajout avec succès du cercle - Couleur: " + color +
            ", Rayon: " + r);
    }
    @Override
    public void deleteDB() {
        System.out.println("Suppression avec succès du cercle - Couleur: " + color +
            ", Rayon: " + r);
    }
}

// Classe de test
```

```
public class TestForme {  
    public static void main(String[] args) {  
        Vector<Forme> dessin = new Vector<>();  
        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();  
        }  
    }  
}
```

}Affichage :



```
/home/issat/.jdkcs/corretto-1.8.0_432/bin/java ...  
Ajout avec succes direct [color], long. [L] et largeur[l]  
Ajout avec succes ducercle [color], du rayon [r]  
Ajout avec succes direct [color], long. [L] et largeur[l]  
Ajout avec succes direct [color], long. [L] et largeur[l]  
Ajout avec succes ducercle [color], du rayon [r]  
  
Process finished with exit code 0
```

Exercice 2 :

code :

```
import java.util.Vector;  
  
// Interface pour les objets empruntables  
interface Emprutable {  
    void emprunter();  
    void retourner();  
    void listerEmprunts();  
}  
  
// Interface des services de base de données  
interface DbServices {  
    void addDB();
```

```
        void deleteDB();
    }

    // Classe abstraite Ouvrage
    abstract class Ouvrage {
        protected String titre;

        public Ouvrage(String titre) {
            this.titre = titre;
        }

        public abstract void afficherInfo();
    }

    // Sous-classe Livre
    class Livre extends Ouvrage implements Empruntable, DbServices {
        private String auteur;
        private int nbPages;

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

        @Override
        public void afficherInfo() {
            System.out.println("Livre : " + titre + ", Auteur : " + auteur + ", Pages : " + nbPages);
        }

        @Override
        public void emprunter() {
            System.out.println("Le livre '" + titre + "' a été emprunté.");
        }
    }
}
```

```
}
```

```
@Override
```

```
public void retourner() {
```

```
    System.out.println("Le livre " + titre + " a été retourné.");
```

```
}
```

```
@Override
```

```
public void listerEmprunts() {
```

```
    System.out.println("Livre emprunté : " + titre);
```

```
}
```

```
@Override
```

```
public void addDB() {
```

```
    System.out.println("Ajout dans la base : Le livre " + titre + "");
```

```
}
```

```
@Override
```

```
public void deleteDB() {
```

```
    System.out.println("Suppression de la base : Le livre " + titre + "");
```

```
}
```

```
}
```

```
// Sous-classe Magazine
```

```
class Magazine extends Ouvrage implements Emprunable, DbServices {
```

```
    private int numero;
```

```
    public Magazine(String titre, int numero) {
```

```
        super(titre);
```

```
        this.numero = numero;
```

```
}
```

```
@Override
```

```
public void afficherInfo() {
```

```
    System.out.println("Magazine : " + titre + ", Numéro : " + numero);
```

```
}
```

```
@Override
public void emprunter() {
    System.out.println("Le magazine '" + titre + "' a été emprunté.");
}

@Override
public void retourner() {
    System.out.println("Le magazine '" + titre + "' a été retourné.");
}

@Override
public void listerEmprunts() {
    System.out.println("Magazine emprunté : " + titre);
}

@Override
public void addDB() {
    System.out.println("Ajout dans la base : Le magazine '" + titre + "'");
}

@Override
public void deleteDB() {
    System.out.println("Suppression de la base : Le magazine '" + titre + "'");
}
}

// Programme principal
public class Main {
    public static void main(String[] args) {
        Vector<Emprunable> bib = new Vector<>();
        bib.add(new Livre("Le Petit Prince", "Antoine de Saint-Exupéry", 96));
        bib.add(new Magazine("Science & Vie", 123));
        // Emprunter tous les ouvrages
        for (Emprunable e : bib) {
            e.emprunter();
        }
    }
}
```

```
// Lister les emprunts  
for (Emprunable e : bib) {  
    e.listerEmprunts();  
}  
  
// Ajouter à la base de données  
for (Emprunable e : bib) {  
    ((Dbsevice) e).addDB();  
}  
}
```

}Affichage :

```
/home/issat/.jdkcs/corretto-1.8.0_432/bin/java ...  
Le livre 'Le Petit Prince' a été emprunté.  
Le magazine 'Science & Vie' a été emprunté.  
Livre emprunté : Le Petit Prince  
Magazine emprunté : Science & Vie  
Ajout Le livre Le Petit Prince  
Ajout Le livre Science & Vie  
  
Process finished with exit code 0
```