

## Corrigé du TP 2 : programme de dessin

### 1 Classes des formes

#### 1.1 Classe Carre

```
import java.awt.Graphics;
import java.awt.Point;

class Carre
{
    Point coin;
    int cote;

    Carre(int x, int y, int c)
    {
        coin = new Point(x,y);
        cote = c;
    }

    void dessiner(Graphics g)
    {
        g.drawRect(coin.x, coin.y, cote, cote);
    }
}
```

#### 1.2 Classe Cercle

```
import java.awt.Graphics;
import java.awt.Point;

class Cercle
{
    Point centre;
    int rayon;

    Cercle(int x, int y, int r)
    {
        centre = new Point(x,y);
        rayon = r;
    }

    void dessiner(Graphics g)
```

```

    {
        g.drawOval(centre.x - rayon, centre.y - rayon, 2*rayon, 2*rayon);
    }
}

```

### 1.3 Classe Triangle

```

import java.awt.Graphics;
import java.awt.Point;

class Triangle
{
    Point[] p = new Point[3];

    Triangle(int x1, int y1, int x2, int y2, int x3, int y3)
    {
        p[0] = new Point(x1, y1);
        p[1] = new Point(x2, y2);
        p[2] = new Point(x3, y3);
    }

    void dessiner(Graphics g)
    {
        int[] px = { p[0].x, p[1].x, p[2].x };
        int[] py = { p[0].y, p[1].y, p[2].y };
        g.drawPolygon(px, py, 3);
    }
}

```

## 2 Programme interactif

```

import javax.swing.JApplet;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Point;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;

// On déclare que la classe Dessin "est un" JApplet. En plus, on
// déclare que notre classe Dessin implémente toutes les
// fonctionnalités d'un "MouseListener". C'est a dire, qu'elle
// implémente toutes les méthodes nécessaires pour gérer les clics de
// la souris. Regardez la documentation de "MouseListener" dans la
// doc Java.
public class Dessin extends JApplet implements MouseListener
{
    // Le tableau contenant les cercles
    Cercle cercles[] = new Cercle[100];
    // Nombre de cercles dans le tableau
    int nbCercles=0;
}

```

```

// Cette méthode "init" est toujours appelé automatiquement par
// JApplet après le démarrage du programme. On pourra y mettre le
// code dont on a besoin pour initialiser notre programme. Dans
// notre cas on voudra juste initialiser la gestion de la souris.
public void init()
{
    System.out.println("Bonjour, je m'initialise");

    // Ici on dit à l'applet qu'on veut recevoir les évènements
    // souris. A partir de maintenant JApplet sait que s'il y a un
    // évènement de souris, il peut appeler des méthodes de la
    // classe Dessin associés à l'objet courant (this).
    addMouseListener(this);
}

// Toutes les méthodes qui suivent sont associés à l'interface
// MouseListener. Les 4 premières ne nous intéressent pas pour
// l'instant (mais elles doivent obligatoirement être présentes ..
// sinon, on ne serait pas conforme à l'interface MouseListener).
// Toutes ces méthodes vont être appelés automatiquement par
// JApplet si les évènements correspondants surviennent.
public void mouseEntered (MouseEvent e) {};
public void mouseExited (MouseEvent e) {};
public void mousePressed (MouseEvent e) {};
public void mouseReleased(MouseEvent e) {};
public void mouseClicked (MouseEvent e)
{
    System.out.println("Position:" + e.getX() + "," + e.getY());
    cercles[nbCercles] = new Cercle(e.getX(), e.getY(), 20);
    ++nbCercles;
    repaint();
}

public void paint (Graphics g)
{
    // effacer avant de tout redessiner
    g.clearRect(0,0,getWidth(),getHeight());
    for(int i=0; i<nbCercles; ++i)
    {
        cercles[i].dessiner(g);
    }
}
}

```

## 2.1 Ajout de la couleur

### 2.1.1 Nouvelle classe Cercle

Il suffit de rajouter un attribut de type `Color` à la classe `Cercle` et de modifier la méthode `dessiner` pour que la couleur soit prise en compte.

```

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Point;

```

```

class Cercle
{
    Point centre;
    int rayon;
    Color couleur;

    Cercle(int x, int y, int r)
    {
        centre = new Point(x,y);
        rayon = r;
    }

    void dessiner(Graphics g)
    {
        g.setColor(couleur);
        g.drawOval(centre.x - rayon, centre.y - rayon, 2*rayon, 2*rayon);
    }
}

```

### 2.1.2 Nouvelle classe Dessin

Seule la méthode mouseClicked est modifiée, elle devient :

```

public void mouseClicked (MouseEvent e)
{
    System.out.println("Position:" + e.getX() + "," + e.getY());
    cercles[nbCercles] = new Cercle(e.getX(), e.getY(), 20);
    cercles[nbCercles].couleur = new Color( (27*nbCercles)% 255, (58*nbCercles)%255, (238*nbCercles)%255);
    ++nbCercles;
    repaint();
}

```