**C. Recanati**

**Java Graphique**

**AIR2**

# Eléments pour la feuille de TP n° 1

Exercice 1

import java.awt.\*;

import javax.swing.\*;

public class Dessin extends JFrame {

 public static void main(String args[]) {

 Panneau panel;

 Dessin fenetre = new Dessin();

 Toolkit tlk = fenetre.getToolkit();

 Dimension dim = tlk.getScreenSize();

 fenetre.setBounds(dim.width/4, dim.height/4,

 dim.width/2, dim.height/2);

 fenetre.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

 panel = new Panneau();

 fenetre.getContentPane().add(panel, BorderLayout.CENTER);

 fenetre.setVisible(true);

 }

}

class Panneau extends JPanel {

 public void paintComponent(Graphics g) {

 Graphics2D g2d = (Graphics2D) g;

 Stroke stk = new BasicStroke(5); // épaisseur de ligne

 super.paintComponent(g);

 setBackground(Color.white); //fond du panneau

 g2d.setColor(Color.red); // pour dessiner

 g2d.setStroke(stk); // pour épaisseur de ligne

 g2d.drawLine(0,0, 100, 150);

 g2d.setColor(Color.blue);

 Font f = new Font ("Serif", Font.BOLD, 36);

 g2d.setFont(f);

 g2d.drawString( "UNE CHAINE", 50,50);

 g2d.setColor(Color.green);

 g2d.drawRect(10, 10, 90, 200);

 }

}

Exercice 2

1) public interface Constantes {

 public static final int ROND = 1;

 public static final int CROIX = 2;

 public static final int POLY = 3;

 public static final int defaultMode = rond;

}

public class Test implements Constantes {

protected static int mode = defaultMode; // ou ROND

 public static void main (String [ ] args) {

 System.out.println ("le mode est" + mode) ;

 }

}

2)

import java.util.\*;

import java.awt.\*;

public class Test2 implements Constantes {

 Vector ensPts ; // stocke les points à afficher

 int mode;

 private void init(int i) {

 mode=i;

 ensPts = new Vector();

 }

 public Test2(int i) {

 this.init(i);

 }

 public Test2() {

 this.init(defaultMode);

 }

 public void initAlea(int nb, int larg, int haut) {

 for (int i=0; i<nb ; i++) {

 Point p = new Point(

 (int) Math.round(Math.random()\*larg),

 (int) Math.round(Math.random()\*haut));

 ensPts.addElement(p);

 }

 }

 public static void main (String[] args) {

 int largeur=800,hauteur=500;

 Test2 f = new Test2();

 f.initAlea(12, largeur, hauteur);

for (Enumeration e = f.ensPts.elements() ;

 e.hasMoreElements() ;)

{ Point pt = (Point) e.nextElement();

System.out.println("(" + pt.x + ","

 + pt.y + ")");

 }

 }

}

3) et 4)

import java.awt.\*;

import javax.swing.\*;

import java.util.\*;

public class Dessin extends JFrame {

 Vector ensPts ; // stocke les points à afficher

 /\* mode d'affichage de l'ensemble des points

 ROND : petits ronds

 CROIX : petites croix

 POLY : comme polygone

 \*/

 int mode;

 MonJPanel pane; // le panneau dans lequel on dessine

// constantes définies normalement dans une interface

 final static int ROND=0,CROIX=1,POLY=2;

 static int modeMin=ROND, modeMax=POLY;

 private void init(int i) {

 mode=i;

 ensPts = new Vector();

 this.setBackground(Color.white);

 }

 public Dessin(String titre, int i)**throws Exception {**

 **// traitement pour la dernière question**

 super(titre);

 if (i < modeMin || i > modeMax)

 **throw new Exception("mode " + i +" incorrect");**

 this.init(i);

 Container content = getContentPane();

 pane = new MonJPanel();

 content.add(pane, BorderLayout.CENTER);

 }

 public Dessin(String titre) {

 super(titre);

 this.init(rond);

 Container content = getContentPane();

 pane = new MonJPanel();

 content.add(pane, BorderLayout.CENTER);

 }

 public class MonJPanel extends JPanel {

 public void paintComponent(Graphics g) {

 super.paintComponent(g) ;

 g.setPaint(Color.red);

 int nbPoints = ensPts.size();

 int[] abs = new int[nbPoints];

 int[] ords = new int[nbPoints];

 for(int i=0; i<nbPoints; i++) {

 Point p= (Point)ensPts.elementAt(i);

 abs[i]=p.x;

 ords[i]=p.y;

 }

 switch(mode){

 case ROND :

 for(int i=0; i<nbPoints; i++) {

 g.drawOval(abs[i], ords[i], 5, 5);

 }

 break;

 case CROIX :

 for(int i=0; i<nbPoints; i++) {

 g.drawLine(…);

 g.drawLine(…);

 }

 break;

 case POLY :

 …

 break;

 } // end switch

 } // end paintComponent

 } // end MonJPanel

 public void initAleatoire(int nb,

 int larg, int haut) {

 for (int i=0; i<nb ; i++) {

 Point p = new Point(

 (int) Math.round(Math.random()\*larg),

 (int) Math.round(Math.random()\*haut));

 ensPts.addElement(p);

 }

 }

 public void setPaneSize(int largeur, int hauteur) {

 pane.setSize(largeur, hauteur);

 pane.setPreferredSize(

 new Dimension(largeur,hauteur));

 }

 public static void main (String args[]) {

 int largeur=800,hauteur=500;

 for (int i=Dessin.modeMin;

 i<=Dessin.modeMax ; i++){

 **try {**

Dessin f = new Dessin("Dessin de points "

 + i,i);

 f.setPaneSize(largeur, hauteur);

 f.initAleatoire(12, largeur, hauteur);

 f.pack();

 f.setLocation(20\*i, 20\*i);

 f.setVisible(true);

 **}**

 **catch (Exception e){ System.out.println(e);**

 **}**

 }

 }

}

Exercice 3

import java.io.\* ;

public class Test {

protected String getClassName(Object o) {

 String classString = o.getClass().getName();

 int dotIndex = classString.lastIndexOf(".");

 return classString.substring(dotIndex+1);

 }

 public static void main (String [ ] args) {

 Test essai = new Test () ;

 Object obj = System.out ;

 System.out.println (essai.getClassName(obj)) ;

 }

}