Java 程序设计

第7章 图形程序设计

田英鑫 tyx@hit.edu.cn
哈尔滨工业大学软件学院

第7章 图形程序设计

& 本章导读

- 7.1 Swing 概述
- 7.2 创建框架
- 7.3 框架定位
- 7.4 在面板中显示信息
- 7.5 2D 图形
- 7.6 颜色
- 7.7 字体
- 7.8 图像
- 7.9 动画

第7章 图形程序设计

& 本章重点

- 7.2 创建框架
- 7.4 在面板中显示信息
- 7.9 动画

& 本章难点

- 7.5 2D 图形
- 7.7 字体

7.1 Swing 概述

- AWT（抽象窗口工具集）
  - 基本AWT库将处理用户界面元素的任务委派给每个目标平台（Windows、Solaris、Macintosh等等）的本地GUI工具箱进行处理
  - 使用AWT编写的程序观感效果依赖于目标平台
- AWT 的缺点
  - 在不同平台上的操作行为很难保持一致
  - 不能使用每个平台特有的图形元素
  - 在不同平台上的AWT用户界面库存在不同的Bug

7.1 Swing 概述

- Swing 用户界面
  - Sun和Netscape合作开发的全新的用户界面库
  - Swing并没有对AWT完全替代，Swing只是提供了更好的用户界面组件
  - Swing仍然采用AWT的事件处理模型
- Swing 的优势
  - Swing拥有一个丰富、便捷的用户界面元素集合
  - Swing对底层平台的依赖更少
  - Swing给不同平台上的用户一致的感观效果
  - 更充分体现Java程序的“一次编写，到处运行”

7.2 创建框架

- 框架
  - 在Java中，顶层窗口称为框架（Frame）
  - 框架是一种容器，可以在框架中容纳其他用户界面组件，如：按钮、文本域等
- AWT 和 Swing 中的框架
  - AWT中的框架是Frame类
  - Swing中的框架是JFrame类，它从Frame类扩展而来
  - 大多数 的Swing组件类都以“J”开头

7.2 创建框架

- 在屏幕上显式一个空的框架

例7-1:SimpleFrameTest.java

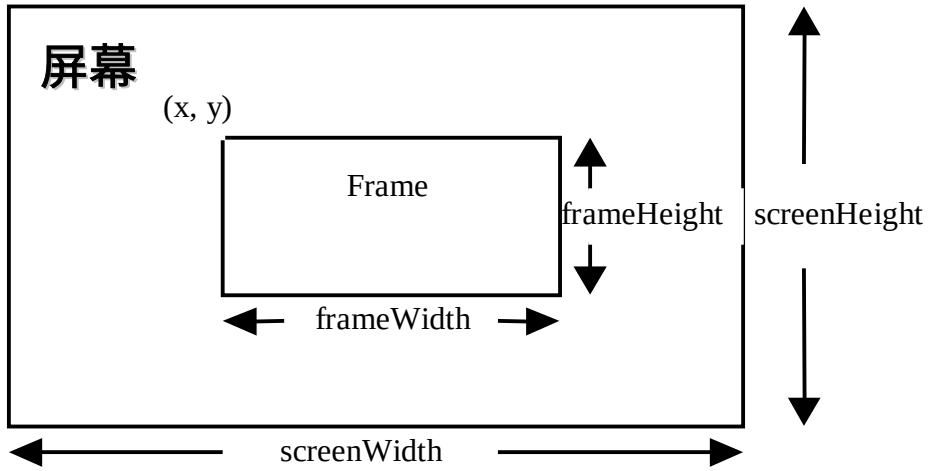
```
import javax.swing.*;
public class SimpleFrameTest {
   public static void main(String[] args) {
      SimpleFrame frame = new SimpleFrame();
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);
   }
}
class SimpleFrame extends JFrame {
   public SimpleFrame() {
      setSize(DEFAULT_WIDTH, DEFAULT_HEIGHT);
   }
   public static final int DEFAULT_WIDTH = 300;
   public static final int DEFAULT_HEIGHT = 200;
}
```

7.3 框架定位

- 框架的常用方法
  - dispose ()
  - setIconImage (Image image)
  - setTitle (String title)
  - setResizeable (boolean resizable)
  - setLocation (int x, int y)
  - setBounds (int x, int y, int width, int height)
  - getContentPane()

7.3 框架定位

- 使框架居中
  - 获取屏幕分辨率




```
Toolkit kit = Toolkit.getDefaultToolkit();
Dimension screenSize = kit.getScreenSize();
screenWidth = screenSize.width;
screenHeight = screenSize.height;
frameWidth = screenWidth / 2;
frameHeight = screenHeight / 2;
x = screenWidth / 4;
y = screenHeight / 4;
```

参见例7-2:CenteredFrameTest.java

7.4 在面板中显示信息

- 创建面板（Panel）
  - 定义一个扩展JPanel的新类
  - 重写paintComponent方法

```
class NotHelloWorldPanel extends JPanel {
   public void paintComponent(Graphics g) {
      super.paintComponent(g);
      g.drawString("Not a Hello,World program",
         MESSAGE_X,MESSAGE_Y);
   }
   public static final int MESSAGE_X = 75;
   public static final int MESSAGE_Y = 100;
}
```

参见例7-3:NotHelloWorld.java

7.5 2D 图形

- 使用 Graphics 类绘制简单图形
  - 定义一个扩展JPanel类的面板
  - 重写其paintComponent方法
  - 通过Graphics类的绘图方法绘制简单图形

```
class MyPanel extends JPanel
{
  public void paintComponent(Graphics g)
  {
    super.paintComponent(g);
    g.drawOval(80,80,200,100);
  }
}
```

7.5 2D 图形

- 使用 Java 2D 库的绘图功能
  - 获取一个Graphics2D类的对象
  - 定义要绘制的图形对象
  - 用Graphics2D类对象的draw方法将其绘制出来

```
class MyPanel extends JPanel {
   public void paintComponent(Graphics g) {
      super.paintComponent(g);
      Graphics2D g2 = (Graphics2D)g;
      Ellipse2D ellipse =
         new Ellipse2D.Double(80,80,200,100);
      g2.draw(ellipse);
  }
}
```

参见例7-4:DrawTest.java

7.6 颜色

- 设置图形前景色
  - 使用Graphics2D类的setPaint方法设置颜色
  - 接下来就按照设置好的颜色进行绘图

```
class MyPanel extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D)g;
        g2.setPaint(Color.RED);
        Ellipse2D ellipse =
            new Ellipse2D.Double(80,80,200,100);
        g2.draw(ellipse);
  }
}
```

7.6 颜色

- 填充图形
  - 使用Graphics2D类的setPaint方法设置颜色
  - 把draw函数替换为fill函数即可

```
class MyPanel extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D)g;
        g2.setPaint(Color.RED);
        Ellipse2D ellipse =
            new Ellipse2D.Double(80,80,200,100);
        g2.fill(ellipse);
  }
}
```

参见例7-5:FillTest.java

7.7 字体

- 创建字体对象
  - 根据字体名、字体风格和字体大小创建一个Font类对象
- 设置字体
  - 使用Graphics2D类的setFont方法设置字体

```
public void paintComponent(Graphics g)
{
   ...
   Font sansbold14 = new Font("SansSerif",Font.BOLD,14);
   g2.setFont(sansbold14);
   String message = "Hello,World!";
   g2.drawString(message,75,100);
}
```

参见例7-6:FontTest.java

7.8 图像

- 将图像文件读到 Java 程序中
  - 构造File类或URL类的对象，该对象是对本地图像文件或网络上图像的URL的封装
  - 使用ImageIO类的read方法构造Image对象
- 绘制图像
  - 使用Graphics类的drawImage方法绘制图像

```
String filename = "...";
Image image = ImageIO.read(new File(filename)); 或者
String urlname = "...";
Image image = ImageIO.read(new URL(urlname));
// 绘制图像
g.drawImage(image,x,y,null);
```

参见例7-7:ImageTest.java

7.9 创建动画

- 动画的原理

7.9 创建动画

- 利用定时器创建动画
  - 每隔一定的时间定时器会自动触发
  - 在Java中提供了Timer类实现定时器的功能

```
Timer t = new Timer(1000, listener);
t.start();
```

```
class MyTimerListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        //执行此处的代码
    }
}
```

7.9 创建动画

- 利用键盘控制物体运动
  - 在键盘的事件处理程序中控制坐标

```
class MyKeyListener implements KeyListener {
    public void keyPressed(KeyEvent e) {
        //执行此处的代码
    }

    public void keyReleased(KeyEvent e) {
    }

    public void keyTyped(KeyEvent e) {
    }
}
```

7.9 创建动画

- 利用鼠标控制物体运动
  - 在鼠标的事件处理程序中控制坐标

```
class MyMouseListener implements MouseListener {
    public void mousePressed(MouseEvent e) {
        //执行此处的代码
    }
    public void mouseReleased(MouseEvent e) {}

    public void mouseClicked(KeyEvent e) {}

    public void mouseEntered(KeyEvent e) {}
    public void mouseExited(KeyEvent e) {}
}
```

Any Question?