

本文链接: <http://blog.csdn.net/xietansheng/article/details/74362076>

[Java Swing 图形界面开发 \(目录\)](#)

1. 概述

官方JavaDocsApi: [javax.swing.JLabel](#)

JLabel, 标签。标签主要用于展示 **文本** 或 **图片**, 也可以 **同时显示文本和图片**。

JLabel 的构造方法:

```
1  /** 构造方法中的参数含义:
2   * text: 标签显示的 文本
3   * image: 标签显示的 图片
4   * horizontalAlignment: 标签内容 (在标签内) 的水平对其方式 (竖直方向默认居中,
   可通过方法设置)
5   */
6  JLabel()
7  JLabel(String text)
8  JLabel(String text,int horizontalAlignment)
9  JLabel(Icon image)
10 JLabel(Icon image,int horizontalAlignment)
11 JLabel(String text, Icon image,int horizontalAlignment)
```

JLabel 的常用方法:

```
1  // 设置 文本 和 图片
2  void setText(String text)
3  void setIcon(Icon icon)
4  // 设置文本相对于图片的位置 (文本默认在图片右边垂直居中)
5  void setHorizontalTextPosition(int textPosition)
6  void setVerticalTextPosition(int textPosition)
7  // 设置标签内容 (在标签内) 的对其方式 (默认左对齐并垂直居中)
8  void setHorizontalAlignment(int alignment)
9  void setVerticalAlignment(int alignment)
10 // 设置图片和文本之间的间隙
11 void setIconTextGap(int iconTextGap)
12 /* * 下面的方法定义在 JComponent 基类中 */
13 // 设置文本的字体类型、样式 和 大小
14 void setFont(Font font)
15 // 设置字体颜色
16 void setForeground(Color fg)
17 // 当鼠标移动到组件上时显示的提示文本
```

```

18 void setToolTipText(String text)
19 // 设置组件的背景
20 void setBackground(Color bg)
21 // 设置组件是否可见
22 void setVisible(boolean visible)
23 // 设置组件是否为 不透明，JLabel默认为透明，设置为不透明后才能显示背景
24 void setOpaque(boolean isOpaque)
25 // 设置组件的 首选 大小
26 void setPreferredSize(Dimension preferredSize)
27 // 设置组件的 最小 大小
28 void setMinimumSize(Dimension minimumSize)
29 // 设置组件的 最大 大小
30 void setMaximumSize(Dimension maximumSize)
31 /** 上面的对齐参数具体的取值常量参考 DocsApi;
32  * setXXX(...) 方法往往都对应着 getXXX() 方法。
33  */

```

显示图片时，**Icon**实例的创建（一般创建**ImageIcon**，**ImageIcon** 实现了 **Icon**）：

ImageIcon 的构造方法:

```

1 // 使用 本地路径 图片创建
2 ImageIconImageIcon(String filename)
3 // 使用 网络路径 图片创建
4 ImageIconImageIcon(URL location)
5 // 使用图片文件的字节数组创建
6 ImageIconImageIcon(byte[] imageData)
7 // 使用 java.awt.Image 实例创建
8 ImageIconImageIcon(java.awt.Image image)

```

java.awt.Image 实例的创建:

```

1 // 方法一：通过 java.awt.Toolkit 工具类来读取本地、网络 或 内存中 的 图片（支持 GIF、JPEG 或 PNG）
2 Image image = Toolkit.getDefaultToolkit().getImage(String filename);
3 Image image = Toolkit.getDefaultToolkit().getImage(URL url);
4 Image image = Toolkit.getDefaultToolkit().createImage(byte[] imageData);
5 // 方法二：通过 javax.imageio.ImageIO 工具类读取本地、网络 或 内存中 的 图片（BufferedImage 继承自 Image）
6 BufferedImage bufImage = ImageIO.read(File input);
7 BufferedImage bufImage = ImageIO.read(URL input);

```

```

8  BufferedImage bufImage = ImageIO.read(InputStream input);
9  /** PS_01: 图片宽高: BufferedImage 可以通过 bufImage.getWidth() 和 bufImage.getHeight() 方法直接获取图片的宽高;
10   * Image 获取宽高需要另外传入一个 ImageObserver 参数。
11   *
12   * PS_02: 图片裁剪: BufferedImage 通过 bufImage.getSubimage(int x, int y, int w, int h) 方法可以截取图片的
13   * 任意一部分, 返回一个新的 BufferedImage 实例。
14   *
15   * PS_03: 图片缩放: Image 通过 image.getScaledInstance(int width, int height, int hints) 方法可以对图片进行
16   * 缩放, 返回新的一个 Image 实例。
17   */

```

2. 代码实例

```

1  package com.xiets.swing;
2  import javax.swing.*.*;
3  import java.awt.*.*;
4  public class Main{
5      public static void main(String[] args){
6          JFrame jf =newJFrame("测试窗口");
7          jf.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
8          // 创建内容面板, 默认使用流式布局
9          JPanel panel =newJPanel();
10         /* * 只显示文本 */
11         JLabel label01 =newJLabel();
12         label01.setText("Only Text");
13         label01.setFont(newFont(null, Font.PLAIN,25));
14         // 设置字体, null 表示使用默认字体
15         panel.add(label01);
16         /* * 只显示图片 */
17         JLabel label02 =newJLabel();
18         label02.setIcon(newImageIcon("demo01.jpg"));
19         panel.add(label02);
20         /* * 同时显示文本和图片 */
21         JLabel label03 =newJLabel();
22         label03.setText("文本和图片");
23         label03.setIcon(newImageIcon("demo02.jpg"));
24         label03.setHorizontalTextPosition(SwingConstants.CENTER);
25         // 水平方向文本在图片中心

```

```
26  label03.setVerticalTextPosition(SwingConstants.BOTTOM);
27  // 垂直方向文本在图片下方
28  panel.add(label03);
29  jf.setContentPane(panel);
30  jf.pack();
31  jf.setLocationRelativeTo(null);
32  jf.setVisible(true);
33  }
34  }
```

结果展示:



result.png