**Ingénierie inverse**

L’objectif de cet exercice est de voir comment avec un outil comme *StarUML* on peut générer un diagramme de classes UML à partir d’un code Java. On parle dans ce cas *d’ingénierie inverse ou de rétro-conception*.

Q1 Reprenez le code java ci-dessous dans un outil comme *Eclipse*

Q2 Générez automatiquement avec *StarUML* le diagramme de classes correspondant au code java

Q3 Comparer le résultat obtenu avec la théorie (les règles de transformation)

Il n’y a pas de cardinalités sur les associations entre Rectangle, Losange, Cercle et la Point.

Côté navigabilité on devrait avoir 1 pour le centre et 2 pour les sommets.

Les associations précédentes pourraient être élevées au rang de composition ?

public interface Dessinable {

public void dessiner ( );

public void effacer ( );

}

abstract public class Figure implements Dessinable {

protected String couleur;

protected String getCouleur ( ) { return couleur; }

protected void setCouleur ( String c ) { couleur = c; }

}

public class Point {

private float x;

private float y;

public float getX ( ) { return x; }

public float getY ( ) { return y; }

public void Point ( float x, float y) { ... }

}

public class Cercle extends Figure {

private float rayon;

private Point centre;

public Cercle ( Point centre, float rayon) { ... }

public void dessiner ( ) { ... }

public void effacer ( ) { ... }

}

public class Rectangle extends Figure {

protected Point sommets[] = new Point[2];

public Rectangle ( Point p1, Point p2) { ... }

public void dessiner ( ) { ... }

public void effacer ( ) { ... }

}

public class Losange extends Figure {

protected Point sommets[] = new Point[2];

public Losange ( Point p1, Point p2) { ... }

public void dessiner ( ) { ... }

public void effacer ( ) { ... }

}