HERITAGE MULTIPLE D'INTERFACES

interface Coloriable {
      abstract public void colorier();
}

interface Tracable {
      abstract public void tracer();
}

interface Dessinable extends Coloriable, Tracable {
      void dessiner();
}
Un objet Dessinable doit pour être colorié, tracé et dessiné.

interface Jettable {
      public abstract void jeter();
}

Préambule:
public class Forme implements Dessinable, Jettable
public class Image extends Forme
public class Cercle extends Forme
public class Ligne extends Forme

```
// REDONDANT public class Image extends Forme implements Dessinable, Jettable {
```
public class Image extends Forme {
      ....
      public void colorier() { ... }
      public void tracer() { ... }
      public void dessiner() { ... }
      public void jeter() { ... }
}

```
public class Test {
    public static void main(String args[]) {
        Image i1 = new Image(); // colorier tracer dessiner manger
        Image i2 = new Image(); // colorier tracer dessiner manger
        Forme i3 = new Image(); // héritage de classes méthodes de formes
        Dessinable i4 = new Image(); // héritage d'interface colorier tracer dessiner
        Jettable i5 = new Image(); // héritage d'interface manger
        Coloriable i6 = new Image(); // héritage d'interface colorier
        Tracable i7 = new Image(); // héritage d'interface  tracer
```

```java
List<Image> images = new ArrayList<Image>();
images.add(i1);
images.add(i2);

for (Image image : images) {
        image.colorier();
        image.dessiner();
        image.tracer();
        image.manger();
}

List<Dessinable> dessinables = new ArrayList<Dessinable>();
dessinables.add(i1);
dessinables.add(i2);
dessinables.add(i4);

for (Dessinable dessinable  : dessinables) {
        dessinable.colorier();
        dessinable.dessiner();
        dessinable.tracer();
    }
}
```