



INSTITUT SUPERIEUR POLYTECHNIQUE DE MADAGASCAR  
TP JAVA n°3 – Dessin et Animation

## Partie A – Dessin Simple

```
1 package dessin;
2
3 import java.awt.Color;
4 import java.awt.Graphics;
5 import javax.swing.*;
6
7 public class Dessin extends JFrame{
8     public Dessin(){
9         super("ISPM is the BEST");
10        this.setSize(400,400);
11        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
12        this.setVisible(true);
13    }
14    @Override
15    public void paint(Graphics g){
16        super.paint(g);
17        g.drawLine(0, 0, this.getWidth(), this.getHeight());
18        g.setColor(Color.BLUE);
19        g.drawOval(50,50,100,100);
20        g.setColor(new Color(255,0,0,150));
21        g.fillOval(100,100,100,100);
22    }
23    public static void main(String[] args) {
24        new Dessin();
25    }
26 }
```

- 1) Copier le code ci-dessus et exécuter.
- 2) Expliquer ce qui se passe dans les lignes 8 à 13. Modifier la taille de fenêtre en 800x600.
- 3) Après avoir expérimenté le code, en effectuant plusieurs modifications, expliquer brièvement le fonctionnement des méthodes suivantes :
  - a. setSize(...)
  - b. drawLine(...)
  - c. setColor(...)
  - d. drawOval(...)
  - e. fillOval(...)
- 4) Modifier le code source de la méthode public void paint(...) {...} (lignes 15..22) pour montrer le dessin d'une maison et d'un bonhomme à côté.  
Votre sens de l'esthétique sera noté.

## Partie B – Animation Simple

```
1 package dessin;
2 import javax.swing.*;
3 import java.awt.*;
4 import java.awt.event.*;
5
6 public class AnimationSimple extends JPanel implements ActionListener {
7     private int x; // position horizontale de la balle
8     private int vitesse;
9     public AnimationSimple() {
10         x = 0;
11         vitesse = 1;
12         // Timer déclenchant l'animation toutes les 10 millisecondes
13         Timer timer = new Timer(10, this);
14         timer.start();
15     }
16     @Override
17     protected void paintComponent(Graphics g) {
18         super.paintComponent(g);
19         g.fillOval(x, 50, 30, 30);
20     }
21     @Override
22     public void actionPerformed(ActionEvent e) {
23         x += vitesse;
24         if (x > getWidth()) { // on atteint le bord droit
25             x = -30;
26         }
27         repaint();
28     }
29     public static void main(String[] args) {
30         JFrame frame = new JFrame("Animation Simple");
31         AnimationSimple animation = new AnimationSimple();
32         frame.add(animation);
33         frame.setSize(400, 200);
34         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
35         frame.setVisible(true);
36     }
37 }
```

- 5) Copier le code ci-dessus et exécuter. Décrire l'animation qui s'affiche à l'écran.
- 6) Que fait la méthode `repaint()` ?
- 7) Modifier la méthode `void actionPerformed(...){...}` de manière à ce que la balle inverse sa vitesse à chaque fois qu'elle atteint une extrémité de la fenêtre (gauche ou droite). La balle fera, alors un va-et-vient de gauche à droite.