

Java 程序设计

第 14 章 图形、图像与音频





导读

主要内容

- 绘制基本图形
- 图形的布尔运算
- 绘制钟表
- 绘制图像
- 播放音频

重点和难点

- 重点：绘制基本图形和播放音频
- 难点：绘制基本图形



- Component 类有一个方法 **public void paint(Graphics g)**，程序可以在其子类中重写这个方法。当程序运行时，java 运行环境会用 **Graphics2D**（Graphics 的一个子类）将参数 g 实例化，对象 g 就可以在重写 paint 方法的组件上内绘制图形、图像等
- Graphics2D 的“画笔”分别使用 **draw** 和 **fill 方法** 来绘制和填充一个图形。
- 组件都是矩形形状，组件本身有一个默认的坐标系，组件的**左上角的坐标值是 (0,0)**。

例子1绘制了太极图和四边形，效果如图 14.2 。



图 14.2 绘制基本图形





```
class MyCanvas extends JPanel {  
    public void paint(Graphics g) {  
        Graphics2D g2D = (Graphics2D)g;  
        Arc2D arc = new Arc2D.Double(0,0,100,100,-90, -180, Arc2D.PIE);  
        g2D.setColor(Color.blue);  
        g2D.fill(arc);  
  
        arc = new Arc2D.Double(150,150,100,100,-90, -180, Arc2D.OPEN);  
        g2D.setColor(Color.red);  
        g2D.draw(arc);  
    }  
}
```



- 有时需要平移、缩放或旋转一个图形。可以使用 AffineTransform 类来实现对图形的这些操作。例如：
 - `AffineTransform trans=new AffineTransform();`
- 将图形沿顺时针或逆时针以 (x, y) 为轴点旋转 number 个弧度：
 - `trans.rotate(60.0*3.1415927/180,100,100);`
 - `g_2d.setTransform(trans);`

例子2 旋转椭圆和字符串，效果如图 14.3 。

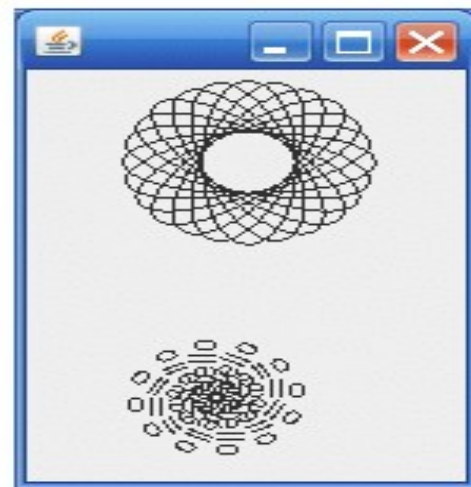


图 14.3 · 旋转





```
class MyCanvas extends JPanel {
```

```
    public void paint(Graphics g) {
```

```
        Graphics2D g2D = (Graphics2D)g;
```

```
        Ellipse2D e = new Ellipse2D.Double(30, 30, 150, 50);
```

```
        g2D.setColor(Color.blue);
```

```
        g2D.fill(e);
```

```
        AffineTransform trans = new AffineTransform();
```

```
        trans.translate(100, 250);
```

```
        g2D.setTransform(trans);
```

```
        g2D.draw(e);
```

```
    }
```

```
}
```



- 两个图形进行布尔运算之前，必须分别用这两个图形创建两个 Area 区域对象，例如：
 - `Area a1 = new Area(T1);`
 - `Area a2 = new Area(T2);`
- a1 就是图形 T1 所围成的区域； a2 就是 T2 所围成的区域。那么，a1 调用 add 方法：
 - `a1.add(a2);`
- 之后， a1 就变成 a1 和 a2 经过布尔“或”运算后的图形区域。可以用 Graphics2D 对象 g 来绘制或填充一个 Area 对象（区域）：
 - `g.draw(a1);`
 - `g.fill(a1);`

例子3绘制图形的布尔运算，效果如图 14.4 。

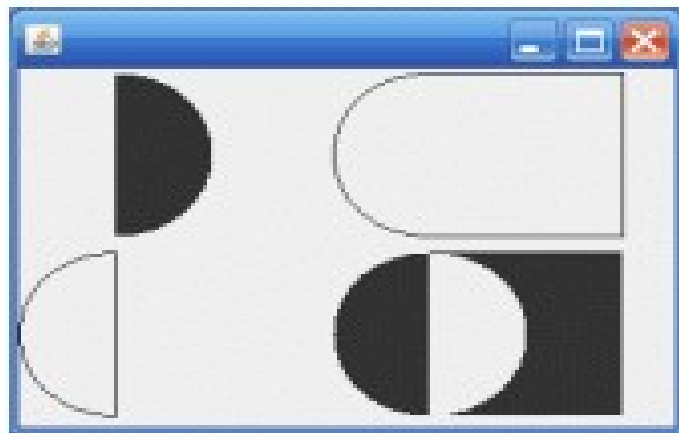


图 14.4 布尔运算





```
class MyCanvas extends JPanel {
```

```
    public void paint(Graphics g) {
```

```
        Graphics2D g2D = (Graphics2D)g;
```

```
        Ellipse2D e2D = new Ellipse2D.Double(30, 30, 150, 50);
```

```
        g2D.setColor(Color.blue);
```

```
        g2D.draw(e2D);
```

```
        Area a1 = new Area(e2D);
```

```
        Rectangle2D r2D = new Rectangle2D.Double(70, 30, 150, 50);
```

```
        g2D.setColor(Color.red);
```

```
        g2D.draw(r2D);
```

```
        Area a2 = new Area(r2D);
```

```
        a1.intersect(a2);
```

```
        g2D.setColor(Color.green);
```

```
        g2D.fill(a1);
```

```
    }
```

```
}
```



1. 加载图像

组件调用 `getToolkit ()` 方法可以返回这个对象的引用。 `Tollkit` 类的对象调用方法 `Image getImage(String fileNme)`
或 `Image getImage(File file)`。

可以返回一个 `Image` 对象，该对象封装着参数 `file`（或参数 `fileName`）指定的图像文件。

2. 绘制图像

`public boolean drawImage(Image img,int x,int y,ImageObserver observer);`

参数 `img` 是被绘制的 `Image` 对象， `x`、 `y` 是要绘制指定图像的矩形的左上角所处的位置， `observer` 是加载图像时的图像观察者。

例子5绘制了一幅图像，
并更改了窗口左上角的
咖啡图像



图 14.6 绘制图像。





```
class ImageCanvas extends Canvas
```

```
{
```

```
    Toolkit tool;
```

```
    Image image;
```

```
    ImageCanvas() {
```

```
        setSize(200, 200);
```

```
        tool = getToolkit();
```

```
        image = tool.getImage("D:\\1.jpg");
```

```
    }
```

```
    public void paint(Graphics g) {
```

```
        g.drawImage(image, 10, 10, image.getWidth(this),
```

```
                image.getHeight(this), this);
```

```
    }
```

```
}
```



播放音频的步骤如下。

(1) 创建 File 对象

```
File musicFile=new File("hello.wav");
```

(2) 获取 URI 对象（ URI 类属于 java.net 包）

```
URI uri=musicFile.toURI();
```

(3) 获取 URL 对象

```
URI url=uri.toURL();
```

(4) 创建音频对象 (AudioClip 和 Applet 类属于 java.applet 包)

```
AudioClip clip=Applet.newAudioClip(url);
```

(5) 播放，循环与停止

clip.play() 开始播放，

clip.loop() 循环播放，

clip.stop() 停止播放。

例子6(Example14_6.java
)， AudioClipDialog.java
在应用程序中播放音频，
界面效果如图 14.7



图 14.7 播放音频



1. 制作 JPG 图像文件

制作 JPG 图像步骤如下：

- (1) 用 `java.awt.image` 包中的 `BufferedImage` 类建立一个 `BufferedImage` 对象。
- (2) `BufferedImage` 对象调用 `createGraphics()` 获得一个 `Graphics2D` 对象。
- (3) `Graphics2D` 对象调用相应的方法绘制图形。
- (4) `JPEGCodec` 用 `createJPEGEncoder(OutputStream out)` 返回 `JPEGImageEncoder` 对象。
- (5) `JPEGImageEncoder` 用 `encode(Image image)` 将 `BufferedImage` 对象写入到输出流

`JPEGCodec` 类和 `ImageEncoder` 类在 `c`

例子 7 将绘制的椭圆和抛物线保存为名字是 `my.jpg` 的图像文件。



图 14.8 `my.jpg` 图像文件

2. 弹奏音节

例子 8(PlayMusicWindow.java , MusicButton.java) 中用户用鼠标单击 7 个按钮或敲击键盘对应的数字键，程序播放音乐的 7 个音节。效果如图 14.9 。

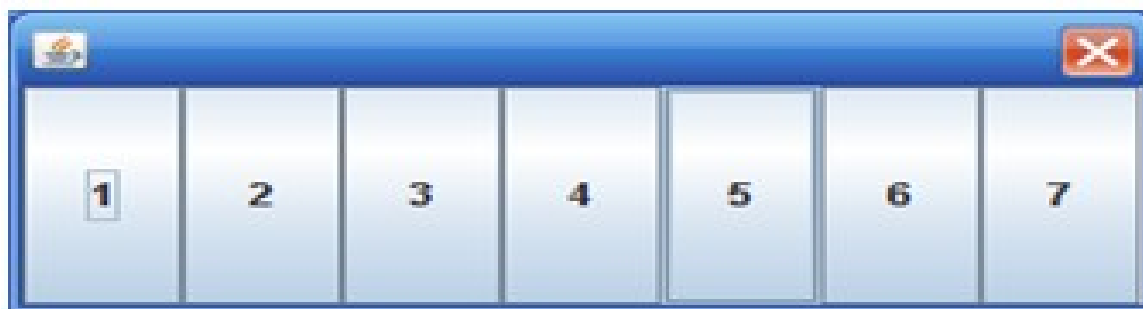


图 14.9 播放音节

```

class AudioClipDialog extends JDialog implements Runnable, ItemListener, ActionListener {
    Thread thread;
    JComboBox choiceMusic;
    AudioClip clip;
    JButton btnPlay, btnLoop, btnStop;
    String str;

```

```

    AudioClipDialog() {
        thread = new Thread(this);
        choiceMusic = new JComboBox();
        choiceMusic.addItem("选择音频文件");
        choiceMusic.addItem("1.wav");
        btnPlay = new JButton("播放");
        btnPlay.addActionListener(this);
        setLayout(new FlowLayout());
        add(choiceMusic);
        add(btnPlay);
        setSize(350, 120);
    }

    public void actionPerformed(ActionEvent e) {
        System.out.println(e.getSource().toString());
        if (e.getSource() == btnPlay)
        {
            System.out.println(clip);
            clip.play();
        }
    }

    public void itemStateChanged(ItemEvent e) {
        str = choiceMusic.getSelectedItem().toString();
        if (!(thread.isAlive()))
            thread = new Thread(this);
        try {
            thread.start();
        }
        catch (Exception ex) {}
    }

    public void run() {
        try {

```

```

public class Less0525 {
    public static void
    main(String[] args) {
        AudioClipDialog dialog =
        new AudioClipDialog();
        dialog.setVisible(true);
    }

```





1 使用 Graphics 类或其子类 Grapgics2D 类绘制各种基本图形图像

2 在应用程序中可以播放 .au 、 .aiff 、 .wav 、 .midi 、 .rfm 格式的音频。

