C. Recanati
Interfaces Graphiques
INFO2, 2012

Elements de correction pour la feuille de TP n° 3

```
// Constants.java
import java.awt.*;

public interface Constants {

    // Les types d'elements
    int LIGNE = 101;
    int RECTANGLE = 102;
    int CERCLE = 103;
    int COURBE = 104;

    // Les conditions initiales
    int DEFAULT_ELEMENT_TYPE = RECTANGLE;
    Color DEFAULT_ELEMENT_COLOR = Color.green;

}

// SketcherFrame.java sans la barre de menu

import java.awt.*;
import javax.swing.*;

public class SketcherFrame extends JFrame
                      implements Constants {

    private String title ;
    private Color elementColor = DEFAULT_ELEMENT_COLOR;
    private int elementType = DEFAULT_ELEMENT_TYPE;

    public SketcherFrame(String titre) {
        setTitle(titre);
        title = titre ;
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}

//  Sketcher.java

import java.awt.*;

public class Sketcher {
    private SketcherModel model;
    private SketcherFrame window;
    private Sketcher  theApp;

    public static void main(String[] args) {
```

```java
        theApp = new Sketcher();
        theApp.init();
    }
    public SketcherFrame getWindow() {
        return window;
    }
    public SketcherModel getModel() {
        return model;
    }
    public void init() {
        window = new SketcherFrame("Dessin");
                Toolkit leKit = window.getToolkit();
                Dimension wndSize = leKit.getScreenSize();
                window.setBounds(wndSize.width/4, wndSize.height/4,
                                   wndSize.width/2, wndSize.height/2);

        model = new SketcherModel();
        window.setVisible(true);
    }
}

//  SketcherModel.java

import java.util.*;

class SketcherModel {
    protected LinkedList ListeElements = new LinkedList();

        public boolean remove(Element element) {
            boolean removed = ListeElements.remove(element);
            return removed;
        }

        public void add(Element element) {
            ListeElements.add(element);
        }

}

//  Element.java

import java.awt.*;
import java.awt.geom.*;

public abstract class Element {

    protected Color color;

    public Element(Color color) {
        this.color = color;
    }

    public Color getColor() {
        return color;
    }
    public abstract Shape getShape();
    public abstract java.awt.Rectangle getBounds();
    public abstract void modify(Point debut, Point fin);

    public static class Ligne extends Element {

        private Line2D.Double  ligne;

        public Ligne(Point debut, Point fin, Color couleur) {
            super(couleur);
            ligne = new Line2D.Double(debut, fin);
        }

        public Shape getShape() {
            return ligne;
        }

        public java.awt.Rectangle getBounds() {
            return ligne.getBounds();
        }

        public void modify(Point debut, Point fin) {
            ligne.x1 = debut.x;
            ligne.y1 = debut.y;
            ligne.x2 = fin.x;
            ligne.y2 = fin.y;
        }

    }

    public static class Rectangle extends Element {
        private Rectangle2D.Double rectangle;

        public Rectangle(Point debut, Point fin, Color couleur) {
            super(couleur);
            rectangle = new Rectangle2D.Double (
                Math.min(debut.x, fin.x), Math.min(debut.y, fin.y),
Math.abs(debut.x - fin.x), Math.abs(debut.y - fin.y)); }
        public Shape getShape() {
            return rectangle;
        }
        public java.awt.Rectangle getBounds() {
            return rectangle.getBounds();
        }
        public void modify(Point debut, Point fin) {
            rectangle.x = Math.min(debut.x, fin.x);
            rectangle.y = Math.min(debut.y, fin.y);
            rectangle.width = Math.abs(debut.x - fin.x);
            rectangle.height = Math.abs(debut.y - fin.y);
        }
    }

    public static class Cercle extends Element {
        private Ellipse2D.Double cercle;
```

```java
        public Cercle(Point centre,
                        Point circonference, Color couleur) {
            super(couleur);
            double rayon = centre.distance(circonference);
            cercle = new Ellipse2D.Double (centre.x - rayon,
                                centre.y - rayon, 2.*rayon, 2.*rayon);
         }
        public Shape getShape() {
            return cercle;
        }
        public java.awt.Rectangle getBounds() {
            return cercle.getBounds();
        }
        public void modify(Point centre, Point circonference) {
            double rayon = centre.distance(circonference);
            cercle.x = centre.x - (int)rayon;
            cercle.y = centre.y - (int)rayon;
            cercle.width = cercle.height = 2*rayon;
        }
    }

    public static class Courbe extends Element {
        private GeneralPath courbe;

        public Courbe(Point debut, Point next, Color couleur) {
            super(couleur);
            courbe = new GeneralPath();
            courbe.moveTo(debut.x, debut.y);
            courbe.lineTo(next.x, next.y); // ou bien debut.x, debut.y
         }
        public Shape getShape() {
            return courbe;
        }
        public java.awt.Rectangle getBounds() {
            return courbe.getBounds();
        }
        public void modify(Point debut, Point next) {
            courbe.lineTo(next.x, next.y);  // prolonger la courbe
        }
    }

}
```