

Programmation Graphique en Swing PO2 (2015-2016)

Chapitre 5 Dessins

Les principes de base du dessin

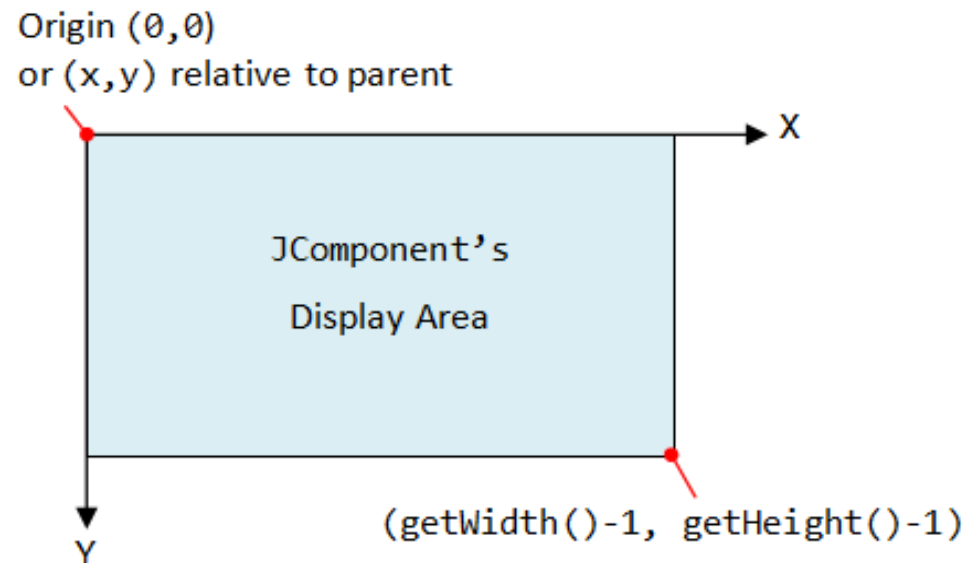
- Système de coordonnées
- Graphics
- Color

Dessiner

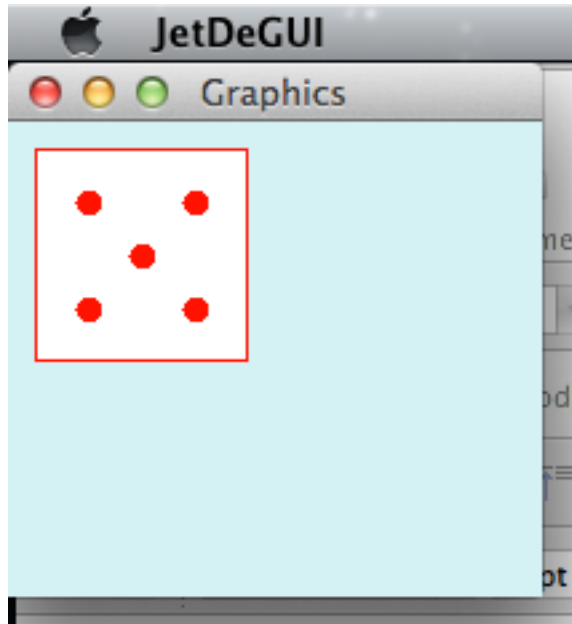
- Pour dessiner, il faut une toile, un crayon et une mesure :
 - La mesure : le système de coordonnées
 - La toile : la zone de dessin (JPanel)
 - Le crayon : le contexte graphique (Graphics)

Systeme de coordonnees

- Le systeme de coordonnees de Java utilise le pixel comme unite de mesure. Toutes les valeurs de pixels sont des entiers.
- Les coordonnees d'un point sont par rapport au conteneur ou se situe ce point, donc un point pourra avoir plusieurs coordonnees selon de differents conteneurs.



Systeme de coordonnees



La taille de 'de' : `java.awt.Dimension[width=80,height=80]`

Les coordonnees du point en haut et en gauche de ce dé dans 'de' (JPanel) : `[x=0,y=0]`

Les coordonnees du point en haut et en gauche de ce dé dans 'panel' (JPanel) : `java.awt.Point[x=10,y=10]`

Les coordonnees du point en haut et en gauche de ce dé dans 'this' (JFrame) : `java.awt.Point[x=10,y=54]`

Systeme de coordonnees

```
public class JetDeGUI extends JFrame {
    private DePanel de;
    private JPanel panel;
public JetDeGUI() {
    this.setTitle("Graphics");
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setPreferredSize(new Dimension(200, 200));
    this.setBackground(new Color(204, 238, 241));

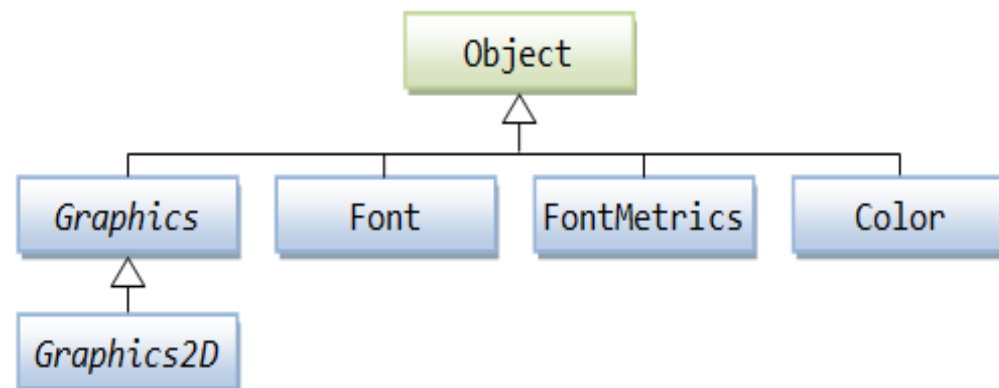
    panel = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 10));
    de = new DePanel();
    panel.add(de);

    this.setContentPane(panel);
    this.pack();
    this.setVisible(true);
}
```

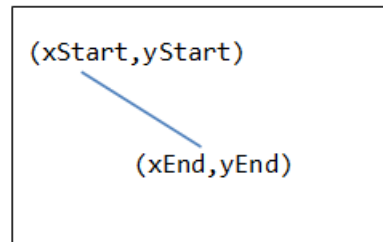
```
System.out.println("La taille de 'de' : "+
de.getSize());
System.out.println("Les coordonnees du
point en haut et en gauche de ce de dans
'de' (JPanel) : [x=0,y=0]");
System.out.println("Les coordonnees du point en
haut et en gauche de ce de dans 'panel'(JPanel) :
"+de.getLocation());
System.out.println("Les coordonnees du point en
haut et en gauche de ce de dans 'this' (JFrame) :
"+de.getLocationOnScreen());
}
```

Graphics

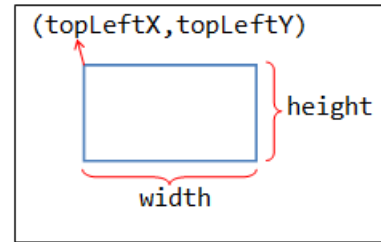
- Graphics contient des méthodes permettant de dessiner des textes ou des formes sur une zone de dessin (JPanel).



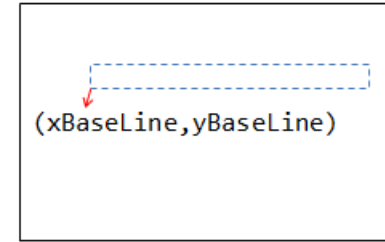
Graphics



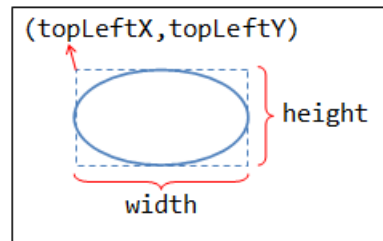
drawLine()



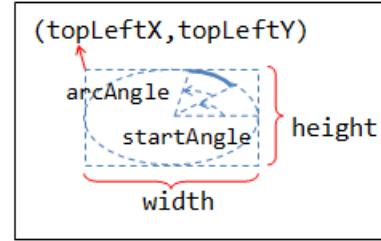
drawRect()



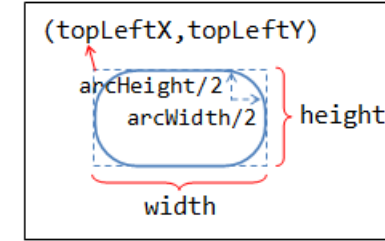
drawString()



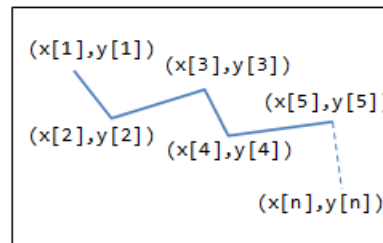
drawOval()



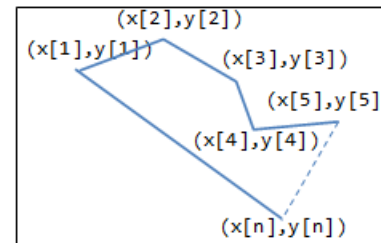
drawArc()



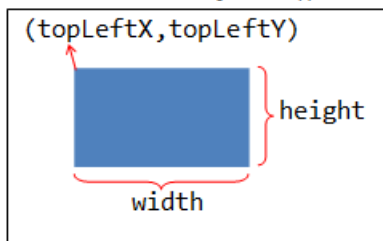
drawRoundRect()



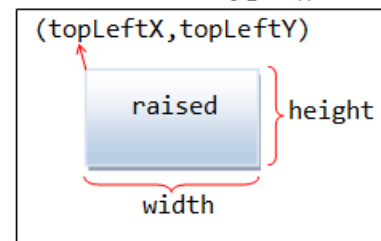
drawPolyline()



drawPolygon()



fillRect()



fill3DRect()

Graphics

- Etapes pour dessiner une figure:
 - Définir une classe de JPanel
 - Redéfinir la méthode `paintComponent(Graphics g)` de JPanel
 - Utiliser `repaint()` pour obliger le composant JPanel à se redessiner

Exemple : DePanel

```
public class DePanel extends JPanel {
    private static final int SPOT_DIAMETER = 10; // Diameter of spots
    private int val;

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        int w = getWidth(); // Panel height and width
        int h = getHeight();
        .....
        g.drawRect(0, 0, w-1, h-1); // Draw border
        switch (val) {
            case 1: drawSpot(g, w/2, h/2);
                    break;
            case 3: drawSpot(g, w/2, h/2);
                    // go to next case
            case 2: drawSpot(g, w/4, h/4);
                    drawSpot(g, 3*w/4, 3*h/4);
                    break;
            case 5: drawSpot(g, w/2, h/2);
                    // go to next case
            case 4: drawSpot(g, w/4, h/4);
                    drawSpot(g, 3*w/4, 3*h/4);
                    drawSpot(g, 3*w/4, h/4);
                    drawSpot(g, w/4, 3*h/4);
                    break;
            case 6: drawSpot(g, w/4, h/4);
                    drawSpot(g, 3*w/4, 3*h/4);
                    drawSpot(g, 3*w/4, h/4);
                    drawSpot(g, w/4, 3*h/4);
                    drawSpot(g, w/4, h/2);
                    drawSpot(g, 3*w/4, h/2);
                    break;
        }
    }

    private void drawSpot(Graphics g, int x, int y) {
        g.fillOval(x-SPOT_DIAMETER/2, y-SPOT_DIAMETER/2, SPOT_DIAMETER,
                  SPOT_DIAMETER);
    }
}
```

Color

- Lorsqu'on dessine sur un composant, le composant lui attribue une couleur, et cette couleur peut être modifiée par la méthode :
 - `setColor(Color)` : fixer une nouvelle couleur de dessin

Color

- La classe Color :
 - 13 couleurs standards : Color.RED, Color.GREEN, etc
 - Color(int r, int g, int b) : une couleur avec les proportions de rouge (r), de vert (g) et de bleu (b) spécifiées. Chacun de ces trois nombres doit être compris entre 0 et 255

