

Java 语言的绘图功能

玛依拉* 王克宏

摘要 本文主要对 Java 语言的绘图功能进行了介绍,对如何利用 Java 语言类库中的 Java. awt. image(AWT image Package,AWT 图形包)绘制图形和显示图形及字体进行了介绍。

关键字 Java Graphics

1 Java 语言

Java 语言在这样短的时间里,受到人们的如此欢迎,是由 Java 语言所具有的特点而决定的。它不仅仅是一种网络编程语言,而且广泛应用于各个领域。Java 语言中,对象是以类的形式实现的。一个类声明可以看作是用于创建类的模板,就是说,建立一个类声明后,我们就可以基于这个类声明创建尽可能多的对象,从这个意义上说,类可以看成是 Java 语言中的一个基本类型,只不过这个类的基本类型的具体操作需要由编程人员提供。

Java 的设计使它适合于一个不断发展的环境,在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行,并且 Java 通过接口来支持多重继承,使之比严格的类继承具有更灵活的方式和扩展性。用 Java 语言编写绘制的图形是那样的逼真,你可以看见咖啡杯子正在冒着热气,而且杯子还能转动,这是别的程序设计语言目前所不能做到的。

为了简化程序设计,Java 提供了一整套用于实现各种基本功能和其它某些常用功能的 Java 编程的标准类库,即 Java API(Java 应用程序编程接口),这个库的功能包括网络设备编程、图形用户界面编程、操作系统编程、编译器编程等等。它为 Java 语言编程人员提供了巨大的便利。

2 Java 语言绘图功能

Java 语言类库中的 AWT 包,即 Java. awt. image(AWT image Package,AWT 图形包)。提供了用于图形处理的类和接口。例如类 ColorModel、RGBImageFilter、接口 ImageObserver 等等,用于产

生和处理图形与色彩的程序会经常用到这个包。

因为 Java 语言具有丰富的图形功能和交互功能,还支持多媒体。我们利用 Java. awt. image 包可以方便地实现各种绘图功能。Graphics 类是一个处理各种图形对象的基本抽象类,Graphics 类是画图的关键,它支持 2 种绘图:基本绘图(如:画线、画矩形、画圆等等)和画图象(主要用于动画的制作)。

用 Java 语言绘图时,和别的编程语言和绘图程序一样,在绘图的平面区域上我们想象有一个二维坐标,通过它就可以把绘图和写字位置表示清楚,在 Java 中坐标原点 (0,0) 位于整个绘图平面的左上角,从原点 (0,0) 出发向右为 x 轴,向下为 y 轴。所有的坐标都必须是正整数,不能带有小数,一个单位代表一个象素。所以要把一个图象或字放在屏幕的一个指定位置,只要指定出其 x 和 y 坐标即可。

2.1 字型设置

字型设置在 Java 语言中是比较简单的。在 Java 字型设置中类 Font 是很重要的。如要在 Java 小应用程序上输出文字,在我们要确定显示的字型时,就要用到类 Font 来决定显示文字时的字型,要产生类 Font 的对象实体,其使用的构造函数是:Font (String fontName,int style,int size)。其中:参数 fontName 表示所要选的字体的名称,例如英文的有:"TimesRoman"、"Courier"等等,中文的有:"明细体"等等。参数 style 为字体的表现形式,正常字体 (PLAIN)、粗体 (BOLD) 等等。参数 size 是用来指定字体的大小的。

2.2 图形及字型的显示

图形及字型的显示方法可以用 Graphics 类中的方法:

(1) `drawString(String str,int x,int y)`;其功能为使用当前图形对象的字体和颜色输出字符串 `str`。其中参数:`str` 表示被输出的字符串;`x` 和 `y` 分别表示 `x` 和 `y` 轴坐标。

(2) `drawChars(char data [],int offset,int length,int x,int y)`;其功能为使用当前图形对象的字体和颜色输出字符串数组 `data[]` 中的指定部分。其中参数:`data` 表示被输出的字符串数组;`offset` 表示起始下标;`length` 表示输出字节数;`x` 和 `y` 分别表示 `x` 和 `y` 轴坐标。

(3) `drawBytes(byte data [],int offset,int length,int x,int y)`;其功能为使用当前图形对象的字体和颜色输出字节数组 `data` 中的指定部分。其中参数:`data` 表示被输出的字节数组;`offset` 表示起始下标;`length` 表示输出字节数;`x` 和 `y` 分别表示 `x` 和 `y` 轴坐标。

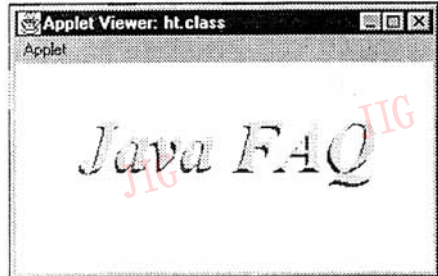
2.3 颜色设置

如果要显示彩色的字体或图形,当然效果更亮丽了。要显示出彩色的字,首先要设置要显示的字的颜色,产生颜色的方法和产生字体的方法是很相似的。在 Java 小应用程序上输出彩色的文字,我们除了确定显示的字型外,还要用到类 `Color`,在 Java 中 `Color` 类将颜色按照 RGB 格式进行封装。RGB 格式中,红蓝绿 3 原色的取值范围都是 0~255。若某种颜色取值为 0,则表示该颜色不起任何作用;若取值为 255,则表示起最大作用。如果将 3 种颜色进行叠加就可以生成一种新的颜色,合成的 RGB 颜色的值为"000000"表示黑色,"ffffff"表示白色。首先调用 `Color` 的构造函数,产生一个 `Color` 对象,再用 `Graphics` 类里的 `SetColor` 把它设置到 `Graphics` 对象上。而且我们还可以创造出自己想要的颜色,当然 `Color` 中已经定义好的各种颜色常数我们也可以方便地直接使用。我们还可以利用 2 种反差比较大的颜色。使字体看起来具有立体感。下面的 Java 小应用程序 `applet` 用以上的几种方法,编写出了能够显示具有立体感彩色字体的程序:

```
import java.awt.Graphics;
import java.awt.Font;
public class ht extends java.applet.Applet
public void init() {
    resize(300,150);
}
public void paint(Graphics g)
    Font ltz=new Font ("TimesRoman",Font.BOLD
```

```
+Font.ITALIC,40);
g.set Font(ltz);
g.setColor(Color.red);
g.drawString("Java FAQ",45,80);
g.setColor(Color.yellow);
g.drawString("Java FAQ",47,77);
}
```

程序执行结果如下图:



另外用 Java 语言还能绘制各种几何图形:

(1) 绘制各种矩形

如果我们想绘制矩形,就要用到 Java 各种各样的基本绘图指令了,首先我们先看绘制矩形这一绘图指令,Java 提供了 3 种矩形的绘制方式:

`drawRect(int x,int y,int width,int height)`;它的功能是画出一个矩形。

`fillRect(int x,int y,int width,int height)`;其作用为用当前图形对象颜色填充参数指定的矩形。其中参数:`x` 和 `y` 分别代表矩形左上角的 `x` 坐标和 `y` 坐标;`width` 表示矩形的宽;`height` 表示矩形的高。

(2) 绘制圆型

要绘制椭圆和圆形要用到 Java 提供的绘图方式中的方法 `drawOval(int x,int y,int width,int height)` 或者 `fillOval(int x,int y,int width,int height)`,

方法 `drawOval(int x,int y,int width,int height)`;其作用为在指定的外接矩形中绘制一个圆形或椭圆。在这里椭圆的中心就是它外接矩形的中心,这个矩形的左上角坐标为(`x`,`y`)。其中参数:`x` 和 `y` 分别表示 `x` 和 `y` 轴坐标;`width` 为矩形的宽度;`height` 为矩形的高度。

方法 `fillOval(int x,int y,int width,int height)`;其作用是在指定的外接矩形中绘制一个圆形或椭圆,并使用当前图形对象的颜色对它进行填充。在这里椭圆的中心就是它外接矩形的中心,这个矩形的左上角坐标为(`x`,`y`),其中参数:`x` 和 `y` 分别

表示 x 和 y 轴坐标; $width$ 为矩形的宽度; $height$ 为矩形的高度

(3) 绘制多边形

绘制多边形用方法: `drawPolygon(int xPoints [],int yPoints [],int nPoints)` 或 `fillPolygon(int xPoints [],int yPoints [],int nPoints)`。

方法 `drawPolygon(int xPoints [],int yPoints [],int nPoints)`: 其作用为根据参数数组 `xPoints []` 和 `yPoints []` 提供的各个顶点坐标, 绘制一个多边形。这个方法绘制的多边形包括 `nPoints` 条线, 每条线从 `(xPoints[I-1], yPoints[I-1])` 到 `(xPoints[I], yPoints[I])`。最后一条线从最后一个顶点开始到第一个顶点结束。其中参数: `xPoints` 为多边形所有顶点的 x 轴坐标数组; `yPoints` 为多边形所有顶点的 y 轴坐标数组; `nPoints` 为顶点数目。

方法 `fillPolygon(int xPoints [],int yPoints [],int nPoints)`: 其作用为根据参数数组 `xPoints []` 和 `yPoints []` 提供的各个顶点坐标, 绘制一个多边形并使用当前图形对象的颜色对其进行填充。这个方法绘制的多边形包括 `nPoints` 条线, 每条线从 `(xPoints [I-1], yPoints [I-1])` 到 `(xPoints [I], yPoints [I])`。最后一条线从最后一个顶点开始到第一个顶点结束。其中参数: `xPoints` 为多边形所有顶点的 x 轴坐标数组; `yPoints` 为多边形所有顶点的 y 轴坐标数组; `nPoints` 为顶点数目。

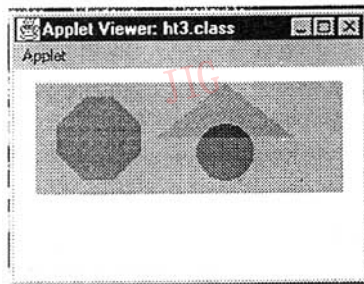
(4) 绘制重叠图形

用类 `Graphics` 中的方法 `XORMode(Color c1)` 可以绘制出相互重叠的图形。其中参数: `c1` 表示第 2 种颜色。下面是一个绘制重叠图形的 Java 小应用程序 `applet`:

```
import java.awt.Graphics;
import java.applet.*;
public class ht3 extends Applet {
    public void init() {
        resize(250,150);
    }
    public void paint(Graphics g) {
        int Dx1={30,50,70,90,90,70,50,30,30};
        int Dy1={40,20,20,40,60,80,80,60,40};
        int Dd1=9;
        int Dx2={100,150,200,100};
        int Dy2={50,10,50,50};
        int Dd2=4;
        g.setColor(Color.red);
        g.fillOval(130,40,40,40);
```

```
g.setXORMode(Color.blue);
g.drawPolygon(Dx2,Dy2,Dd2);
g.setColor(Color.lightGray);
g.fillPolygon(Dx1,Dy1,Dd1);
g.setXORMode(Color.yellow);
g.fillRect(15,10,22,80);
```

其程序运行的结果如下图:



3 Java 语言的应用前景

Java 语言以其独具的特点, 为开发者及使用者提供了良好的开发与运行环境, 由于 Java 与 Internet、特别是 WWW 的紧密结合, 使 Java 语言成为了风靡全球的程序设计语言。在许多国家 Java 语言在大学、大专院校中同样获得了相当高的声誉, 成为所有课程中最受欢迎的一门课程。在高等教育中, Java 语言扮演了重要角色, 许多大学的计算机系统把 Java 语言当作主修课程来介绍的。SUN 公司研究开发部经理 Graham Lovell 说过: “至今, 我还没有发现一个学校或者一个大学说他们对 Java 语言不感兴趣。”Java 语言在如此短的时间里在全球范围内取得了巨大的成功, 从而使网络发生了革命性的变化。用 Java 语言开发的软件可以具有可视化、可听化、可操作化的效果。因为 Java 开发的软件可做到“即时、交互、动画与动作”, 要它停就停, 要它继续就继续。通过 Java 语言这个良好的工具使产生出影视作品、故事情节、文学艺术等成为现实。Java 是一种全新的技术, 可以被利用的方式目前还没有真正挖掘出来, 它做为一种通用的程序设计语言不仅在 Internet 软件开发中起重要作用而且将会在下一个世纪的软件技术开发中起重要的作用。它对整个计算机产业将产生深远的影响。