

Imagens Gráficas e Java 2D

Série Java – Cartão de Referência

1. Herança das classes principais:

```
java.lang.Object
java.awt.Component
  java.awt.Container
    javax.swing.JComponent
      javax.swing.JPanel
```

```
java.lang.Object
java.awt.Component
  java.awt.Canvas
```

```
java.lang.Object
java.awt.Graphics
  java.awt.Graphics2
```

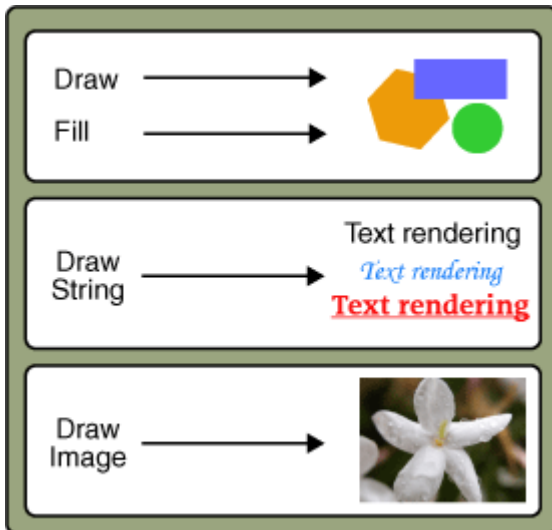
2. Classe Básica para utilização

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

public class Basico extends JFrame {
    public Basico() {
        super("Básico");
        this.setSize(300,300);
        this.setVisible(true);
    }
    public void paint(Graphics g) {
        super.paint(g);
        // outros métodos aqui
    }
    public static void main(String [] args) {
        new Basico(); } }
```

Dica:

Utilize o método **repaint()** é utilizado para redesenhar a tela.



Desenhando Figuras

```
g.drawLine(20, 100, 120, 100); // Uma Linha
g.drawRect(20, 100, 120, 100); // Um retângulo
g.drawOval(20, 100, 120, 100); // Um círculo
```

Desenhando Strings

```
g.drawString("Hello", 10, 10);
```

Desenhando Imagens

```
g2.drawImage(img1, 0, 0, size.width,
size.height, 0, 0, img1.getWidth(null),
img1.getHeight(null), null);
```

Linhas, Retângulos e Ovais

```
drawLine(int x1, int y1, int x2, int y2)
drawRect(int x, int y, int w, int h)
fillRect(int x, int y, int w, int h)
clearRect(int x, int y, int width, int height)
drawRoundRect(int x, int y, int w, int h,
int arcw, int arch)
fillRoundRect(int x, int y, int w, int h,
int arcw, int arch)
draw3DRect(int x, int y, int w, int h,
boolean raised)
fill3DRect(int x, int y, int w, int h,
boolean raised)
drawOval(int x, int y, int w, int h)
fillOval(int x, int y, int w, int h)
```

Polígonos e Objetos Especializados

```
drawPolygon(int[] xPoints, int[] yPoints, int nPoints)
drawPolyline(int[] xPoints, int[] yPoints, int nPoints)
drawArc(int x, int y, int w, int h, int startAngle,
int arcAngle)
fillArc(int x, int y, int w, int h, int startAngle, int
arcAngle)
fillPolygon(int[] xPoints, int[] yPoints, int nPoints)
drawString(String str, int x, int y)
drawBytes(byte[] data, int offset, int length, int x,
int y)
drawChars(char[] data, int offset, int length, int x,
int y)
drawImage(Image img, int x, int y, Color bgcolor,
ImageObserver observer)
```

Dica:

Utilize o método **setColor(Color c)** para modificar a cor antes de criar um objeto.

Objeto java.awt.Color – Objeto Color através de Atributos Estáticos

Color.BLACK	Color.BLUE	Color.CYAN	Color.DARK_GRAY
Color.GRAY	Color.GREEN	Color.LIGHT_GRAY	Color.MAGENTA
Color.ORANGE	Color.PINK	Color.RED	Color.WHITE

Imagens Gráficas e Java 2D

Série Java – Cartão de Referência

[] Color.YELLOW

Outra Forma: new Color(R, G, B); // RGB varia de 0 a 255

Desenhando um retângulo com uma determinada cor específica:

```
...
Rectangle2D boundsTemp = font.getStringBounds(tempString, frc);
Rectangle2D boundsCond = font.getStringBounds(condStr, frc);
int wText = Math.max((int)boundsTemp.getWidth(), (int)boundsCond.getWidth());
int hText = (int)boundsTemp.getHeight() + (int)boundsCond.getHeight();
int rX = (size.width-wText)/2;
int rY = (size.height-hText)/2;
g.setColor(Color.LIGHT_GRAY);
g2.fillRect(rX, rY, wText, hText);
...
```

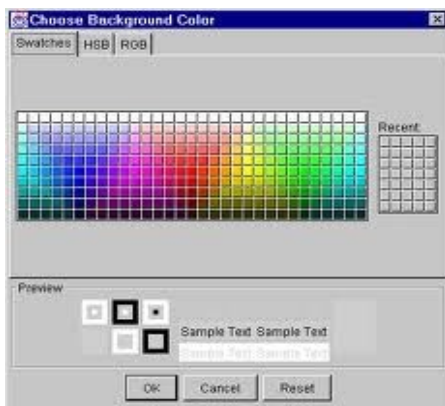
Objeto java.awt.Font – Atributos Estáticos

Font.BOLD

Font.ITALIC

Font.PLAIN

Font.TRUETYPE_FONT



Janela de Cor

```
color = JColorChooser.showDialog(this, "Sua cor", null);
```

O método aceita três argumentos:

- Uma referência ao componente **pai**.
- Uma String para exibir na barra de título
- Objeto Color inicial selecionada para o diálogo

Por exemplo, para determinar o valor de uma janela utilizamos:

```
this.setBackground(Color c);
```

Brincando com fontes

```
g.setFont(new Font("SansSerif", Font.BOLD, 12));
g.setFont(new Font("Arial", Font.ITALIC, 14));
```

Desenhando Polígonos

```
int x1Points[] = {0, 100, 0, 100};
int y1Points[] = {0, 50, 50, 0};
GeneralPath pol =
    new GeneralPath(GeneralPath.WIND_EVEN_ODD, x1Points.length);
pol.moveTo(x1Points[0], y1Points[0]);
for (int index = 1; index < x1Points.length; index++)
    pol.lineTo(x1Points[index], y1Points[index]);
pol.closePath();
g2.draw(pol);
```



Mais algumas imagens

```
final static float dash1[] = {10.0f};
final static BasicStroke dashed = new BasicStroke(1.0f,
    BasicStroke.CAP_BUTT, BasicStroke.JOIN_MITER, 10.0f, dash1,
    0.0f);
g2.setStroke(dashed);
g2.draw(new RoundRectangle2D.Double(x, y, rectWidth,
    rectHeight, 10, 10));

redtowhite = new GradientPaint(0,0,color.RED,100,
    0,color.WHITE);
g2.setPaint(redtowhite);
g2.fill(new Ellipse2D.Double(0, 0, 100, 50));
```



Última Dica: Pesquise sobre o pacote **java.awt.geom**