

Java语言与网络编程

第8章 GUI

网络安全与网络工程系 杨东平
jsxhbc@163.com

第8章 图形用户界面

- ◆ 8.1 Java Swing概述
- ◆ 8.2 一个简单的Swing程序
- ◆ 8.3 Swing与AWT的对比
- ◆ 8.4 界面图形的绘制及文字显示与颜色控制
- ◆ 8.5 窗体、对话框与面板
- ◆ 8.6 事件处理
- ◆ 8.7 常用的组件
- ◆ 8.8 布局管理器
- ◆ 8.9 图形用户界面综合案例

网络安全与网络工程系杨东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 2

图形用户界面概述

- ✿ 图形界面作为用户与程序交互的窗口，是软件开发中一项非常重要的工作
- ✿ Java Swing 是 Java 语言中用来实现 GUI 的类库，它为用户提供了直观的、可视化的交互界面
- ✿ Swing 组件是完全用 Java 语言编写、操作和显示的
- ✿ 本章将介绍如何使用 Java 语言编写图形界面，利用 Java 丰富的图形组件创建交互性更好的用户界面

网络安全与网络工程系杨东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 3

图形用户界面概述(续)

- 无论采取何种语言、工具实现图形界面，其原理都基本相似。简单而言，图形界面就是用户界面元素的有机合成。这些元素不仅在外观上相互关联，在内在上也具有逻辑关系，通过相互作用、消息传递，完成用户操作的响应
- 设计和实现图形用户界面时，主要包含两项内容
 - ❖ 创建图形界面中需要的元素，进行相应的布局
 - ❖ 定义界面元素对用户交互事件的响应以及对事件的处理

网络安全与网络工程系杨东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 4

8.1 Java Swing概述

- ✿ Swing 是 Sun 公司推出的第二代用户图形接口包，通过 Swing 可开发出功能强大的、界面优美的客户应用程序
- ✿ Swing 不仅具有很多功能完善的组件，而且还具有良好的扩展能力

网络安全与网络工程系杨东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 5

8.1 Java Swing概述

- ✿ Swing 的发展
 - Swing项目开始于 1996 年底，用这个名称的目的在于开发一个不依赖于本地系统的图形用户接口工具包
 - Swing 于 1997 年下半年进入测试阶段，并于 1998 年春随同 JDK1.1 首次发行
 - 自 JDK1.2 之后被完全集成到 Java 的标准类库中，成为 Java 基础类库(JFC)的一个重要组成部分

网络安全与网络工程系杨东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 6

Swing库中的包及其功能

包名	功能说明
javax.accessibility	提供辅助功能的类和接口，可以与 Swing 组件进行交互，在技术上不是 Swing 的组成部分，但在 Swing 中使用非常广泛
javax.swing	提供了大量的 Swing 组件的模型接口及支持类
javax.swing.border	提供了绘制组件边界的特殊边框类，包括抽象边框类及 8 种预定义的边框实现
javax.swing.colorchooser	包含为 JColorChoser 组件提供支持的类
javax.swing.event	提供 Swing 监听器和事件的类与接口
javax.swing.filechooser	包含了为 JFileChooser 组件提供支持的类
javax.swing.plaf	提供支持不同操作系统平台显示外观风格的类，包括 Meta 和 Multi 外观风格的类，通过使用该类中的类可以调整界面的外观风格
javax.swing.table	为表格组件提供支持类，包括表格模型和视图

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 7

Swing库中的包及其功能（续）


包名	功能说明
javax.swing.text	为文本组件提供的支持类
javax.swing.text.html	为 HTML 文本编辑器类提供支持类
javax.swing.text.html.parser	提供解析 HTML 格式文本的工具类
javax.swing.text.rtf	为 RTF 文本编辑器类提供支持类
javax.swing.tree	为树状列表组件提供支持类，包括树状列表模型和视图
javax.swing.undo	提供了一些用户撤消功能的类与接口

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 8

8.2 一个简单的Swing程序

❁ 例8_1 在对话框中显示一段文字信息SwingExample.java

```
import javax.swing.*; //导入swing包
public class ShowString extends JFrame{ //扩展JFrame类
    private JLabel jLabelWelcome; //定义标签成员
    public ShowString() {
        this.setLayout(null); //设置窗体的布局管理器为null
        JLabelWelcome = new JLabel(); //创建标签对象并为其初始化显示的文字信息
        //设置标签中需要显示的文字信息
        jLabelWelcome.setText("欢迎您来到Swing的编程世界，"
            + "这将是您第一个图形界面交互程序！！");
        jLabelWelcome.setBounds(40,30,450,30); //设置标签在窗体中的位置
        this.add(jLabelWelcome); //将标签添加进窗体中
        this.setTitle("第一个Swing程序"); //设置窗体的标题、位置大小以及可见性
        this.setBounds(330,250,500,150);
        this.setVisible(true);
        this.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    }
    public static void main(String[] args) {
        new ShowString(); //创建窗体对象
    }
}
```



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程

8.3 Swing与AWT的对比

❁ AWT、Swing 作为图形界面的开发包同时存在于同一标准库中，那么二者之间的区别在哪里呢？创建图形界面时如何取舍？本节详细讲述 Swing 与 AWT 的关系，以及如何取舍 Swing 与 AWT

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 10

8.3.1 Swing与AWT之间的关系

- ❁ Swing 诞生之前，Java 中用来进行图形用户界面开发的工具包为 **AWT** (**A**bstract **W**indow **T**oolkit, **抽象窗体工具包**)
- ❁ AWT 是所有 Java 版本中都包含的基本 GUI 工具包，其中不仅提供了基本的控件，并且还提供了丰富的事件处理接口
 - Swing 是继 AWT 之后 Sun 推出的又一款 GUI 工具包。Swing 建立在 AWT 1.1 基础上的，也就是说，AWT 是 Swing 大厦的基石
- ❁ AWT 提供的组件数量有限，远没有 Swing 丰富

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 11

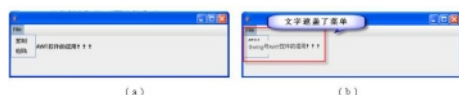
Swing与AWT之间的关系(续)

- AWT 是重量级组件，用其所编写的程序必须在每一个平台上单独测试，无法真正实现“一次编写，到处运行”
- Swing 不是为了替换 AWT，而是支持功能更丰富的开发选择，Swing 中使用的事件处理机制就是 AWT 1.1 中提供的
 - ❖ 实际开发中，组件通常只采用 Swing 的，而很多辅助类则是需要使用 AWT 的，特别是进行事件处理开发时

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 12

8.3.2 Swing与AWT控件的混用建议

- 由于 AWT 中提供的控件，均依赖本地系统实现
 - 而 Swing 控件属于轻量级控件，是由纯 Java 编写的，使用基本图形元素直接在屏幕上绘制
 - 在搭建界面时，如果将二者同时使用，就有可能出现遮挡的现象
 - 遮挡现象的原因：AWT 控件的显示优先级高于 Swing，即 AWT 控件总是绘制在 Swing 控件上面
- 例：Swing 与 AWT 控件混用 SwingAWT.java



修改前运行结果

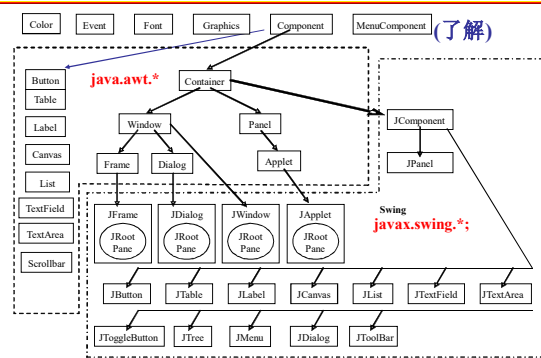
8.4 界面图形的绘制及文字显示与颜色控制

- 随着时代的发展，现代的 GUI 应用程序中或多或少的会设置一些图形或动画。这样既能美化界面，使应用程序的内容丰富多采，也可以增加应用程序的交互性。使用带有丰富图象与动画的应用程序，用户会有更好的体验，不会感到乏味无趣

本小节介绍

- 图形绘制
- 文本显示
- 颜色控制
- 动画效果

Java包中部分类的层次关系图



8.4.1 图形绘制

画布的相关知识

- 若要绘制图形，必须具备两个要素

❖ 画布

- Swing 中任何 JComponent 类的子类都可以充当画布的角色，因此所有的 Swing 控件都可以作为画布

❖ 画笔(Graphics类)

- 所有 Swing 控件中都包含 paint 方法，它负责在需要的时候对控件进行绘制
- paint 方法语法

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

- 缺少了任何一个元素，图形的绘制都将无法进行

- 实际开发中，如果不是要自定义特定控件的外观，通常都是采用继承 JPanel 类并重写其 paint 方法的方式来实现画布

画笔的相关知识

- 画笔：java.awt.Graphics

- Graphics 类中提供了很多绘制简单二维图形的方法，开发人员使用这些方法就可以在画布上绘制指定的图形

常用的Graphics绘制方法

方法声明

```
public void clearRect(int x,int y,int width,int height)
public void drawLine(int x1,int y1,int x2,int y2)
public void fillRect(int x,int y,int width,int height)
public void drawRect(int x,int y,int width,int height)
public void drawRoundRect(int x,int y,int width,int height,int arcWidth,int arcHeight)
public void fillRoundRect(int x,int y,int width,int height,int arcWidth,int arcHeight)
public void draw3DRect(int x,int y,int width,int height,boolean raised)
public void fill3DRect(int x,int y,int width,int height,boolean raised)
public void drawOval(int x,int y,int width,int height)
public void fillOval(int x,int y,int width,int height)
public void drawArc(int x,int y,int width,int height,int startAngle,int arcAngle)
public void fillArc(int x,int y,int width,int height,int startAngle,int arcAngle)
public void drawPolyline(int[] xPoints,int[] yPoints,int nPoints)
public void drawPolygon(int[] xPoints,int[] yPoints,int nPoints)
public void fillPolygon(int[] xPoints,int[] yPoints,int nPoints)
```

获取Graphics对象

➤ Graphics 是一个抽象类，开发人员不能创建 Graphics 类对象

➤ 方法1:

- ❖ 使用 paint 接收的 Graphics 引用直接绘制，paint 方法由系统调用

➤ 方法2:

- ❖ 通过相应类的 getGraphics 方法获得
 - ☞ JComponent 继承自 Component 类，所有 Component 类的子类对象都具有 getGraphics 方法，但对于控件一般不应使用此方法获取画笔，而应重写 paint 方法
 - ☞ Image 类的子类也具有该方法，对于 Image 采用此方法获得的画笔一般在缓冲区进行绘制

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 19

坐标系

➤ Java 中绘制图形采用的是笛卡尔坐标系，该坐标系以像素为单位

➤ 画布左上角为该坐标系的原点 (0,0)，X轴向右延伸，Y轴向下延伸

➤ 在画布上定位图形是通过该图形的最左上角点的来进行的

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 20

java.awt.Color 颜色类

➤ Color 类使用 32 位表示颜色，其中 24 位表示颜色的 RGB 分量，每个分量用 8 位，剩下的 8 位用来表示颜色的 A (Alpha，透明度)

➤ Color 类的构造方法

构造方法	描述
public Color(float r,float g,float b)	每个颜色分量值范围在 0.0~1.0, alpha 的默认值为 1.0(不透明)
public Color(float r,float g,float b,float a)	alpha 值范围在 0.0~1.0
public Color(int r,int g,int b)	每个颜色分量值范围在 0~255, alpha 的默认值为 255 (不透明)
public Color(int r,int g,int b,int a)	每个量的值范围在 0~255

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 21

Color类的常用方法

方法	描述
public Color brighter()	获取比当前颜色亮的同色调颜色
public Color darker()	获取比当前颜色暗的同色调颜色
public getAlpha()	返回颜色的透明度分量值
public int getRed()	返回颜色的红色分量值
public int getGreen()	返回颜色的绿色分量值
public int getBlue()	返回颜色的蓝色分量值

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 22

Color类的常用颜色常量

常量名称	说明	常量名称	说明
Color.BLACK	黑色	Color.GREEN	绿色
Color.BLUE	蓝色	Color.LIGHT_GRAY	浅灰色
Color.CYAN	蓝绿色	Color.MAGENTA	紫红色
Color.DARK_GRAY	深灰色	Color.ORANGE	橙色
Color.GRAY	灰色	Color.PINK	桃红色
Color.RED	红色	Color.YELLOW	黄色
Color.WHITE	白色		

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 23

调配画笔颜色

➤ 调用画笔(Graphics)对象的 setColor 方法设置画笔的颜色
public void setColor(Color c)

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 24

8.4.2 文本显示

- ✿ 使用画笔也可以绘制指定内容的文本，并且可以指定字体、字号等

Graphics 类的绘制文本 drawString() 方法

> 语法:

```
public void drawString(String str,int x,int y)
```

- ✦ 参数 str 为指定要绘制的内容对应的字符串，而 x 与 y 分别表示该字符串最左上侧点在画布上的 x、y 坐标位置

字体的控制

- > 绘制文本的时候若总是使用默认的字體，将使得程序的界面非常死板，没有新意
- > Java 中提供对字体支持的类 `java.awt.Font`，通过该类可以使用系统中支持的各种字体进行文本显示
- > `Font` 类中封装了字体的 3 个属性：字体名称、样式以及字号(以磅为单位)
- > `Font` 类一共提供了 3 个构造器，最常用到的是使用 3 个属性作为参数的构造器，语法：

```
public Font(String name, int style, int size)
```

字体样式

> 字体样式常量有:

- ✦ `PLAIN`(普通字体)
- ✦ `BOLD`(粗体)
- ✦ `ITALIC`(斜体)

> 字体样式可以用位或运算符组合使用

- ✦ 如: `Font.BOLD | Font.ITALIC`

Font类的常用方法

方法	功能
<code>public boolean isPlain()</code>	判断 Font 对象的字体样式是否为 PLAIN
<code>public boolean isBold()</code>	判断 Font 对象的字体样式是否为 BOLD
<code>public boolean isItalic()</code>	判断 Font 对象的字体样式是否为 ITALIC
<code>public int getStyle()</code>	返回字体样式，可以是样式的位或组合
<code>public int getSize()</code>	返回字体的磅值大小，舍入为整数
<code>public String getFontName()</code>	返回字体的外观名称
<code>public String getName()</code>	返回字体的逻辑名称
<code>public String getFamily()</code>	返回字体的系列名称

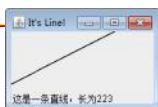
字体的控制（续）

> 设置画笔(Graphics)使用的字体

```
public void setFont(Font font)
```

例8_2 绘制直线和字符串

```
import java.awt.Graphics;
import javax.swing.*;
public class Line extends JFrame {
    public void paint(Graphics g) {
        super.paint(g); //必须调用, JFrame中用于防止图层透明
        g.drawLine(0, 100, 200, 0);
        g.drawString("这是一条直线, 长为222",
            (int)Math.sqrt(200*200+100*100), 10, 120);
    }
    public static void main(String args[]) {
        Line f = new Line();
        f.setTitle("It's Line!");
        f.setBounds(10, 10, 200, 130); //调整框体位置及大小
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时37分 31

例8_3 绘制矩形

```
import java.awt.*;
import javax.swing.*;
public class Rect extends JFrame {
    public void paint(Graphics g) {
        super.paint(g); //必须调用, 用于初始化
        g.setColor(Color.orange); //设置绘图颜色(左)
        g.drawRect(10, 40, 100, 60);
        g.fillRect(120, 40, 100, 60); //绘制并填充矩形(右)
    }
    public static void main(String args[]) {
        Rect f = new Rect();
        f.setTitle("It's Rect!");
        f.setBounds(10, 10, 230, 120);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时37分 32

例8_4 绘图综合示例

```
import java.awt.*;
import javax.swing.*;
public class aFace extends JFrame {
    public void paint(Graphics g) {
        super.paint(g); //必须调用, 用于初始化
        g.setColor(Color.blue); //设置绘图颜色
        g.drawOval(50, 50, 80, 100);
        g.fillOval(70, 80, 14, 7);
        g.fillOval(110, 80, 14, 7);
        g.drawArc(70, 110, 40, 20, 180, 180);
    }
    public static void main(String args[]) {
        aFace f = new aFace();
        f.setTitle("It's me!");
        f.setBounds(10, 10, 180, 180);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时37分 33

例：绘制曲线面板(自学)

```
import java.awt.Graphics;
import java.awt.Color;
import javax.swing.JPanel;

enum CoordinateOrigin {
    //坐标原点位置枚举常量
    HTOP_VLEFT, // X轴垂直居上, Y轴水平居左
    HTOP_VCENTER, // X轴垂直居上, Y轴水平居中
    HTOP_VRIGHT, // X轴垂直居上, Y轴水平居右
    HCENTER_VLEFT, // X轴垂直居中, Y轴水平居左
    HCENTER_VCENTER, // X轴垂直居中, Y轴水平居中
    HCENTER_VRIGHT, // X轴垂直居中, Y轴水平居右
    HBOTTOM_VLEFT, // X轴垂直居下, Y轴水平居左
    HBOTTOM_VCENTER, // X轴垂直居下, Y轴水平居中
    HBOTTOM_VRIGHT // X轴垂直居下, Y轴水平居右
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时37分 34

例：绘制曲线面板(续, 自学)

```
public class DrawPanel extends JPanel {
    int x0, y0; // 坐标原点值
    double[] x = null, y = null; // 绘制曲线的参数
    Color coordinateColor = Color.BLUE; // 坐标线的颜色, 默认为蓝色
    Color curveColor = Color.RED; // 曲线的颜色, 默认红色
    /* 坐标原点位置, 默认为在组件的中心位置 */
    CoordinateOrigin origin = CoordinateOrigin.HCENTER_VCENTER;
    double sx=1, sy=1; // 绘制数据的缩放系数

    public DrawPanel() { } // 构造函数
    // 设置坐标原点的位置
    public void setOrigin(CoordinateOrigin origin) { this.origin = origin; }
    // 取坐标原点位置
    public CoordinateOrigin getOrigin() { return origin; }
    // 设置水平方向的缩放系数
    public void setScaleX(double sx) { this.sx = sx; }
    // 设置垂直方向的缩放系数
    public void setScaleY(double sy) { this.sy = sy; }
    // 设置水平与垂直方向的缩放系数
    public void setScaleXY(double sx, double sy) { this.sx=sx; this.sy=sy; }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时37分 35

例：绘制曲线面板(续, 自学)

```
// 取水平方向的缩放系数
public double getScaleX() { return sx; }
// 取垂直方向的缩放系数
public double getScaleY() { return sy; }
// 设置绘制参数
public void setXY(double[] x, double[] y) { this.x = x; this.y = y; }
// 设置坐标线颜色
public void setCoordinateColor(Color color) { this.coordinateColor = color; }
// 取坐标线颜色
public Color getCoordinateColor() { return coordinateColor; }
// 设置曲线的颜色
public void setCurveColor(Color color) { this.curveColor = color; }
// 取曲线的颜色
public Color getCurveColor() { return curveColor; }
// 设置坐标线和曲线的颜色
public void setColor(Color coordinateColor, Color curveColor) {
    this.coordinateColor = coordinateColor;
    this.curveColor = curveColor;
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时37分 36

例：绘制曲线面板(续，自学)

```
protected void computeOrigin() { // 计算坐标原点
    switch(origin) {
    default:
    case HTOP_VLEFT: // X轴垂直居上, Y轴水平居左
        x0 = 0; y0 = 0; break;
    case HTOP_VCENTER: // X轴垂直居上, Y轴水平居中
        x0 = getWidth()/2-1; y0 = 0; break;
    case HTOP_VRIGHT: // X轴垂直居上, Y轴水平居右
        x0 = getWidth()-2; y0 = 0; break;
    case HCENTER_VLEFT: // X轴垂直居中, Y轴水平居左
        x0 = 0; y0 = getHeight()/2-1; break;
    case HCENTER_VCENTER: // X轴垂直居中, Y轴水平居中
        x0 = getWidth()/2-1; y0 = getHeight()/2-1; break;
    case HCENTER_VRIGHT: // X轴垂直居中, Y轴水平居右
        x0 = getWidth()-2; y0 = getHeight()/2-1; break;
    case HBOTTOM_VLEFT: // X轴垂直居下, Y轴水平居左
        x0 = 0; y0 = getHeight()-2; break;
    case HBOTTOM_VCENTER: // X轴垂直居下, Y轴水平居中
        x0 = getWidth()/2-1; y0 = getHeight()-2; break;
    case HBOTTOM_VRIGHT: // X轴垂直居下, Y轴水平居右
        x0 = getWidth()-2; y0 = getHeight()-2; break;
    }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 37

例：绘制曲线面板(续，自学)

```
// 绘制坐标
protected void drawXYCoordinate(Graphics g){
    g.setColor(coordinateColor);
    switch(origin){
    default:
    case HTOP_VLEFT: // X轴垂直居上, Y轴水平居左
        g.drawLine(1,1,getWidth()-2,1); // 绘制X轴
        g.drawLine(1,1,getHeight()-2); // 绘制Y轴
        break;
    case HTOP_VCENTER: // X轴垂直居上, Y轴水平居中
        g.drawLine(1,1,getWidth()-2,1); // 绘制X轴
        g.drawLine(getWidth()/2-1,getHeight()-2); // 绘制Y轴
        break;
    case HTOP_VRIGHT: // X轴垂直居上, Y轴水平居右
        g.drawLine(1,1,getWidth()-2,1); // 绘制X轴
        g.drawLine(getWidth()-2,getHeight()-2); // 绘制Y轴
        break;
    case HCENTER_VLEFT: // X轴垂直居中, Y轴水平居左
        g.drawLine(1,getHeight()/2-1,getWidth()-1,getHeight()/2-1); // 绘制X轴
        g.drawLine(1,1,getHeight()-2); // 绘制Y轴
        break;
    case HCENTER_VCENTER: // X轴垂直居中, Y轴水平居中
        g.drawLine(1,getHeight()/2-1,getWidth()-1,getHeight()/2-1); // 绘制X轴
        g.drawLine(1,1,getHeight()-2); // 绘制Y轴
        break;
    case HCENTER_VRIGHT: // X轴垂直居中, Y轴水平居右
        g.drawLine(1,getHeight()/2-1,getWidth()-1,getHeight()/2-1); // 绘制X轴
        g.drawLine(1,1,getHeight()-2); // 绘制Y轴
        break;
    case HBOTTOM_VLEFT: // X轴垂直居下, Y轴水平居左
        g.drawLine(1,getHeight()-2,getWidth()-1,getHeight()-2); // 绘制X轴
        g.drawLine(1,1,getHeight()-2); // 绘制Y轴
        break;
    case HBOTTOM_VCENTER: // X轴垂直居下, Y轴水平居中
        g.drawLine(1,getHeight()-2,getWidth()-1,getHeight()-2); // 绘制X轴
        g.drawLine(1,1,getHeight()-2); // 绘制Y轴
        break;
    case HBOTTOM_VRIGHT: // X轴垂直居下, Y轴水平居右
        g.drawLine(1,getHeight()-2,getWidth()-1,getHeight()-2); // 绘制X轴
        g.drawLine(1,1,getHeight()-2); // 绘制Y轴
        break;
    }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 38

例：绘制曲线面板(续，自学)

```
case HCENTER_VCENTER: // X轴垂直居中, Y轴水平居中
    g.drawLine(1,getHeight()/2-1,getWidth()-2,getHeight()/2-1); // 绘制X轴
    g.drawLine(getWidth()/2-1,1,getWidth()/2-1,getHeight()-2); // 绘制Y轴
    break;
case HCENTER_VRIGHT: // X轴垂直居中, Y轴水平居右
    g.drawLine(1,getHeight()/2-1,getWidth()-1,getHeight()/2-1); // 绘制X轴
    g.drawLine(getWidth()/2-1,1,getWidth()-2,getHeight()-2); // 绘制Y轴
    break;
case HBOTTOM_VLEFT: // X轴垂直居下, Y轴水平居左
    g.drawLine(1,getHeight()-2,getWidth()-2,getHeight()-2); // 绘制X轴
    g.drawLine(1,1,1,getHeight()-2); // 绘制Y轴
    break;
case HBOTTOM_VCENTER: // X轴垂直居下, Y轴水平居中
    g.drawLine(1,getHeight()-2,getWidth()-2,getHeight()-2); // 绘制X轴
    g.drawLine(getWidth()/2-1,1,getWidth()-2,getHeight()-2); // 绘制Y轴
    break;
case HBOTTOM_VRIGHT: // X轴垂直居下, Y轴水平居右
    g.drawLine(1,getHeight()-2,getWidth()-2,getHeight()-2); // 绘制X轴
    g.drawLine(getWidth()-2,1,getWidth()-2,getHeight()-2); // 绘制Y轴
    break;
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 39

例：绘制曲线面板(续，自学)

```
// 绘制曲线
protected void drawCurve(Graphics g) {
    if(x==null || y==null) return; //如果没有提供绘制数据则不能绘制
    if(x.length==0 || y.length==0) return; //如果没有提供绘制数据则不能绘制
    int len = x.length>y.length?y.length:x.length; //按最少的元素个数绘制
    /* 因程序中需要根据坐标原点位置修正绘制值, 因此需要将实参复制过来, 防止修改实参 */
    int[] dx = new int[len];
    int[] dy = new int[len];
    for(int i=0; i<len; i++) { //复制并修正绘制数据在界面中的位置
        dx[i] = (int)(x[i]*sx) + x0;
        dy[i] = (int)(y[i]*sy) + y0;
    }
    g.setColor(curveColor);
    g.drawPolyline(dx,dy,len); //用参数绘制折线
}

public void paint(Graphics g) {
    super.paint(g);
    computeOrigin();
    drawXYCoordinate(g);
    drawCurve(g);
}
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 40

例：绘制曲线面板测试程序(自学)

```
import javax.swing.JFrame;
import java.awt.BorderLayout;

public class DrawCurveDemo{
    public static void main(String[] args) {
        //准备曲线数据
        double[] x = new double[200];
        double[] y = new double[200];
        for(int i=0; i<200; i++) {
            x[i] = -4+1*i*0.1;
            y[i] = 2*x[i]*x[i]*x[i]-4*x[i]*x[i]+3*x[i]-6; //y=2x^3-4x^2+3x-6
        }
        JFrame f = new JFrame("绘制曲线");
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        DrawPanel d = new DrawPanel();
        //将坐标原点设置在界面的中间位置
        d.setOrigin(CoordinateOrigin.HCENTER_VCENTER);
        d.setXY(x,y); //设置绘制参数
        d.setScaleXY(10,1); //绘制时横坐标放大10倍, 纵坐标不变
        f.add(d, BorderLayout.CENTER); f.setSize(300,350);
        f.setVisible(true);
    }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 41

**8.4.3 动画效果**

- 动画的实现原理
 - 让程序根据一定的规则不断地对画布进行重新绘制
- 动画实现策略
 - 将绘制规则编码到方法 `paint` 中, 定时地让 `paint` 方法重新绘制
- 重新绘制
 - 调用画布的 `repaint()` 方法请求系统执行 `paint` 方法进行重绘, 而不能直接调用 `paint` 方法重绘

repaint方法声明	功能
<code>public void repaint()</code>	重绘整个画布
<code>public void repaint(int x,int y,int width,int height)</code>	重绘指定的矩形区域

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 42

例：用线程实现滑块的左右移动(自学)

```
import java.awt.Color;
import java.awt.Canvas;
import java.awt.Graphics;
import java.awt.BorderLayout; // 演示用
import javax.swing.JFrame; // 演示用
import javax.swing.JPanel; // 演示用

public class SimpleAnimation extends Canvas implements Runnable {
    private Thread thread = null; // 控制滑块动作的线程
    private boolean toRight = true; // true:自左向右移动滑块 false:自右向左移动滑块
    private int x = 0; // 滑块左上角的初始坐标
    private Color backColor = Color.BLUE; // 滑块移动轨迹的背景颜色, 初始为蓝色
    private Color foreColor = Color.YELLOW; // 滑块颜色, 初始为黄色
    private int delayTime = 50; // 滑块移动的延迟时间, 初始为100ms
    private int sliderWidth = 30; // 滑块的宽度和高度
    private boolean moving = false; // 滑块的初始状态为停止滑动

    public SimpleAnimation() {
        this.setBackground(backColor);
    }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 43

例：用线程实现滑块的左右移动(续, 自学)

```
// 背景色处理函数
public void setBackColor(Color backColor){
    if(moving) this.stop();
    this.backColor = backColor;
    setBackground(this.backColor);
    if(moving) this.start();
}

public Color getBackColor() { return this.backColor; }

// 滑块颜色处理函数
public void setForeColor(Color foreColor) {
    if(moving) this.stop();
    this.foreColor = foreColor;
    if(moving) this.start();
}

public Color getForeColor() { return this.foreColor; }

// 延迟时间处理函数
public void setDelayTime(int delayTime){
    if(moving) this.stop();
    this.delayTime = delayTime;
    if(moving) this.start();
}
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 44

例：用线程实现滑块的左右移动(续, 自学)

```
public int getDelayTime() { return this.delayTime; }

// 滑块的宽度和高度处理函数
public void setSliderWidth(int sliderWidth) {
    if(moving) this.stop();
    this.sliderWidth = sliderWidth;
    if(moving) this.start();
}

public int getSliderWidth() { return this.sliderWidth; }

// 滑块滑动的启动与停止
public void start() {
    if(moving) return; // 如果正在滑动则返回
    moving = true;
    thread = new Thread(this); // 否则创建线程并启动
    thread.start();
}

public void stop() {
    if(!moving) return; // 如果已停止滑动则返回
    moving = false;
    thread.stop(); // 否则停止滑动并销毁线程
    thread = null;
}
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 45

例：用线程实现滑块的左右移动(续, 自学)

```
// 线程体
public void run() {
    while(moving) {
        for(int i=0; i<getWidth()-sliderWidth; i++){
            try { thread.sleep(delayTime); }
            catch (InterruptedException e) { }
            if(toRight) x = i; // 设置滑块新位置, 从左向右移动
            else x = getWidth()-sliderWidth-i; // 设置滑块新位置, 从右向左移动
            repaint(); // 在新位置上重新绘制滑块
        }
        toRight = !toRight;
    }
}

public void paint(Graphics g) {
    super.paint(g);
    g.setColor(foreColor);
    g.fillRect(x, 0, sliderWidth, getHeight());
}
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 46

例：用线程实现滑块的左右移动(续, 自学)

```
public static void main(String[] args) {
    JFrame f = new JFrame("滑块动画演示程序");
    SimpleAnimation canvas = new SimpleAnimation();
    f.add(canvas, BorderLayout.CENTER);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setSize(300,80);
    f.setVisible(true);
    canvas.start();
}
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 47

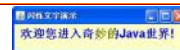
例：闪烁文字(自学)

```
import java.awt.Font;
import java.awt.Canvas;
import java.awt.Color;
import java.awt.Graphics;

import java.awt.BorderLayout; // 测试
import javax.swing.JFrame; // 测试

public class BlinkWords extends Canvas implements Runnable {
    private String words = "欢迎您进入奇妙的Java世界!"; // 闪烁的文字串
    private Font font = new Font("TimesRoman", Font.BOLD, 20); // 文字串字体
    private Thread thread = null;
    private boolean toRight = true; // 方向
    private int fps = 24; // 每秒帧速
    private int x = 0; // 闪烁的坐标
    private int x0 = 20, y0 = 20; // 文字在画布中显示的左上角坐标
    private Color backColor = new Color(255,255,200); // 淡黄色背景

    public BlinkWords() { setBackground(backColor); }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 48

例：闪烁文字(续，自学)

```

public void setWords(String words) { this.words = words; }
public String getWords() { return this.words; }
public void setFont(Font font) { this.font = font; }
public Font getFont() { return this.font; }
public void setFps(int fps) { this.fps = fps; }
public int getFps() { return this.fps; }
public void setBackColor(Color backColor) { this.backColor = backColor; }
public Color getBackColor() { return backColor; }
public void setX(int x) { this.x0 = x; }
public int getX() { return this.x0; }
public void setY(int y) { this.y0 = y; }
public int getY() { return this.y0; }
public void setXY(int x, int y) { this.x0 = x, this.y0 = y; }

```

**例：闪烁文字(续，自学)**

```

public void start() {
    if (thread == null) {
        thread = new Thread(this);
        thread.start();
    }
}

public void stop() {
    if (thread != null) {
        thread.stop();
        thread = null;
    }
}

```

**例：闪烁文字(续，自学)**

```

public void run() {
    long time;
    int delayTime = 1000/fps; //延迟时间
    time = System.currentTimeMillis(); //开始绘制动画的时间
    while(true) {
        for(int i=0; i<getWidth(); i=i+5) {
            if(toRight) x = i + x0;
            else x = getWidth()-i + x0;
            time = time+delayTime; //计算延迟时间,使满足帧速率为fps
            try {
                Thread.sleep(Math.max(0, time-System.currentTimeMillis()));
            } catch (InterruptedException e) {}
            repaint();
        }
        toRight = !toRight; // 反向
    }
}

```

**例：闪烁文字(续，自学)**

```

public void paint(Graphics g) {
    g.setFont(font); // 设置字体
    g.setColor(Color.BLUE); // 设置当前颜色为蓝色
    g.drawString(words, x0, y0);

    g.setColor(new Color((float)x/getWidth(),
        ((float)(getWidth()-x)/getWidth(),0));
    g.clipRect(x,y0,30,getHeight()); // 设置剪切区,使以下的工作仅在此区间内完成
    g.drawString(words, x0, y0); // 在相同的位置上用最新设置的当前颜色写字符串
    // 注意: 写出的字符串仅显示在剪切区内部的部分
}

public static void main(String[] args) {
    JFrame f = new JFrame("闪烁文字演示");
    BlinkWords blink = new BlinkWords();
    f.add(blink, BorderLayout.CENTER);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setSize(300,80);
    f.setVisible(true);
    blink.start();
}

```

**用 Timer 类实现动画**

➤既可以用线程，也可以用 javax.swing.Timer 定时器实现定时重绘

➤Timer构造方法

public Timer(int delay, ActionListener listener)

- ❖ delay: 指定初始延迟和动作事件间延迟的毫秒数
- ❖ listener: 指定初始的监听器，可以为 null

➤注意：

- ❖ 定时器一旦启动，就会按指定的时间间隔触发动作事件 (ActionEvent 类对象) 而调用注册到 ActionListener 监听器中的 actionPerformed 方法，也就是说，对定时器而言，actionPerformed 方法表示要执行的任务，应该把任务代码(这里是启动重绘 repaint 方法)编写在 actionPerformed 方法中

Timer类的常用方法

方法声明	功能
public boolean isRunning()	判断定时器是否正在运行
public void start()	启动定时器
public void stop()	停止定时器
public void restart()	重新启动定时器
public void setDelay(int delay)	设置定时器的时间间隔
public void setInitialDelay(int initDelay)	设置定时器的初始延迟
public int getInitialDelay()	返回定时器的初始延迟
public void addActionListener(ActionListener listener)	为定时器对象注册指定的动作事件监听器
public void removeActionListener(ActionListener listener)	从定时器对象中注销指定的动作事件监听器

8.5 对话框与面板

● 组件简介

- 在图形用户界面设计中，窗口系统中所显示的各种对象都统称为“控件”或“组件”，组件分为：
 - ❖ **基本组件**——不可再分割的组件
 - ❖ **容器组件**——用来放置其它组件的组件



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 55

容器(Container)

- 容器是用来放置其它组件的组件，故称为**容器组件**
- 用户可以把各种组件放到容器组件中，甚至把一个容器组件放到另一个容器组件中
- 主要的容器类：
 - ❖ AWT: Window、Frame、Dialog、Panel、Applet
 - ❖ Swing: JWindow、JFrame、JDialog、JPanel、JApplet

容器	名称	功能
JWindow	窗口类	Window 类的实例没有边框和标题,并且不能附加或嵌入到另一个容器中
JFrame	框架类	Frame 类的实例可以有一个菜单条,否则它与 Window 类的对象非常相似
JDialog	对话框类	Dialog 类的对象只有在有一个相关的 Frame 类对象存在时才存在
JPanel	面板类	Panel 类的是实例提供一个可以加入组件的容器,是容器组件的通用容器

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 56

顶级容器

- 建立一个窗口系统，首先需要创建一个图形用户界面，这就需要**使用顶级容器类 JWindow、JFrame、JDialog、JApplet 等**
 - ❖ 顶级容器不能作为组件放入到其它容器中
- JWindow 类提供了用于窗口操作的基本功能，但通常使用的是 JWindow 类的子类 **JFrame (框架)类**和 **JDialog (对话框)类**
- 所有的 Swing 组件 **必须**被添加到容器中，才能被显示出来
- javax.swing.JComponent 类是 Container 类的子类
 - ❖ Container 类用来管理一组相关的组件

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 57

8.5.1 JFrame窗体

- 在 Swing 中，JFrame 组件不是用纯 Java 语言编写的，是一个**重量级组件**
- 重量级组件的含义：
 - 组件在创建的时候，都会有一个相应的本地计算机中的组件在为其工作
- 利用 JFrame 类可以创建包含**标题、边框**以及**最大化、最小化、关闭按钮**的窗口

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 58

JFrame窗体（续）

- JFrame 类构造器创建的窗体是**不可见的**，需要在代码中编码设置才可以在屏幕上看见窗体
 - ❖ 这样做的目的是因为很多对窗体的特殊操作都需要在窗体被设置为可见之前完成，否则将会在运行时抛出异常
- JFrame 类构造器创建的窗体默认的尺寸为 0×0 像素，默认的位置坐标为 [0, 0]
 - ❖ 在开发中要注意，仅仅将窗体设置为可见的还是不能看到窗体，还需要**显式地**指定窗体的具体尺寸

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 59

JFrame窗体（续）

- 注意：
 - ❖ 在创建了 Swing 窗体后，不应直接把组件添加到创建的窗体中
 - ☞ Swing 窗体有一个称为内容窗格(ContentPane)的容器，组件应添加到窗体对应的内容窗格中
 - ❖ JFrame 框架默认的布局方式是 BorderLayout，它将容器划分为东西南北中
 - ❖ 获取窗体内容窗格的方法


```
Container content = [窗体对象].getContentPane();
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 60

JFrame类的主要方法

方法声明	功能
<code>public void setSize(int width, int height)</code>	设置窗体的宽度和高度
<code>public void setVisible(boolean b)</code>	设置窗体的可见性
<code>public void setBounds(int x, int y, int width, int height)</code>	设置窗体在屏幕中出现的位置
<code>public void setDefaultCloseOperation()</code>	设置控制窗体关闭时的行为, 常量有: EXIT_ON_CLOSE: 关闭并退出程序 DISPOSE_ON_CLOSE: 关闭并释放 JFrame对象, 程序继续运行 DO_NOTHING_ON_CLOSE: 没有动作 HIDE_ON_CLOSE: 隐藏并继续运行
<code>public Container getContentPane()</code>	返回窗体的 contentPane 对象
<code>public void setTitle(String title)</code>	设置框架的标题
<code>public void pack()</code>	调整窗口正好容纳各组件
<code>public void setLocation(int x, int y)</code>	将组件移到新位置

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 61

Container类的主要方法

方法	功能
<code>Component add(Component comp)</code>	将指定组件追加到此容器的尾部
<code>Component add(Component comp, int index)</code>	将指定组件添加到此容器的给定位置上
<code>void add(Component comp, Object constraints)</code>	使用指定约束, 将指定的组件添加到此容器的尾部
<code>void add(Component comp, Object constraints, int index)</code>	使用指定约束, 将指定组件添加到此容器的指定索引所在的位置上
<code>void remove(Component comp)</code>	从此容器中移除指定组件
<code>void remove(int index)</code>	从此容器中移除 index 指定的组件。
<code>void removeAll()</code>	从此容器中移除所有组件

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 62

例8_5 JFrame使用举例

```
import java.awt.*;
import javax.swing.*;
public class JFrameDemo {
    public static void main(String args[]) {
        JFrame f = new JFrame("简单框架"); //创建框架
        Container c = f.getContentPane(); //获取内容面板
        c.setLayout(new FlowLayout());
        JButton btn = new JButton("Button1"); //创建一个按钮
        c.add(btn); //将按钮加入面板中
        f.setSize(160, 100); //修改框架尺寸
        f.setVisible(true); //显示框架
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



新版本的 JDK, 直接在 JFrame 中可以调用 add (组件对象) 来添加组件, 组件将被添加到内容格中

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 63

例8_6 新版本的JFrame使用举例

```
import java.awt.*;
import javax.swing.*;
public class JFrameDemo {
    public static void main(String args[]) {
        JFrame f = new JFrame("简单框架"); //创建框架
        Container c = f.getContentPane(); //获取内容面板
        f.setLayout(new FlowLayout());
        JButton btn = new JButton("Button1"); //创建一个按钮
        f.add(btn); //将按钮加入面板中
        f.setSize(160, 100); //修改框架尺寸
        f.setVisible(true); //显示框架
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 64

8.5.2 JDialog对话框

- ✱ JDialog 是 Swing 中提供的用来实现自定义对话框的类, 也属于顶层容器, 同时也是重量级容器
 - 有标题条而无菜单条和最小化按钮图标
- ✱ JDialog 是 Dialog 类的子类
- ✱ JDialog 与 JFrame 的区别
 - JDialog 对象需要依赖于其他容器(如 JFrame)的存在而存在, 当它所依赖的容器关闭或最小化时, 该对话框也随之关闭或最小化, 当它所依赖的容器还原时, 该对话框也随之还原
- ✱ JDialog 默认的布局是 BorderLayout
 - 调用 show() 方法显示才可见, 调用 hide() 方法可将其隐藏(注意: 这两个方法已过时)

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 65

两种模式的对话框

- 模式对话框(响应模式):
 - ✱ 对话框出现期间, 它所依附窗口不接收任何操作
 - ✱ 即: 只让程序响应对话框的内部事件, 而对对话框以外的事件不作响应
 - ✱ 如: “关于”对话框
- 非模式(无模式)对话框(非响应模式):
 - ✱ 对话框出现时, 与所依附窗口都可同时接收操作
 - ✱ 即: 程序可以响应对话框以外的事件
 - ✱ 如: 查找对话框
- 对话框通常在应用程序中弹出, 用于提示输入数据、保存文件等

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 66

JDialog的主要构造方法

方法声明	作用
JDialog()	创建一个没有标题并且没有指定 Frame 所有者的无模式对话框
JDialog(Frame owner)	创建一个没有标题但将指定的 Frame 作为其所有者的无模式对话框
JDialog(Frame owner, boolean modal)	创建一个具有指定所有者 Frame、模式和空标题的对话框
JDialog(Frame owner, String title)	创建一个具有指定标题和指定所有者窗体的无模式对话框， 初始状态不可见
JDialog(Frame owner, String title, boolean modal)	创建一个具有指定标题、所有者 Frame 和模式的对话框， 初始状态不可见

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 67

JDialog的主要方法

方法声明	作用
public String getTitle()	获取对话框的标题内容
public void setTitle(String s)	设置对话框的标题内容
public void setModal(boolean b)	设置对话框的模式
public void setSize(int width,int height)	设置对话框的大小
public setVisible(boolean b)	设置对话框的可见性


网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 68

例8_7 无模式对话框

```
import java.awt.*;
import javax.swing.*;
class DiaDemo extends JDialog {
    private JButton JB1 = new JButton("按钮1");
    private JButton JB2 = new JButton("按钮2");
    private JButton JB3 = new JButton("按钮3");
    private JButton JB4 = new JButton("按钮4");
    private JButton JB5 = new JButton("按钮5");

    public DiaDemo(Frame owner) {
        super(owner);
        this.add(JB1, BorderLayout.CENTER);
        this.add(JB2, BorderLayout.EAST);
        this.add(JB3, BorderLayout.WEST);
        this.add(JB4, BorderLayout.NORTH);
        this.add(JB5, BorderLayout.SOUTH);
        this.setSize(300, 200);
        this.setVisible(true);
    }
}

public class DialogDemo1 {
    public static void main(String arg[]){
        new DiaDemo(new JFrame());
    }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 69

标准对话框：JOptionPane——确认对话框

```
static int showConfirmDialog(Component parentComponent, Object message)
static int showConfirmDialog(Component parentComponent, Object message,
    String title, int optionType)
static int showConfirmDialog(Component parentComponent, Object message,
    String title, int optionType, int messageType)
static int showConfirmDialog(Component parentComponent, Object message,
    String title, int optionType, int messageType, Icon icon)
```

- ❖ parentComponent: 对话框依附的Frame; 如果为 null 或者是 Frame 类型, 则使用默认的 Frame
- ❖ message: 要显示的 Object
- ❖ title: 对话框的标题字符串
- ❖ optionType: 指定选项: YES_NO_OPTION、YES_NO_CANCEL_OPTION 或 OK_CANCEL_OPTION
- ❖ messageType: 指定外观图标, 包括: ERROR_MESSAGE、INFORMATION_MESSAGE、WARNING_MESSAGE、QUESTION_MESSAGE 或 PLAIN_MESSAGE
- ❖ 返回: YES_OPTION、NO_OPTION、CANCEL_OPTION

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 70

标准对话框：JOptionPane——消息对话框

```
static void showMessageDialog(Component parentComponent,
    Object message)
static void showMessageDialog(Component parentComponent,
    Object message, String title, int messageType)
static void showMessageDialog(Component parentComponent,
    Object message, String title, int messageType, Icon icon)
```

- ❖ parentComponent: 对话框依附的 Frame; 如果为 null 或者是 Frame 类型, 则使用默认的 Frame
- ❖ message: 要显示的 Object
- ❖ title: 对话框的标题字符串
- ❖ messageType: 显示的消息类型, 包括 ERROR_MESSAGE、INFORMATION_MESSAGE、WARNING_MESSAGE、QUESTION_MESSAGE 或 PLAIN_MESSAGE

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 71

标准对话框：JOptionPane——输入对话框

```
static String showInputDialog(Component parentComponent,
    Object message)
static String showInputDialog(Component parentComponent,
    Object message, Object initialSelectionValue)
static String showInputDialog(Component parentComponent,
    Object message, String title, int messageType)
static Object showInputDialog(Component parentComponent,
    Object message, String title, int messageType, Icon icon,
    Object[] selectionValues, Object initialSelectionValue)
static String showInputDialog(Object message)
static String showInputDialog(Object message, Object initialSelectionValue)
```

- ❖ parentComponent: 对话框的父 Component
- ❖ message: 要显示的 Object
- ❖ title: 要在对话框的标题栏中显示的 String
- ❖ messageType: 要显示的消息类型, 包括: ERROR_MESSAGE、INFORMATION_MESSAGE、WARNING_MESSAGE、QUESTION_MESSAGE 或 PLAIN_MESSAGE

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 72

其他对话框

➤ javax.swing.JColorChooser

- ❖ 颜色选择器

➤ javax.swing.JFileChooser

- ❖ 文件选择器
- ❖ 打开文件: `showOpenDialog()`
- ❖ 存储文件: `showSaveDialog()`



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 73

8.5.3 JPanel面板

- ✿ 面板是常用的非顶层容器之一，可以将其他控件放在面板中以组织一个子界面
- ✿ 在 Swing 中，可以使用 `javax.swing.JPanel` 类创建面板，它继承自 `javax.swing.JComponent` 类
- ✿ JPanel 缺省使用顺序布局 `FlowLayout`
- ✿ 如果创建具有双缓冲和流布局的 JPanel 对象，可以使用不带任何参数的构造函数

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 74

常用的JPanel方法

方法声明	描述
<code>public Component add(Component c)</code>	将指定组件加入面板中
<code>public int getHeight()</code>	返回面板的当前高度
<code>public getWidth()</code>	返回面板的当前宽度
<code>public void setToolTipText(String text)</code>	注册要在工具提示中显示的文本text，当鼠标停留在该面板上时显示该文本

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 75

例8_8 面板的创建与使用

```
import java.awt.*;
import javax.swing.JFrame;
import javax.swing.*;
public class UsePanel extends JFrame {
    public UsePanel() {
        setLayout(new FlowLayout());
        JPanel panel1 = new JPanel(); // 创建一个面板对象
        JPanel panel2 = new JPanel();
        add(panel1); add(panel2);
        panel1.setBackground(Color.lightGray); // 为能看清面板修改面板的背景颜色
        panel2.setBackground(Color.yellow);
        panel1.add(new JButton("left"));
        panel1.add(new JButton("right"));
        panel2.add(new JButton("Panel2"));
        pack(); // 根据所放组件，调整窗口正好容纳各组件
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
    public static void main(String args[]) {
        UsePanel frm = new UsePanel();
    }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 76

8.6 事件处理

- ✿ 对于 GUI 的应用程序来说，事件处理是必不可少的，用户与程序之间的交互都是通过事件处理来实现的
- ✿ 当用户与 GUI 交互时，无论是单击鼠标，还是按下键盘，都会触发相应的事件。事件将通知应用程序发生的情况，接着应用程序会根据不同的事件作出相应的回应

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 77

8.6.1 委托事件处理模型

- ✿ 概述
 - 事件是描述“发生了什么事情”的对象，是系统对可能处于某种状态或操作的预定义
 - 用户是通过响应各种事件来和程序交互的

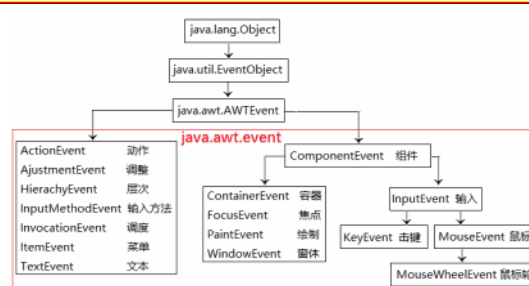
网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 78

事件处理由三个组成部分

- 1) 事件源(Event Source): 事件产生者或来源
 - ❖ 如, 单击了登录按钮, 则登录按钮为事件源
 - ❖ 事件源通常是程序中某个可交互的组件, 也可以是定时器等其它对象
 - 2) 事件对象(Event Object): 事件源产生的事件通常由用户的操作触发, 每个事件均被 Java 运行时环境封装为事件对象
 - ❖ 事件对象包含与该事件相关的必要信息, 如鼠标按下事件产生时鼠标所处的坐标等
 - 3) 事件监听器(Event Listener): 用于接收和处理事件的对象
 - ❖ 那些用于处理事件的代码所在的类的对象就是事件监听器
- 事件源提供了注册和注销事件监听器的方法

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 79

java.awt.event包中提供的主要事件类



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 80

java.awt.Event

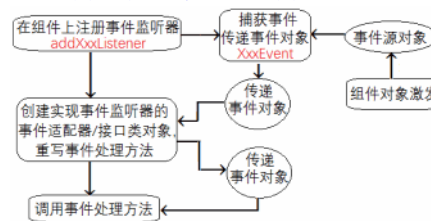
➢ java.awt.Event 类对象的主要状态信息

事件状态	含义
int id	事件类型: 按键或单击鼠标等的事件
Object target	发生事件的对象
long when	时间戳, 指出事件发生时的时间
int x	发生事件时的坐标
int y	
int key	在键盘事件中被按下的键的代码
Object arg	一个与事件相关的参数。例如: 在组件上显示的字符串
int modifiers	修饰键的状态(即Alt, Ctrl键的状态)

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 81

Java2的委托事件处理模型思想

- 每个事件源可以发出若干种不同类型的事件。一个事件对象可以被多个事件监听处理器对象处理, 反过来, 一个事件监听器对象也能处理多个事件对象



委托事件模型的实质是将用户界面和事件处理分离

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 82

一般处理事件的流程

- 1) 编写一个事件监听器类
 - ❖ 方法1(实现监听器接口): implements XxxListener
 - ❖ 方法2(继承监听适配器类): extends XxxAdapter
 - ☞ 其中 Xxx 是事件的名字
- 2) 在事件监听器类中重写处理事件的方法, 编写处理代码
- 3) 调用组件的 addXxxListener 方法注册组件对象的事件监听器类对象
 - ❖ 注册监听器: addXxxListener(监听器对象)
 - 注销监听器: removeXxxListener(监听器对象)

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 83

事件适配器类和事件监听器接口

- 1) 它们所包含的方法相同
 - ❖ 适配器类: 实现了全部方法, 但方法体为空, 为完成特定的功能需覆盖对应的方法
 - ❖ 监听器接口: 全部为抽象方法, 需要重写全部方法, 为特定功能编写处理代码, 其余的方法可用空方法体实现
- 2) 所有的事件都有对应的监听器接口, 但部分事件没有对应的适配器类
- 3) 通常使用监听器接口实现事件处理, 很少用适配器类

➢ 对应关系

事件类	监听器接口	适配器	注册处理器
XxxEvent	XxxListener	XxxAdapter	addXxxListener(Object)
KeyEvent	KeyListener	KeyAdapter	addKeyListener(Object)

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 84

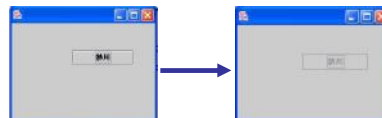
常见事件、接口及适用组件

事件、接口	适用组件
ActionEvent ActionListener	JButton, JList, JtextField, JMenuitem, JCheckBoxMenuItem, JMenu, JRadioButtonMenuItem
AdjustmentEvent AdjustmentListener	JScrollbar, 实现Adjustable接口的子类
ComponentEvent ComponentListener	Component及其子类, 包括JButton, JCheckBox, JComboBox, Container, JPanel, JApplet, JScrollPane, Window, JDialog, JFileDialog, JFrame, JLabel, JList, JScrollbar, JTextArea, JtextField
FocusEvent FocusListener	Component及其子类
KeyEvent KeyListener	Component及其子类
MouseEvent (点击及滚轮) MouseListener (点击接口) MouseEvent (点击及滚轮) MouseMotionListener (滚轮接口)	Component及其子类

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 85

8.6.2 灵活设计事件监听器类

- 方法1: 让窗体本身来完成这个监听器
- 方法2: 用内置类/匿名的内置类来实现
- 方法3: 用外部类来实现, 但需传入待处理的窗体做处理



- 注意: 组件上可以注册多个监听器类

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 86

8.7 常用的组件

- Java Swing 提供了 20 多种不同的用户组件, 所有的 Swing 组件都是从 `javax.swing.JComponent` 类派生而来, JComponent 类继承自 `java.awt.Container` 类
- Java Swing 为每一种组件都提供了事件处理机制, 有相应的事件监听器查看组件上是否发生了事件
- 通过事件处理机制, 所编写的程序都可以捕获用户对任何组件的操作, 同时予以响应

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 87



常用组件的一般使用步骤

- 1) 创建组件
 - 2) 调用容器对象的 `add()` 方法将组件加入到容器中
 - 3) 调用组件的各种方法设置组件的属性
 - 4) 调用组件的 `addXxxListener()` 注册组件事件监听器
 - 5) 编写组件的事件监听器程序
- 除第一步外, 其它步骤没有特别的顺序规定

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 89

AbstractButton抽象类

- AbstractButton 类是常见按钮 JButton、JCheckBox 和 JRadioButton 等的基类

AbstractButton 的主要方法

主要方法	描述
boolean isSelected()	检查按钮是否被选中
void setSelected(boolean b)	设置按钮的被选状态
String getText()	返回按钮的标签文本串
void setText(String text)	设置按钮的标签文本串
Icon getIcon()	返回按钮的图标
void setIcon(Icon icon)	设置按钮的图标
Icon getDisabledIcon()	返回按钮禁用时显示的图标
void setDisabledIcon(Icon icon)	设置按钮被禁用时显示的图标
Icon getPressedIcon()	返回按钮被按下时显示的图标
void setPressedIcon(Icon icon)	设置按钮被按下时显示的图标
void setMnemonic(int mnemonic)	设置当前模型上的键盘助记符

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 90

8.7.1 javax.swing.JButton按钮

✿ GUI 应用程序中，按钮是与用户交互使用得最多的控件之一，很多功能都是通过用户按下按钮来触发代码完成的

✿ javax.swing.JButton 类是最简单的按钮类型

✿ 常用事件：java.awt.event.ActionEvent

✿ JButton 创建的按钮组件既可以使用字符串，也可以使用图标，还可以是两者作为按钮的标签显示出来

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 91

JButton类的主要方法

构造方法	作用
JButton()	创建一个空白标签按钮
JButton(Icon icon)	创建一个有图像标签的按钮
JButton(String text)	创建一个有文字标签的按钮
JButton(String text,Icon icon)	创建一个有图像和文字标签的按钮

➤ 例8_9：按钮被禁用、启用、快捷键、显示组件提示信息、改变框架底色TestJButton.java

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 92

例8_9：按钮的综合示例TestJButton.java

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

class JButtonFrame extends JFrame implements ActionListener {
    private JButton b1, b2, b3;
    private JPanel panel;
    private Container container;
    public JButtonFrame() {
        setTitle("JButton应用");        setSize(350,200);
        try { //系统外观
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) { }
        container = getContentPane();

        b1 = new JButton("禁用(D)");
        b1.setFont(new Font(Font.SANS_SERIF, Font.PLAIN, 14));
        b1.setMnemonic(KeyEvent.VK_D); // Alt+D快捷键

        b2 = new JButton("设置框架底色(S)");
        b2.setFont(new Font(Font.MONOSPACED, Font.PLAIN, 14));
        b2.setMnemonic(KeyEvent.VK_S); // Alt+S快捷键
    }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 93

例8_9：按钮的综合示例TestJButton.java

```
b3 = new JButton("启用(E)");
b3.setFont(new Font(Font.SERIF, Font.PLAIN, 14));
b3.setMnemonic(KeyEvent.VK_E); // Alt+E快捷键
b3.setEnabled(false);

b1.addActionListener(this);
b3.addActionListener(this);
b2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) { //事件处理方法
        int red = (int)(Math.random() * 256);
        int green = (int)(Math.random() * 256);
        int blue = (int)(Math.random() * 256);
        container.setBackground(new Color(red, green, blue));
    }
});

//设置3个按钮的提示信息
b1.setToolTipText("禁用设置框架底色按钮");
b2.setToolTipText("设置框架底色");
b3.setToolTipText("启用设置框架底色按钮");
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 94

例8_9：按钮的综合示例TestJButton.java

```
panel = new JPanel();
panel.add(b1); panel.add(b2); panel.add(b3);
container.add(panel, BorderLayout.NORTH);

setVisible(true);

public void actionPerformed(ActionEvent e) {
    if((JButton)e.getSource() == b1) { //b1按钮
        //或 if("禁用(D)".equals(e.getActionCommand())) {
        b2.setEnabled(false);
        b1.setEnabled(false);
        b3.setEnabled(true);
    }
    else { //b3按钮
        b2.setEnabled(true);
        b1.setEnabled(true);
        b3.setEnabled(false);
    }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 95

例8_9：按钮的综合示例TestJButton.java

```
public class TestJButton {
    public static void main(String[] args) {
        JButtonFrame frame = new JButtonFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 96

小结

➤ 要在容器中创建组件，以按钮为例，需要如下步骤：

- ❖ 1) 定义一个按钮对象
- ❖ 2) 用 **new** 实例化按钮对象
- ❖ 3) 用 **add** 方法将按钮加入容器中
 - ☞ 容器就是窗口、框架、对话框、面板及 Applet 等
 - ☞ 创建其它组件的步骤同样如此
- ❖ 4) 对顶层容器，最后需要调整尺寸，并显示：
 - ☞ 调整尺寸：
 - ◆ setSize(int, int)
 - ◆ setLocation(int x, int y)
 - ◆ setBounds(int x, int y, int width, int height)
 - ☞ 显示：
 - ◆ setVisible(true)

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 97

8.7.2 文本组件

❖ 1 javax.swing.JLabel 标签类(静态文本组件)

- 一种不响应事件、非交互(不能编辑)的组件，主要用于显示一些说明性描述信息
- 没有修饰，界面中看不到该组件的边界
- 不仅可以显示文本，还可以显示图标
- 标签在容器中的对齐方式：
 - ❖ 左对齐： javax.swing.SwingConstants.LEFT
 - ❖ 居中： javax.swing.SwingConstants.CENTER
 - ❖ 右对齐： javax.swing.SwingConstants.RIGHT
 - ❖ 使用从左到右和从右到左的语言的文本开始边： javax.swing.SwingConstants.LEADING
 - ❖ 使用从左到右和从右到左的语言的文本结束边： javax.swing.SwingConstants.TRAILING

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 98

javax.swing.JLabel构造方法

构造方法	功能描述
JLabel()	创建空白标签
JLabel(Icon icon)	创建含指定图像的标签，icon 的默认排列方式为 javax.swing.SwingConstants.CENTER (水平居中)
JLabel(Icon icon, int horizontalAlignment)	创建含指定图像的标签，并按指定方式水平排列图像
JLabel(String text)	创建含文字串的标签，文字串的默认排列方式为 javax.swing.SwingConstants.LEFT (水平居左)
JLabel(String text, int horizontalAlignment)	创建含文字串的标签，并按指定方式水平排列文字串
JLabel(String text, Icon icon, int horizontalAlignment)	创建含文字串和图像的标签，并按指定方式水平排列文字串

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 99

javax.swing.JLabel主要方法

主要方法	功能描述
String getText()	返回标签的文本内容
void setText(String text)	设置标签的文本内容
Icon getIcon()	返回标签的图标
void setIcon(Icon icon)	设置标签的图标
int getHorizontalAlignment()	返回标签中内容的水平对齐方式
void setHorizontalAlignment(int alignment)	设置标签中内容的水平对齐方式
int getVerticalAlignment()	返回标签中内容的垂直对齐方式
void setVerticalAlignment(int alignment)	设置标签中内容的垂直对齐方式
void setBackground(Color c)	设置标签的背景色
void setForeground(Color c)	设置标签的前景色
void setFont(Font f)	设置标签中文字内容的字体

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 100

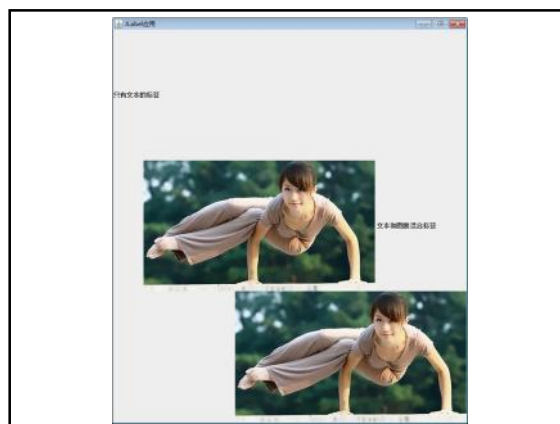
例8_10 标签应用举例TestJLabel.java

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

class JLabelFrame extends JFrame {
    JLabel label1, label2, label3;
    public JLabelFrame() {
        setTitle("JLabel应用");
        label1 = new JLabel("只有文本的标签", SwingConstants.LEFT);
        label2 = new JLabel("文本和图像混合标签",
            new ImageIcon("picture.jpg"), SwingConstants.CENTER);
        label3 = new JLabel(new ImageIcon("picture.jpg"), SwingConstants.RIGHT);
        getContentPane().setLayout(new GridLayout(3, 1));
        getContentPane().add(label1);
        getContentPane().add(label2);
        getContentPane().add(label3);
        setSize(700, 800);
        setVisible(true);
        setResizable(false);
    }
}

public class TestJLabel {
    public static void main(String[] args) {
        new JLabelFrame().setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 101



文字输入组件

➤在 Swing 中与文字输入有关的组件有：

- ❖ **JTextField** 该组件为单行文本编辑器
- ❖ **JPasswordField** 该组件为密码框编辑器
- ❖ **TextArea** 该组件为多行文本编辑器
- ❖ **EditorPane** 该组件可显示多种文件格式
- ❖ **TextPane** 该组件可设置文件各种样式
- ❖ **ScrollPane** 当组件内容大于显示区域时会自动产生滚动轴

网络安全与网络工程系蔡东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 103

2 javax.swing.JTextField与javax.swing.JPasswordField

➤两者都只能用于输入单行文本，如果文本的长度超出了控件可以显示的范围，其会自动滚动文本

➤JPasswordField 会把用户输入的字符显示为"*"(默认)，也可以设置其它字符来回显密码

➤对于 JTextField，所有的剪切、复制、粘贴及其快捷键的操作都可以自动实现

网络安全与网络工程系蔡东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 104

JTextField的构造方法

构造方法	描述
JTextField()	创建一个空的文本框
JTextField(Document doc, String text, int cols)	使用指定的存储模式 doc 创建文本框，并设置初始化为 text，字段长度为 cols
JTextField(int cols)	创建文本框，并设置初始化为 cols
JTextField(String text)	创建文本框，并设置初始化为 text
JTextField(String text, int cols)	创建文本框，并设置初始化为 text，字段长度为 cols

网络安全与网络工程系蔡东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 105

JTextField的主要方法

主要方法	描述
void setText(String str)	将文本框中的文本设置为 str
String getText()	获取文本框中输入的文本
boolean isEditable()	检测文本框中的文本是否可编辑
void setEditable(boolean b)	设置文本框中的文本可编辑性
int getColumns()	返回文本框所显示的字符列数
void setColumns(int col)	设置文本框能显示的字符列数

网络安全与网络工程系蔡东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 106

JPasswordField的构造方法

构造方法	描述
JPasswordField()	创建一个空的密码框
JPasswordField(Document doc, String text, int cols)	使用指定的存储模式 doc 创建密码框，并设置初始化为 text，字段长度为 cols
JPasswordField(int cols)	创建密码框，并设置初始化为 cols
JPasswordField(String text)	创建密码框，并设置初始化为 text
JPasswordField(String text, int cols)	创建密码框，并设置初始化为 text，字段长度为 cols

网络安全与网络工程系蔡东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 107

javax.swing.JTextField与javax.swing.JPasswordField常用事件

➤JTextField 和 JPasswordField 在用户输入内容并按回车键时，产生 java.awt.event.ActionEvent 事件

网络安全与网络工程系蔡东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 108

例8_11 用户登录界面TestTextField.java

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

class JPanelClass extends JPanel implements ActionListener {
    JLabel nameLabel, passwordLabel;
    JTextField name;
    JPasswordField password;
    JButton okButton, cancelButton;

    public JPanelClass() {
        nameLabel = new JLabel("姓名:");
        passwordLabel = new JLabel("密码:");
        name = new JTextField(20);
        password = new JPasswordField(20);
        okButton = new JButton("确定");
        okButton.addActionListener(this);
        cancelButton = new JButton("取消");
        add(nameLabel); add(name);
        add(passwordLabel); add(password);
        add(okButton); add(cancelButton);
    }
}
```



网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 109

例8_11 用户登录界面TestTextField.java

```
public void actionPerformed(ActionEvent e) {
    String nameStr = name.getText();
    String passwordStr = new String(password.getPassword()); //getText()返回String, getPassword()返回char[]
    JOptionPane.showMessageDialog(null, nameStr + "\n" + passwordStr, "用户登录信息显示", JOptionPane.INFORMATION_MESSAGE);
}

class JFrameClass extends JFrame {
    JPanelClass panel;
    public static final int DEFAULT_WIDTH = 300;
    public static final int DEFAULT_HEIGHT = 120;

    public JFrameClass() {
        setSize(DEFAULT_WIDTH, DEFAULT_HEIGHT);
        setTitle("用户登录界面");
        panel = new JPanelClass();
        getContentPane().add(panel);
        setVisible(true);
        setResizable(false);
    }
}
```



网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 110

例8_11 用户登录界面TestTextField.java

```
public class TestTextField {
    public static void main(String[] args) {
        new JFrameClass().setDefaultCloseOperation(
            JFrame.EXIT_ON_CLOSE);
    }
}
```



网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 111

3 javax.swing.JTextArea文本区类

➤ **JTextArea** 类是 **Swing** 中提供的用单一字体和格式显示和输入多行文本的组件

➤ 默认情况下不会自动换行，但可以通过设置让其自动换行

➤ **JTextArea** 是以跨平台的方式处理换行符，根据不同的操作系统平台，文本文件中的行分隔符可以是换行符、回车或者二者的组合

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 112

JTextArea类的构造方法

构造方法	作用
JTextArea()	创建一个新的文本区
JTextArea(Document doc)	使用指定文本存储模式 doc 创建文本区
JTextArea(Document doc, String text, int row, int col)	使用指定文本存储模式 doc 及初始行数 row 和列数 col 的文本区，默认字符串为 text
JTextArea(int row, int col)	创建初始 row 行和 col 列的文本区
JTextArea(String text)	创建初始文本为 text 的文本区
JTextArea(String text, int row, int col)	创建初始文本为 text 及 row 行和 col 列的文本区

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 113

JTextArea类的主要方法

主要方法	作用
void setText(String str)	设置文本内容为 str，并清除原有全部文本内容
String getText()	获取文本区的内容，以字符串返回
void insert(String str, int x)	在指定位置 x 插入指定文本串 str
void replaceRange(String str, int begin, int end)	将文本区中从 begin 位置起到 end 位置止的文本子串用 str 替换
void append(String str)	将字符串追加到文本区中文本的尾部
void setRows(int rows)	设置文本区的行数
void setColumns(int cols)	设置文本区的列数
void setLineWrap(boolean b)	打开或关闭换行
void setWrapStyleWord(boolean b)	参数为 true 时超长部分自动转到下一行，否则超长部分将被截断

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 114

JTextArea类的常用事件

- 当输入回车时产生 `java.awt.event.ActionEvent` 事件
- 当文本内容发生变化时产生 `java.awt.event.TextEvent` 事件

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 115

javax.swing.JTextArea的滚动处理

- JTextArea 本身没有滚动功能
 - ❖ 如果需要滚动，可以将文本区和带滚动视图的容器面板 `javax.swing.JScrollPane` 配合使用，实现滚动显示的效果
- 例8_12 在上方文本域中输入文本信息，当键入回车键后，将其中的文本信息追加到下方的文本区中，文本区具有滚动功能 `TestJTextArea.java`

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 116

例8_12 具有滚动功能的JTextArea(TestJTextArea.java)

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

class JTextAreaFrame extends JFrame implements ActionListener {
    JTextField textField;
    JTextArea textArea;
    JScrollPane scrollPane;

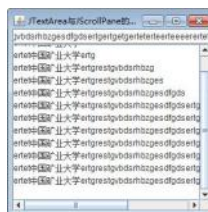
    public JTextAreaFrame() {
        setTitle("JTextArea与JScrollPane的使用");
        setSize(300,300);
        textField = new JTextField(20);
        textArea = new JTextArea(10,20);
        //设置滚动功能,不能完整显示时出现滚动条
        scrollPane = new JScrollPane(textArea);
        //将组件放到容器中
        getContentPane().add(textField,BorderLayout.NORTH);
        getContentPane().add(scrollPane,BorderLayout.CENTER);
        textField.addActionListener(this);
        textArea.setEditable(false); //文本区不可编辑
        setVisible(true);
    }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 117

例8_12 具有滚动功能的JTextArea(TestJTextArea.java)

```
public void actionPerformed(ActionEvent e) { //回车时将输入内容放入多行文本框
    String text = textField.getText();
    textArea.append(text+"\n");
}

public class TestJTextArea {
    public static void main(String[] args) {
        new JTextAreaFrame().setDefaultCloseOperation(
            JFrame.EXIT_ON_CLOSE);
    }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 118

4 其他文本输入组件

- `javax.swing.JEditorPane`类：增加支持纯文本(text/plain)、HTML 和 RTF (text/rtf)的文本编辑功能

构造方法	说明
<code>JEditorPane ()</code>	建立一个新的展示文件内容的窗口界面
<code>JEditorPane (String url)</code>	以详细的 URL 字符串为基础，创建一个展示 HTML文 件内容的窗口界面
<code>JEditorPane (String type, String text)</code>	建立一个按指定文件格式类型显示文本
<code>JEditorPane (URL initialPane)</code>	以详细的 URL 字符串当作输入值来创建一个展示 HTML 文件内容的窗口界面

- `java.swing.JTextPane`类：进一步扩展了 `JEditorPane` 的功能，允许文本中嵌入图像或其他组件

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 119

8.7.3 树类javax.swing.JTree

- ✿ `JTree` 类创建的实例可实现一个树状分层结构显示数据的视图
- ✿ 所有的节点都是从根节点引出
- ✿ 除根节点外，其他节点都具有两类：
 - 1) 带子节点的**分枝节点**
 - 2) 不带子节点的节点，称为**叶节点**

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 120

Tree类的常用方法

方法名称	方法说明
JTree()	返回带有示例模型的 JTree
JTree(Object[] value)	返回 JTree, 指定数组的每个元素作为不被显示的新根节点的子节点
JTree(TreeModel newModel)	返回 JTree 的一个实例, 它显示根节点 - 使用指定的数据模型创建树
JTree(TreeNode root)	返回 JTree, 指定的 TreeNode 作为其根, 它显示根节点
JTree(TreeNode root, boolean asksAllowsChildren)	返回 JTree, 指定的 TreeNode 作为其根, 它用指定的方式显示根节点, 并确定节点是否为叶节点
addTreeExpansionListener(TreeExpansionListener tel)	为 TreeExpansion 事件添加侦听器
addTreeSelectionListener(TreeSelectionListener tsl)	为 TreeSelection 事件添加侦听器
addTreeWillExpandListener(TreeWillExpandListener tel)	为 TreeWillExpand 事件添加侦听器

与树的类:

- ❖ 可变树节点: `javax.swing.tree.DefaultMutableTreeNode`
- ❖ 渲染: `javax.swing.tree.DefaultTreeCellRenderer`

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 121

例8_13 树(不讲)

➤ 例8_13 树(有关文件: BaseFrame.java、Hardware.java、HardwareTreeCellRenderer.java、ImageFactory.java、JTreeDemo.java、data.txt、cpu.png、hardDisk.png、hardware.png、ram.png、vcard.png)



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 122

8.7.4 javax.swing.JRadioButton单选按钮

❖ 单选按钮是用于在按钮组中, 每一时刻只允许选择一项且仅一项的情形

❖ 单选按钮需要成组出现, 且在每一组中, 每一时刻仅有一个选项被选中

❖ JRadioButton 的成组需要用 `javax.swing.ButtonGroup` 协作才能完成

❖ 成组过程:

- 将一组互斥访问的 JRadioButton 对象加入到 ButtonGroup 对象中

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 123

JRadioButton的构造方法

构造方法	描述
JRadioButton()	创建单选按钮
JRadioButton(Icon icon)	创建含图标但不含文字的单选按钮
JRadioButton(Icon icon,boolean selected)	创建具有是否选中状态且有图标无文字的单选按钮
JRadioButton(String text)	创建含文字的单选按钮
JRadioButton(String text,boolean selected)	创建含指定文字和是否选中状态的单选按钮
JRadioButton(String text,Icon icon)	创建含文字和图标的单选按钮
JRadioButton(String text,Icon icon,boolean selected)	创建含文字和图标且有是否选中状态的单选按钮
JRadioButton(Action a)	创建由a确定属性的单选按钮

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 124

JRadioButton常用事件

➤ `java.awt.event.ActionEvent`

➤ 可以使用成员方法 `boolean isSelected()` 判断是否被选中

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 125

8.7.5 javax.swing.JCheckBox复选框

❖ 复选框组件是指同时可以选择多项的选择组件

❖ 由于在操作时, 通过单击复选框组件在“选中”与“不选中”之间进行切换, 所以, 也可以将此组件归结为一种特殊的按钮, 称为复选按钮

❖ 复选框既可以显示字符串, 也可显示图像

❖ JCheckBox不需要编组使用, 各个选项之间没有逻辑约束关系

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 126

JCheckBox构造方法

构造方法	描述
JCheckBox()	创建无图标无文字串没有被选中的复选框
JCheckBox(Icon icon)	创建有图标的复选框
JCheckBox(Icon icon,boolean selected)	创建有图标及指定状态的复选框
JCheckBox(String text)	创建有文字串的复选框
JCheckBox(String text, boolean selected)	创建有文字串及指定状态的复选框
JCheckBox(String text, Icon icon)	创建有文字串和图标及指定状态的复选框
JCheckBox(String text, Icon icon, boolean selected)	创建有文字串及图标, 及指定状态的复选框

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 127

JCheckBox常用事件

➤ java.awt.ActionEvent

➤ 可以使用成员方法 **boolean isSelected()** 判断是否被选中

➤ 例8_14 用红、绿、蓝3种颜色中的一种及不同的字体组合显示文字串 TestJRadioButton.java

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 128

例8_14 用红、绿、蓝3种颜色中的一种及不同的字体组合显示式文字串

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

class JRadioButtonFrame extends JFrame implements ActionListener {
    JPanel fontPanel, colorPanel; //字体面板和颜色面板分别放置字体和颜色按钮
    JCheckBox bold, italic; //字体检查框按钮
    JLabel strLabel; //显示字符串标签
    JRadioButton red, green, blue; //颜色单选按钮
    ButtonGroup group; //单选按钮管理组件

    public JRadioButtonFrame() {
        setTitle("JRadioButton应用实例");
        setSize(300,150);
        fontPanel = new JPanel();
        bold = new JCheckBox("Bold");
        italic = new JCheckBox("Italic");
        fontPanel.add(bold);
        fontPanel.add(italic);

        strLabel = new JLabel("This is a statement.", SwingConstants.CENTER);
        strLabel.setFont(new Font("Serif", Font.PLAIN, 24));
        strLabel.setForeground(Color.RED);
    }
}
```



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时37分 129

例8_14 用红、绿、蓝3种颜色中的一种及不同的字体组合显示式文字串

```
colorPanel = new JPanel();
group = new ButtonGroup();
red = new JRadioButton("Red", true);
green = new JRadioButton("Green");
blue = new JRadioButton("Blue");
group.add(red); //进行成组管理
group.add(green);
group.add(blue);
colorPanel.add(red);
colorPanel.add(green);
colorPanel.add(blue);

getContentPane().add(strLabel, BorderLayout.NORTH);
getContentPane().add(fontPanel, BorderLayout.CENTER);
getContentPane().add(colorPanel, BorderLayout.SOUTH);

bold.addActionListener(this);
italic.addActionListener(this);
red.addActionListener(this);
green.addActionListener(this);
blue.addActionListener(this);

setVisible(true);
}
```



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 130

例8_14 用红、绿、蓝3种颜色中的一种及不同的字体组合显示式文字串

```
public void actionPerformed(ActionEvent e) {
    int mode = 0;
    if(bold.isSelected()) mode += Font.BOLD;
    if(italic.isSelected()) mode += Font.ITALIC;
    if(red.isSelected()) strLabel.setForeground(Color.RED);
    if(green.isSelected()) strLabel.setForeground(Color.GREEN);
    if(blue.isSelected()) strLabel.setForeground(Color.BLUE);

    strLabel.setFont(new Font("Serif", mode, 24));
}

public class TestJRadioButton {
    public static void main(String[] args) {
        new JRadioButtonFrame().setDefaultCloseOperation(
            JFrame.EXIT_ON_CLOSE);
    }
}
```



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 131

单选按钮、复选框小结

➤ JRadioButton 和 JCheckBox 除了与 JButton 一样都会触发 java.awt.event.ActionEvent 动作事件外, 还会触发 java.awt.event.ItemEvent 事件

➤ ItemEvent 事件的两点注意

- ❖ 1) ItemEvent 事件只在按钮的状态发生变化时才会触发
 - ☞ 例如, 从选中到未选中, 或者从未选中到选中都会触发 ItemEvent 事件
- ❖ 2) 只有向 JRadioButton 和 JCheckBox 注册了实现 ItemListener 监听接口的监听器, 当事件被触发时, 才会执行监听器当中的事件处理方法

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 132

8.7.6 javax.swing.JComboBox下拉列表

- ✿ 下拉列表用来提供一系列的选项，用户可以在下拉列表提供的选项中，进行**单选**
- ✿ 下拉列表有文本框和列表两部分组成，被选项在列表中显示，选中项在文本框中显示
- ✿ 下拉列表中的每一个选项称为一个**列表项**，它是以菜单的形式显示的，如果列表项太多，一次显示不完整，下拉列表将会出现滚动条
- ✿ 用户选择下拉列表中的某一选项，列表自动消失，被选项出现在文本框中
- ✿ 下拉列表中的列表项有统一的编号，编号从**0**开始

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 133

JComboBox构造方法

构造方法	描述
JComboBox()	创建一个无选项的下拉列表
JComboBox(ComboBoxModel model)	使用 model 创建下拉列表
JComboBox(Object[] items)	创建以数组项作选项的下拉列表
JComboBox(Vector items)	创建以向量项作选项的下拉列表

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 134

JComboBox主要方法

主要方法	描述
void setEditable(boolean b)	设置下拉列表的文本框是否可编辑
void addItem(Object item)	增加一个选项
void insertItemAt(Object item,int index)	在指定位置插入一个选项
void removeItem(Object item)	删除指定的选项
void removeItemAt(int index)	删除指定位置的选项
void removeAllItems()	删除全部选项
Object getSelectedItem()	返回当前的被选项
Object getItemAt(int index)	返回指定索引处的列表项
int getItemCount()	返回列表中的项数
int getSelectedIndex()	返回列表中与给定项匹配的项的索引
boolean isEditable()	如果 JComboBox 可编辑，则返回 true
void setEnabled(boolean b)	启用组合框以便可以选择项
void insertItemAt(Object anObject, int index)	在项列表中的给定索引处插入项

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 135

javax.swing.JComboBox主要事件

- java.awt.event.ItemEvent：对列表选项的拖动、单击将产生此事件，主要用于取得用户的项目

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 136

例8_15 下拉列表TestJComboBox.java

```
import java.awt.event.*;
import java.awt.*;

class JComboBoxFrame extends JFrame implements ItemListener {
    JComboBox comboBox;
    JLabel label;

    public JComboBoxFrame() {
        setTitle("JComboBox应用实例");
        setSize(200,200);

        comboBox = new JComboBox();
        comboBox.addItem("Choice1");
        comboBox.addItem("Choice2");
        comboBox.addItem("Choice3");
        comboBox.addItem("Choice4");

        comboBox.addItemListener(this);
        comboBox.setSelectedIndex(-1); //未选

        label = new JLabel("你选择的是:", JLabel.LEFT);
        label.setFont(new Font("Serif", Font.PLAIN, 20));

        getContentPane().add(label, BorderLayout.SOUTH);
        getContentPane().add(comboBox, BorderLayout.NORTH);

        setVisible(true);
    }
}
```

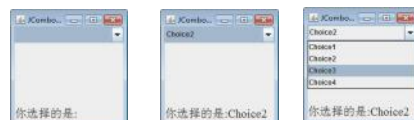


网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 137

例8_15 下拉列表TestJComboBox.java

```
public void itemStateChanged(ItemEvent e) {
    String str = (String)comboBox.getSelectedItem();
    if(str != null) label.setText("你选择的是:" + str);
}

public class TestJComboBox {
    public static void main(String[] args) {
        new JComboBoxFrame().setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 138

8.7.7 javax.swing.JList列表

❖ 列表一般有更多的表项，占据的空间也大，因此通常将其放置在滚动面板上，这样就可以提供一个文本数据项可卷动的列表

❖ 可以选择列表中的一个数据项或多个数据项

❖ 列表项编号也从 0 开始

❖ 列表中用户选中的选项以加色显示，而不是列在一个文本框中

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 139

JList类的构造方法

构造方法	说明
JList()	构造一个具有空的、只读模型的 JList
JList(ListModel dataModel)	根据指定的非 null 模型构造一个显示元素的 JList
JList(Object[] listData)	构造一个 JList，使其显示指定数组中的元素
JList(Vector<?> listData)	构造一个 JList，使其显示指定 Vector 中的元素

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 140

JList常用方法

方法	说明
void clearSelection()	清除选择；调用此方法后，isSelectionEmpty 将返回 true
int getSelectedIndex()	返回选中的列表项索引。若选中多项，则返回最小的索引值；若未任何项则返回-1
int[] getSelectedIndices()	返回所选的全部索引的数组（按升序排列）
Object[] getSelectedValues()	返回所有选择值的数组，根据其列表中的索引顺序按升序排序
Object getSelectedValue()	返回最小的选择单元索引的值；只选择了列表中单个项时，返回所选值
int getSelectionMode()	返回列表的当前选择模式
boolean isSelectionEmpty()	如果什么也没有选择，则返回 true；否则返回 false
void setSelectedIndex(int index)	选择单个单元
void setSelectedIndices(int[] indices)	将选择更改为给定数组所指定的索引的集合
void setSelectionMode(int selectionMode)	设置列表的选择模式

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 141

JList组件的三种选择模式

> 1) ListSelectionModel.SINGLE_SELECTION

❖ 每次只允许选择一个项目，此时，setSelectionInterval 和 addSelectionInterval 等效

> 2) ListSelectionModel.SINGLE_INTERVAL_SELECTION

❖ 允许连续选择多个项目。此时，如果给定间隔没有紧邻着现有选择或与现有选择重叠，则 addSelectionInterval 与 setSelectionInterval 相同(替换当前选择)，并可用于生成选择

> 3) ListSelectionModel.MULTIPLE_INTERVAL_SELECTION

❖ 允许间隔开地选择多个项目(默认)，不存在对选择的限制

> 多选的方法：

- ❖ Ctrl+鼠标单击：单个多选
- ❖ Shift +鼠标单击：连续多选



网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 142

例8_16 JList(TestJList.java)

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

public class TestJList extends JFrame {
    String items[] = {"arg0","arg1","arg2","arg3","arg4","arg5","arg6",
        "arg7","arg8","arg9"};
    JScrollPane jScrollPane1 = new JScrollPane(); //为列表选项可滚动
    JList jList1 = new JList(items);
    public TestJList() {
        try {
            jScrollPane1.setBounds(new Rectangle(5, 5, 120, 150));
            jScrollPane1.setViewportView(jList1);
            jList1.setSelectionMode(
                ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);
            getContentPane().setLayout(null);
            getContentPane().add(jScrollPane1, null);
            setSize(new Dimension(180, 230));
        } catch (Exception e) { e.printStackTrace(); }
    }

    public static void main(String[] args) {
        TestJList frame = new TestJList();
        frame.setDefaultCloseOperation(EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 143

8.7.8 javax.swing.JSlider滑块（不讲）

❖ 滑块用于在一个数据范围内选择数值

❖ 构造方法

构造方法	作用
JSlider()	创建水平滑块，默认刻度为 0~100，初始刻度为 50
JSlider(BoundedRangeModel m)	使用模式m创建滑块
JSlider(int orientation)	创建自定义方向的滑块，默认刻度为 0~100，初始刻度为 50
JSlider(int min,int max)	创建水平方向的滑块，自定义刻度范围，初始刻度为最小值加上最大值的平均值。
JSlider(int min,int max,int value)	创建水平方向的滑块，自定义刻度范围和初始刻度
JSlider(int orientation,int min, int max,int value)	创建指定方向的滑块，自定义刻度范围和初始刻度

网络安全与网络工程系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 144

JSlider主要方法

主要方法	描述
<code>void setPaintTicks(boolean b)</code>	设置是否显示滑块上的标尺
<code>void setMajorTickSpacing(int units)</code>	设置滑块标尺间的最大间距
<code>void setMinorTickSpacing(int units)</code>	设置滑块标尺间的最小间距
<code>void setPaintLabels(boolean b)</code>	设置是否显示滑块标尺上的标签

> 拖动滑块产生 `javax.swing.event.ChangeEvent` 事件

处理按钮事件的接口	<code>ChangeListener</code>
接口中处理事件的方法	<code>stateChanged(ChangeEvent e)</code>
事件源获得监视器的方法	<code>addChangeListener()</code>


网络工程与安全工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 145

例8_17 含有标志名称的滑块TestJSlider.java

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

class JSliderFrame extends JFrame implements ChangeListener {
    JSlider slider1, slider2, slider3;
    JLabel label1, label2, label3;
    JPanel panel1, panel2, panel3;
    Hashtable table;

    public JSliderFrame() {
        setTitle("JSlider应用举例");
    }
}
```



网络工程与安全工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 146

例8_17 含有标志名称的滑块TestJSlider.java

```
panel1 = new JPanel();
panel1.setLayout(new GridLayout(2, 1));
slider1 = new JSlider();
slider1.setPaintTicks(true); //显示标尺
slider1.setMajorTickSpacing(20); //设置主刻度标记的间隔
slider1.setMinorTickSpacing(10); //设置次刻度标记的间隔
slider1.setPaintLabels(true); //显示标尺上的标签
slider1.setPaintTrack(true); //绘制滑道
//滑块(及其表示值)为最靠近用户放置滑块处的刻度标记值
slider1.setSnapToTicks(true);
label1 = new JLabel("目前刻度:" + slider1.getValue());
panel1.add(label1);
panel1.add(slider1);
table = new Hashtable();
table.put(new Integer(0), new JLabel("低"));
table.put(new Integer(50), new JLabel("中"));
table.put(new Integer(100), new JLabel("高"));
slider1.setLabelTable(table); //指定将在给定值处绘制哪个标签
```



网络工程与安全工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 147

例8_17 含有标志名称的滑块TestJSlider.java

```
panel2 = new JPanel();
panel2.setLayout(new GridLayout(2, 1));
slider2 = new JSlider(JSlider.HORIZONTAL);
slider2.setMinimum(0); //设置滑块的最小值为0
slider2.setMaximum(100); //设置滑块的最大值为100
slider2.setValue(30); //设置滑块的当前值设置为30
slider2.setExtent(50); //设置滑块所“覆盖”的范围的大小
slider2.setPaintTicks(true);
slider2.setMajorTickSpacing(10);
slider2.setMinorTickSpacing(5);
slider2.setPaintLabels(true);
//向此组件添加任意的键/值“客户端属性”。提供对每个实例小型
//哈希表的访问。调用者可以使用 get/putClientProperty 为另一个
//模块所创建的组件添加注释, 本例未用
//slider2.putClientProperty("JSlider.isFilled",
// Boolean.TRUE);
label2 = new JLabel("目前刻度:" + slider2.getValue());
panel2.add(label2);
panel2.add(slider2);
```



网络工程与安全工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 148

例8_17 含有标志名称的滑块TestJSlider.java

```
table = new Hashtable();
table.put(new Integer(0), new JLabel("弱"));
table.put(new Integer(25), new JLabel("有点弱"));
table.put(new Integer(50), new JLabel("中"));
table.put(new Integer(75), new JLabel("有点强"));
table.put(new Integer(100), new JLabel("强"));
slider2.setLabelTable(table);
```



```
panel3 = new JPanel();
panel3.setLayout(new GridLayout(2, 1));
slider3 = new JSlider(20, 80);
slider3.setOrientation(JSlider.VERTICAL);
slider3.setPaintTicks(true);
slider3.setMajorTickSpacing(30);
slider3.setMinorTickSpacing(10);
slider3.setPaintLabels(true);
//slider3.putClientProperty("JSlider.isFilled", Boolean.TRUE); //本例未用
label3 = new JLabel("目前刻度:" + slider3.getValue());
panel3.add(label3);
panel3.add(slider3);
```

网络工程与安全工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 149

例8_17 含有标志名称的滑块TestJSlider.java

```
slider1.addChangeListener(this);
slider2.addChangeListener(this);
slider3.addChangeListener(this);

panel1.setPreferredSize(new Dimension(300, 130));
panel2.setPreferredSize(new Dimension(300, 130));
panel3.setPreferredSize(new Dimension(150, 260));

getContentPane().add(panel1, BorderLayout.NORTH);
getContentPane().add(panel2, BorderLayout.CENTER);
getContentPane().add(panel3, BorderLayout.SOUTH);

setSize(300, 500);
setVisible(true);

public void stateChanged(ChangeEvent e) {
    if((JSlider)e.getSource() == slider1)
        label1.setText("目前刻度:" + slider1.getValue());
    if((JSlider)e.getSource() == slider2)
        label2.setText("目前刻度:" + slider2.getValue());
    if((JSlider)e.getSource() == slider3)
        label3.setText("目前刻度:" + slider3.getValue());
}
```



网络工程与安全工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 150

例8_17 含有标志名称的滑块TestJSlider.java

```
public class TestJSlider {
    public static void main(String[] args) {
        new JSliderFrame().setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 151

8.7.9 javax.swing.JProgressBar进度条(不讲)

- ✿ 在安装新软件时经常见到进度条
- ✿ 进度条通常用于显示耗时较长的任务运行的进度
- ✿ 构造方法

构造方法	描述
JProgressBar()	创建进度条
JProgressBar(int min,int max)	按指定最小和最大值创建进度条
JProgressBar(int orient)	创建具有指定方向的进度条
JProgressBar(int orientation,int min,int max)	按指定方向、最小值和最大值创建进度条

- ✿ JProgressBar.VERTICAL
- ✿ JProgressBar.HORIZONTAL
- ✿ 进度条默认为水平方向

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 152

JProgressBar主要方法和事件**➤主要方法**

主要方法	描述
void setMinimum(int min)	设置进度条最小值
void setMaximum(int max)	设置进度条最大值
void setValue(int value)	设置进度条当前值
int getNumberOfFiles()	返回跟踪已复制的文件个数
void setStringPainted(boolean b)	设置进度条是否显示进度百分比

➤ 进度条值改变(setValue()方法)时触发
javax.swing.event.ChangeEvent事件

处理按钮事件的接口	ChangeListener
接口中处理事件的方法	stateChanged(ChangeEvent e)
事件源获得监视器的方法	addChangeListener()

➤ 例8_18 进度条结合使用定时器 TestJProgressBar.java

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 153

例8_18

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

```
class JProgressBarFrame extends JFrame implements ActionListener,
ChangeListener {
```

```
JProgressBar progressbar;
JLabel label;
Timer timer;
JButton b;
JPanel panel;
```

```
public JProgressBarFrame() {
    super("JProgressBar应用");
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 154

例8_18

```
label = new JLabel("", JLabel.CENTER);
progressbar = new JProgressBar();
progressbar.setOrientation(JProgressBar.HORIZONTAL);
progressbar.setMinimum(0);
progressbar.setMaximum(100);
progressbar.setValue(0);
progressbar.setStringPainted(true);
progressbar.addChangeListener(this);
progressbar.setPreferredSize(new Dimension(200, 30));
progressbar.setBorderPainted(false);
```



```
panel = new JPanel();
b = new JButton("Start");
b.addActionListener(this);
panel.add(b);
```

```
timer = new Timer(50, this);
```

```
getContentPane().add(panel, BorderLayout.NORTH);
getContentPane().add(progressbar, BorderLayout.CENTER);
getContentPane().add(label, BorderLayout.SOUTH);
```

```
setSize(300, 120);
setVisible(true);
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 155

例8_18

```
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == b) { timer.start(); }
    if (e.getSource() == timer) {
        int value = progressbar.getValue();
        if (value < 100) { value++; }
        else { timer.stop(); }
        progressbar.setValue(value);
        progressbar.setValue(0);
    }
}
```

```
public void stateChanged(ChangeEvent e1) {
    int value = progressbar.getValue();
    if (e1.getSource() == progressbar) {
        label.setText("当前已完成进度: " + Integer.toString(value) + "%");
    }
}
```

```
public class TestJProgressBar {
    public static void main(String[] args) {
        new JProgressBarFrame().setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 156

8.7.10 javax.swing.JScrollBar滚动条

滚动条在多种组件中出现过，例如在列表和多行文本框中，当显示内容超出了显示区域的范围时，这些组件的旁部就出现滚动条，用户可以通过拖动滚动条来看到所有要显示的内容

滚动条的另一种用途是可以作为一个滑块，用来选择指定的最小值和最大值之间的一个数值。也就是说，可以将滚动条看作一个取值范围，滑块左边界(或上边界)所在的位置给出滚动条的当前值

改变这个值的方法有三个：

- 1) 直接拖动滑块到所需位置
- 2) 单击滚动条两边的箭头使滑块沿箭头所指的方向移动(每次移动1个单位)
- 3) 用鼠标单击滑槽内的任意区域，可以使滑块沿相应方向移动(每次移动10个单位)

网络安全与网络工程系系平 jaxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 157

JScrollBar常用的方法

方法原型	含义
JScrollBar()	创建的滚动条的默认方向、初始值、滑块大小、最小值、最大值分别是垂直、0、10、0、100
JScrollBar(int orientation)	创建指定方向的滚动条，未指定的同上 滚动条方向：JScrollBar.HORIZONTAL、JScrollBar.VERTICAL
JScrollBar(int o, int value, int extent, int min, int max)	参数分别为方向、初始值、滑块大小、最小值、最大值
void setOrientation(int o)	设置滚动条方向
void setValue(int value)	设置滑块位置
void setVisibleAmount(int extent)	设置滑块大小
void setMinimum(int min)	设置滚动条最小值
void setMaximum(int max)	设置滚动条最大值
void setBlockIncrement(int i)	设置滑块增量大小(单击滑块两侧空白区域时)
void setUnitIncrement(int i)	设置滑块增量大小(单击两端的箭头时)
void setValuesAdjusting(Boolean b)	设置滑块位置是否正在改变。当开始拖动滑块时应设为 true，拖动停止后应设为 false，否则会产生连续产生多次调整事件

网络安全与网络工程系系平 jaxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 158

例8_19 滚动条的创建

```
import javax.swing.*;
public class TestJScrollBar {
    public static void main(String args[]) {
        JFrame frm=new JFrame("滚动条的创建");
        JScrollBar myVerScrollBar, myHorScrollBar;
        myVerScrollBar=new JScrollBar(JScrollBar.VERTICAL, 0, 10, 0, 100);
        myHorScrollBar=new JScrollBar();
        myHorScrollBar.setOrientation(JScrollBar.HORIZONTAL);
        myHorScrollBar.setValues(20, 20, 0, 100);
        frm.add("South",myHorScrollBar);
        frm.add("East",myVerScrollBar);
        frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frm.setSize(150,100);
        frm.setVisible(true);
    }
}
```



网络安全与网络工程系系平 jaxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 159

8.7.11 Timer定时器

定时器组件使程序在每隔一段时间就执行指定操作

- javax.swing.Timer
- java.util.Timer

构造方法

构造方法	描述
Timer(int delay, ActionListener listener)	创建指定时间间隔和监听器的组件

- 延迟时间单位为毫秒

主要方法

主要方法	描述
void start()	启动定时器定时
void stop()	停止定时器定时
void restart()	重新激活定时器定时
void setRepeats(boolean b)	设置定时器是否重复多次触发,参数为 false 时只会触发一次 ActionEvent 事件

Timer (续)

➢ 定时器每次延时到时触发java.awt.ActionEvent事件

处理事件的接口	ActionListener
接口中处理事件的方法	actionPerformed(ActionEvent e)
事件源获得监视器的方法	addActionListener()

➢ 例8_20：以拖动滑块设置延时 TestTimer.java

➢ 事实上使用 Timer 组件是在程序背后利用 Threads 在运行 Timer 的工作，因此也可以利用 Thread 的功能来自行制造出这样的效果

网络安全与网络工程系系平 jaxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 161

例8_20：以拖动滑块设置延时TestTimer.java

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



```
class TimerFrame extends JFrame implements ActionListener,ChangeListener {
    ImageIcon[] icons;
    JSlider slider1;
    JLabel label;
    JToggleButton toggleb1, toggleb2; //开关按钮
    JButton b;
    javax.swing.Timer timer; //java.util中也有Timer定时器
    int index = 0;

    public TimerFrame() {
        super("Timer应用");
        icons = new ImageIcon[5];
        for (int i=0; i<5; i++) icons[i] = new ImageIcon("Icons\\FILE" + (i + 1) + ".jpg");
    }
}
```



网络安全与网络工程系系平 jaxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 162

例8_20：以拖动滑块设置延时TestTimer.java

```

label = new JLabel(icons[0]);
JPanel panel1 = new JPanel();
panel1.setLayout(new GridLayout(2, 1));
slider1 = new JSlider();
slider1.setPaintTicks(true);
slider1.setMajorTickSpacing(20);
slider1.setMinorTickSpacing(10);
slider1.setPaintLabels(true);
slider1.addChangeListener(this);
panel1.add(slider1);

JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new GridLayout(1, 3));
toggleb1 = new JToggleButton("Start");
toggleb1.addActionListener(this);
buttonPanel.add(toggleb1);
b = new JButton("Restart");
b.addActionListener(this);
buttonPanel.add(b);
toggleb2 = new JToggleButton("Don't Repeat");
toggleb2.addActionListener(this);
buttonPanel.add(toggleb2);
panel1.add(buttonPanel);

```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 163

例8_20：以拖动滑块设置延时TestTimer.java

```

Hashtable<Integer,JLabel> table = new Hashtable<Integer,JLabel>();
table.put(new Integer(0), new JLabel("快"));
table.put(new Integer(50), new JLabel("中"));
table.put(new Integer(100), new JLabel("慢"));
slider1.setLabelTable(table);

timer = new javax.swing.Timer(slider1.getValue() * 10, this);

getContentPane().add(label, BorderLayout.CENTER);
getContentPane().add(panel1, BorderLayout.SOUTH);

setVisible(true);
}

```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 164

例8_20：以拖动滑块设置延时TestTimer.java

```

public void actionPerformed(ActionEvent e) { // 处理按钮事件与Timer事件
    if (e.getSource() == toggleb1) {
        // 当用户按下"start"按钮时, Timer开始运行, 且"Start"按钮会变成"Stop",若用户再次
        // 按下"stop"按钮, 则Timer暂停运行, 且"stop"按钮变成"start".
        if (e.getActionCommand().equals("Start")) {
            timer.start(); toggleb1.setText("Stop");
        }
        if (e.getActionCommand().equals("Stop")) {
            timer.stop(); toggleb1.setText("Start");
        }
    }
    // 当用户按下"Don't Repeat"按钮时, 则Timer事件只触发一次, 若再按一次"Don't
    // Repeat"按钮, 则Timer继续运行.
    if (e.getSource() == toggleb2) {
        if (timer.isRepeats()) { timer.setRepeats(false); }
        else { timer.setRepeats(true); timer.start(); }
    }
    // 当用户按下"Restart"按钮时, 则Timer组件的delay值恢复成初值, 并重新运行Timer.
    if (e.getSource() == b) { slider1.setValue(50); timer.restart(); }
    // 处理Timer产生的ActionEvent事件, 每次时间一到delay所设置的时间,label上的图片就会更新
    // 一次.
    if (e.getSource() == timer) {
        if (index == 5) index = 0;
        label.setIcon(icons[index]);
        label.repaint();
        index++;
    }
}

```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 165

例8_20：以拖动滑块设置延时TestTimer.java

```

// 处理slider所产生的ChangeEvent事件, 当用户移动slider1的滑动杆时, 等于从新设置Timer的
// delay时间.
public void stateChanged(ChangeEvent e1) {
    timer.setDelay(slider1.getValue() * 10);
}

public class TestTimer {
    public static void main(String[] args) {
        new TimerFrame().setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 166

8.7.12 菜单

- ✳ 菜单的功能通常可用按钮或其他用户界面组件完成
- ✳ 完整的菜单通常需要协同使用的组件
 - javax.swing.JMenuItem:
 - ✧ 用户表示菜单中的一个选项
 - javax.swing.JMenu:
 - ✧ 用于表示一个下拉菜单, 通常是 JMenuItem 的容器
 - javax.swing.JMenuBar:
 - ✧ 用户表示一组下拉菜单, 是 JMenu 的容器
- ✳ 菜单主要用于 JFrame 框架中
- ✳ 菜单项可以使用图标, 可以为菜单项指定助记符或快捷键, 还可以出现单选按钮以及复选框

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 167

在JFrame框架中使用菜单的步骤

- 声明并创建相应的 JMenuBar、JMenu 和 JMenuItem 组件对象
- 获得 JFrame 的内容格, 并向内容格中添加 JMenuBar 组件对象
- 向 JMenuBar 中添加下拉菜单 JMenu 对象
- 向 JMenu 中添加 JMenuItem 对象

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 168

JMenuBar方法

➤ javax.swing.JMenuBar构造方法

构造方法	描述
JMenuBar()	创建新的菜单栏

➤ javax.swing.JMenuBar主要方法

主要方法	描述
JMenu add(JMenu c)	将指定的菜单追加到菜单栏的末尾
JMenu getMenu(int index)	返回菜单栏中指定位置的菜单
boolean isSelected()	如果当前已选择了菜单栏的组件, 则返回 true
int getMenuCount()	返回菜单栏上的菜单数

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 169

JMenu构造方法

构造方法	描述
JMenu()	创建下拉菜单对象
JMenu(Action a)	创建支持 Action 的下拉菜单
JMenu(String s)	创建显示名为指定字符串的下拉菜单
JMenu(String s,boolean b)	创建指定名为指定字符串且能否下拉(b)的下拉菜单

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 170

JMenu的主要方法

主要方法	描述
void add(JMenuItem menuItem)	将某个菜单项追加到此菜单的末尾
void add(String s)	创建具有指定文本的新菜单项, 并将其追加到此菜单的末尾
void addSeparator()	将新分隔符追加到菜单的末尾
void insert(JMenuItem mi, int pos)	在给定位置插入指定的 JMenuItem
void insert(String s, int pos)	在给定位置插入具有指定文本的新菜单项
void insertSeparator(int index)	在指定的位置插入分隔符
void remove(int pos)	从此菜单移除指定索引处的菜单项
void remove(JMenuItem item)	从此菜单移除指定的菜单项
void removeAll()	从此菜单移除所有菜单项
setAccelerator(KeyStroke keyStroke)	为 JMenuItem 定义 setAccelerator
void JMenuItem getItem(int pos)	返回指定位置的 JMenuItem
int getItemCount()	返回菜单上的项数, 包括分隔符

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 171

JMenuItem方法

➤ JMenuItem构造方法

构造方法	描述
JMenuItem()	创建菜单项
JMenuItem(Action a)	创建支持 a 的菜单项
JMenuItem(Icon icon)	创建含图标的菜单项
JMenuItem(String text)	创建含指定字符串的菜单项
JMenuItem(String text,Icon icon)	创建含指定字符串和图标的菜单项
JMenuItem(String text, int mnemonic)	创建含指定字符串和快捷键的菜单项

❖ 快捷键用 java.awt.event.KeyEvent 中的常数指定

➤ JMenuItem的主要方法

主要方法	描述
void setEnabled(boolean b)	启用或禁用菜单项
boolean isEnabled()	返回菜单项是否可用

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 172

8.7.13 javax.swing.JToolBar工具栏

- 工具栏是提供迅速访问常用命令的一个按钮栏
- 工具栏可以拖动到需要的位置
- 构造方法

构造方法	描述
JToolBar()	创建工具栏, 默认为水平方向
JToolBar(int orientation)	创建指定方向的工具栏
JToolBar(String name)	创建指定名称的、水平方向的工具栏
JToolBar(String name,int orientation)	创建指定名称和方向的工具栏

- 使用 JToolBar 时一般都采用水平方向的位置, 因此构造时多采用上表中的第一种构造方式
- 也可以用 setOrientation() 方法来改变方向设置, 或用鼠标拖动来改变 JToolBar 的位置

- 例8_12: 菜单与工具栏的综合示例 TestJMenuAndJToolBar.java

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 173

例8_12: 菜单与工具栏的综合示例

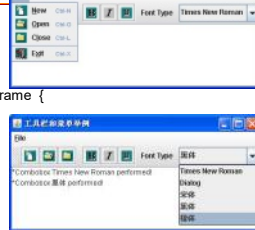
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

class JMenuAndJToolBarFrame extends JFrame {
    JTextArea theArea = null;
    static final String ComboStr[] = {
        "Times New Roman", "Dialog",
        "宋体", "黑体", "楷体"
    };

    public JMenuAndJToolBarFrame() {
        super("工具栏和菜单示例");

        theArea = new JTextArea();
        theArea.setEditable(false);
        getContentPane().add(new JScrollPane(theArea));
        JMenuBar MBar = new JMenuBar();
        MBar.setOpaque(true); // 设置为不透明的菜单

        JMenu mfile = buildFileMenu();
        JToolBar theBar = buildToolBar();
        this.getContentPane().add(theBar, BorderLayout.NORTH); // 将构造好的 JToolBar 放置
    }
}
```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 174

例8_12：菜单与工具栏的综合示例TestJMenuAndJToolBar.java

```

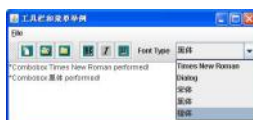
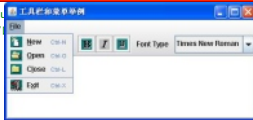
MBar.add(mfile); //将JMenu加入到JMenuBar
setJMenuBar(MBar); //将JMenuBar设置到窗
}

public JMenu buildFileMenu() {
    JMenu thefile = new JMenu("File");
    thefile.setMnemonic('F'); //Alt+F

    JMenuItem newf = new JMenuItem("New", new ImageIcon("icons/new.jpg"));
    JMenuItem open = new JMenuItem("Open", new ImageIcon("icons/open.jpg"));
    JMenuItem close = new JMenuItem("Close", new ImageIcon("icons/close.jpg"));
    JMenuItem quit = new JMenuItem("Exit", new ImageIcon("icons/exit.jpg"));

    newf.setMnemonic('N'); //Alt+N
    open.setMnemonic('O'); //Alt+O
    close.setMnemonic('C'); //Alt+C
    quit.setMnemonic('X'); //Alt+X
}

```



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 175

例8_12：菜单与工具栏的综合示例TestJMenuAndJToolBar.java

```

newf.setAccelerator( //Ctrl+N
    KeyStroke.getKeyStroke('N', java.awt.Event.CTRL_MASK, false));
open.setAccelerator( //Ctrl+O
    KeyStroke.getKeyStroke('O', java.awt.Event.CTRL_MASK, false));
close.setAccelerator( //Ctrl+L
    KeyStroke.getKeyStroke('L', java.awt.Event.CTRL_MASK, false));
quit.setAccelerator( //Ctrl+X
    KeyStroke.getKeyStroke('X', java.awt.Event.CTRL_MASK, false));

newf.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        theArea.append("- Menuitem New Performed -\n");
    }
});

open.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        theArea.append("- Menuitem Open Performed -\n");
    }
});

```



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 176

例8_12：菜单与工具栏的综合示例TestJMenuAndJToolBar.java

```

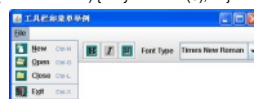
close.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        theArea.append("- Menuitem Close Performed -\n");
    }
});

quit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) { System.exit(0); }
});

thefile.add(newf);
thefile.add(open);
thefile.add(close);
thefile.addSeparator();
thefile.add(quit);

return thefile;
}

```



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 177

例8_12：菜单与工具栏的综合示例TestJMenuAndJToolBar.java

```

public JToolBar buildToolBar() {
    JToolBar toolbar = new JToolBar();
    toolbar.setFloatable(true); //设置JToolBar可否浮动

    //分别构造三个ToolBarAction的类组件,准备将这三个组件设置到JToolBar中.ToolBarAction类是我们自己编写
    //的inner class,这个inner class在程序后半部分.这个类继承AbstractAction,AbstractAction是一个抽象类,实现
    //Action接口,而Action接口又继承ActionListener,interface又继承AActionListener,interface,因此可直接
    //在AbstractAction中覆写actionPerformed()方法,处理ActionEvent事件.在JToolBar中有一个addAction()方法,
    //只要传入Action参数就可以得到一个JButton对象,因此在程序中下面的98,100,102就是以这样的方式来构造出
    //JToolBar上的按钮组件

    ToolbarAction tba_new = new ToolbarAction("new",
        new ImageIcon("icons/new.jpg"));
    ToolbarAction tba_open = new ToolbarAction("open",
        new ImageIcon("icons/open.jpg"));
    ToolbarAction tba_close = new ToolbarAction("close",
        new ImageIcon("icons/close.jpg"));
}

```



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 178

例8_12：菜单与工具栏的综合示例TestJMenuAndJToolBar.java

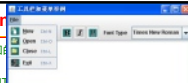
```

//利用setActionCommand()方法分别设置各个组件被触发事件后所返回
JButton JB;
JB = toolbar.add(tba_new); //将ToolBarAction组件加入JToolBar
JB.setActionCommand("#ToolBar_NEW performed!");
JB = toolbar.add(tba_open); //将ToolBarAction组件加入JToolBar
JB.setActionCommand("#ToolBar_OPEN performed!");
JB = toolbar.add(tba_close); //将ToolBarAction组件加入JToolBar
JB.setActionCommand("#ToolBar_CLOSE performed!");

toolbar.addSeparator(); //在JToolBar上加分隔线,如同在JMenu上利用addSeparator()方法加上分
隔线一样,不同的是在Menu中分隔线是以灰色直线的样式表现,而在JToolBar中则是以一小段空来表示
ToolBarAction tba_B = new ToolbarAction("bold",
    new ImageIcon("icons/bold.jpg"));
ToolBarAction tba_I = new ToolbarAction("italic",
    new ImageIcon("icons/italic.jpg"));
ToolBarAction tba_U = new ToolbarAction("underline",
    new ImageIcon("icons/underline.jpg"));

JB = toolbar.add(tba_B); //重复上面的操作
JB.setActionCommand("#ToolBar_Bold performed!");
JB = toolbar.add(tba_I);
JB.setActionCommand("#ToolBar_Italic performed!");
JB = toolbar.add(tba_U);
JB.setActionCommand("#ToolBar_Underline performed!");

```



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 179

例8_12：菜单与工具栏的综合示例TestJMenuAndJToolBar.java

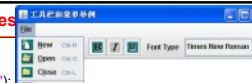
```

toolbar.addSeparator();
//在JToolBar中加入一个JLabel组件
JLabel JLfont = new JLabel("Font Type");
toolbar.add(JLfont);
toolbar.addSeparator();

//利用上面定义的ComboBox类建立一个JComboBox组件
JComboBox jcb = new JComboBox(ComboStr);
//将JComboBox加入事件处理模式,这里是将所选到的组件名称填入JTextArea中
jcb.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        theArea.append("ComboBox "+
            ((JComboBox)e.getSource()).getSelectedItem()+" performed\n");
    }
});

//在JToolBar中加入一个JComboBox组件
toolbar.add(jcb);
return toolbar;
}

```



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 180

例8_12：菜单与工具栏的综合示例TestJMenuAndJToolBar.java

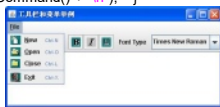
```

class ToolbarAction extends AbstractAction {
    public ToolbarAction(String name, Icon icon) { super(name, icon); }

    // 对于JToolBar上按钮的事件处理是将组件的ActionCommand返回值字符串加入JTextArea中.
    public void actionPerformed(ActionEvent e) {
        try {
            theArea.append(e.getActionCommand() + "\n");
        } catch (Exception ex) {}
    }
}

public class TestJMenuAndJToolBar {
    public static void main(String[] args) {
        JMenuAndJToolBarFrame F = new JMenuAndJToolBarFrame();
        F.setSize(430, 200);
        F.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        F.setVisible(true);
    }
}

```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时 38分 181

8.8 布局管理器

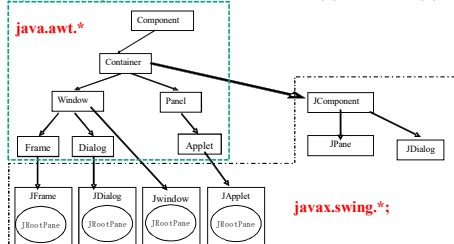
- 除了顶层容器组件外，其他的组件都需要添加到容器当中，容器相当于一个组件仓库
- 布局管理器采用一定的策略来管理容器中各个组件的大小、位置等属性。通过使用不同的布局管理器，可以方便地设计出各种组件组织方式的界面
- 高版本的 JDK 提供了 7 个主要的布局管理器，常用的布局管理器有：
 - (1) FlowLayout
 - (2) BorderLayout
 - (3) CardLayout
 - (4) GridLayout
 - (5) GridBagLayout
 - (6) XYLayout

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时 38分 182

布局管理器(续)

➢ 每个容器都有默认的布局管理器：

- ❖ (1) BorderLayout：除文件对话框(JFileChooser)外所有窗口
- ❖ (2) FlowLayout：所有面板，包括 Applet 和 JApplet



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时 38分 183

设置容器的布局管理器

➢ 调用容器对象的 setLayout 方法指定布局管理器

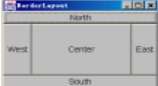
```
void setLayout(java.awt.LayoutManager manager)
```

➢ 设置容器的布局管理器后，调用容器的相关 add() 方法加入其它的容器和组件

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时 38分 184

8.8.1 java.awt.BorderLayout：边界布局

- BorderLayout 布局将空间划分为东、西、南、北、中五个区域，中央的是一个大型组件，四周是 4 个小型组件

**BorderLayout 的构造方法**

构造方法	说明
BorderLayout()	建立对象，窗口之间没有间隔
BorderLayout(int hgap,int vgap)	建立对象，窗口之间水平间隔为hgap，垂直间隔为vgap

BorderLayout主要方法

方法	说明
int getHgap()	返回窗口间的水平间隔
void setHgap(int hgap)	设置窗口间的水平间隔
int getVgap()	返回窗口间的垂直间隔
void setVgap(int vgap)	设置窗口间的垂直间隔

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时 38分 185

java.awt.BorderLayout

➢ BorderLayout 的常量(都是 String 类型)

常量	说明
BorderLayout.AFTER_LAST_LINE	对象放置在窗口最底部
BorderLayout.AFTER_LINE_ENDS	对象放置在窗口右部
BorderLayout.BEFORE_FIRST_LINE	对象放置在窗口的最上方
BorderLayout.BEFORE_LINE_BEGINS	对象放置在窗口的前方
BorderLayout.CENTER	对象放置在窗口的中央
BorderLayout.EAST	对象放置在窗口的右方
BorderLayout.NORTH	对象放置在窗口的上方
BorderLayout.SOUTH	对象放置在窗口的下方
BorderLayout.WEST	对象放置在窗口的左方

- ❖ 上面四个的优先级比下面四个的优先级高

➢ 调用容器的 add 方法将组件放入边界局部的适当位置：

```
void add(String position,Component c);
```

➢ 例8_13 采用BorderLayout布局管理器，每隔1秒动态地隐藏组件，直到隐藏全部的组件，然后按相反顺序重新显示组件TestBorderLayout.java

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时 38分 186

例8_13

```
import java.awt.*;
import javax.swing.*;
class BorderLayoutFrameClass extends JFrame {
    Container contentPane;
    BorderLayout layout;
    JButton buttons[];
    String names[] = {"North","South","East","West","Center"};

    public BorderLayoutFrameClass() {
        super("testBorderLayout");
        contentPane=getContentPane();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    private void fillComponent(Container c) {
        buttons=new JButton[names.length];
        for(int i=0; i<names.length; i++) buttons[i]=new JButton(names[i]);

        // 将各个按钮加入到指定的位置
        c.add(buttons[0],BorderLayout.NORTH);
        c.add(buttons[1],BorderLayout.SOUTH);
        c.add(buttons[2],BorderLayout.EAST);
        c.add(buttons[3],BorderLayout.WEST);
        c.add(buttons[4],BorderLayout.CENTER);
    }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 187

例8_13

```
public void testBorder() {
    layout=new BorderLayout(5,5); // 定义一个布局管理器
    contentPane.setLayout(layout); // 设置内容格的布局
    fillComponent(contentPane); // 向内容格中添加组件
    setSize(300,200);
    setVisible(true);
}

public void hideButton() { // 隐藏按钮
    for(int i=0; i<names.length; i++) {
        try { Thread.sleep(1000); } // 睡眠1秒
        catch(Exception e) { e.printStackTrace(); }
        buttons[i].setVisible(false);
        layout.layoutContainer(contentPane);
    }
}

public void showButton() { // 重新显示按钮
    for(int i=names.length-1; i>=0; i--) {
        try { Thread.sleep(1000); } // 睡眠1秒
        catch(Exception e) { e.printStackTrace(); }
        buttons[i].setVisible(true);
        layout.layoutContainer(contentPane);
    }
}
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 188

例8_13

```
public class TestBorderLayout {
    public static void main(String args[]) {
        BorderLayoutFrameClass obj=new BorderLayoutFrameClass();
        obj.testBorder();
        obj.hideButton(); // 隐藏按钮
        obj.showButton(); // 显示各个按钮
    }
}
```

运行结果演示

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 189

8.8.2 java.awt.FlowLayout: 流(顺序)布局

✿ FlowLayout 布局将组件以从左至右、从上到下的顺序依次摆放。当一行放不下时才往下一行放。每一行组件可以根据 FlowLayout 创建时 alignment 参数的指定要求放在屏幕的中心位置(缺省)、左侧或右侧

✿ FlowLayout的常量

常量名称	描述
FlowLayout.LEFT	此值(0)指示每一行组件都应该是左对齐的
FlowLayout.CENTER	此值(1)指示每一行组件都应该是居中的
FlowLayout.RIGHT	此值(2)指示每一行组件都应该是右对齐的
FlowLayout.LEADING	此值(3)指示每一行组件都应该与容器方向的开始边对齐, 例如, 对于从左到右的方向, 则与左边对齐
FlowLayout.TRAILING	此值(4)指示每一行组件都应该与容器方向的结束边对齐, 例如, 对于从左到右的方向, 则与右边对齐

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 190

FlowLayout的常用方法

方法	说明
FlowLayout()	建立对象, 居中对齐, 对象间水平及垂直间隔为5像素
FlowLayout(int align)	建立指定对齐方式的对象, 水平及垂直间隔为5像素
FlowLayout(int align,int hgap,int vgap)	建立指定对齐方式和水平及垂直间隔的对象
int getAlignment	返回对齐方式
int setAlignment(int align)	设置对齐方式
int getHgap()	返回对象间水平间隔
int setHgap(int align)	设定水平间隔
int getVgap()	返回垂直间隔
int setVgap()	设定垂直间隔
void layoutContainer(Container container)	设置container容器的布局方式为FlowLayout

➤ 例8_14 每隔1秒调整组件的对齐方式TestFlowLayout.java

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 191

例8_14

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

class FlowLayoutFrame extends JFrame {
    FlowLayout layout;
    Container contentPane;

    public FlowLayoutFrame() {
        super("FlowLayout测试");
        contentPane=this.getContentPane();
    }

    public void testFlow() {
        layout=new FlowLayout();
        setLayout(layout);
        for(int i=0; i<3; i++) contentPane.add(new JButton("Button "+i)); // 向框架中添加组件
        setSize(400,200);
        setVisible(true);
    }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 192

例8_14

```

public void changeAlign() {
    for(int i=0; i<5; i++) {
        try { Thread.sleep(1000); } // 睡眠1秒
        catch(Exception e) { e.printStackTrace(); }

        switch(i) {
            case 0: layout.setAlignment(FlowLayout.LEFT); break;
            case 1: layout.setAlignment(FlowLayout.CENTER); break;
            case 2: layout.setAlignment(FlowLayout.RIGHT); break;
            case 3: layout.setAlignment(FlowLayout.LEADING); break;
            case 4: layout.setAlignment(FlowLayout.TRAILING); break;
        }
        layout.layoutContainer(contentPane); // 重新排列有关的组件
    }
}

public class TestFlowLayout {
    public static void main(String args[]) {
        FlowLayoutFrame obj = new FlowLayoutFrame();
        obj.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        obj.testFlow();
        obj.changeAlign();
    }
}

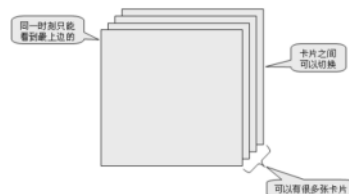
```

运行结果演示

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 193

8.8.3 java.awt.CardLayout: 卡片布局

- ✱ **CardLayout** 类可使两个或更多组件共享同一显示空间。CardLayout 所管理的组件就像放在纸盒里的纸牌，在某一时刻只有最上面的一张可见
- ✱ 所有被添加的组件与容器的大小相同，也就是说，所有添加进卡片布局容器中的组件大小是相同的



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 194

将组件加入卡片布局的容器

➤ 语法:

```
public Component add(String name, Component comp)
```

- ❖ name: 卡片组件名称，用于索引卡片组件
- ❖ comp: 卡片组件对象

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 195

CardLayout的主要方法

方法	说明
CardLayout()	建立对象，卡片之间的垂直和水平间隔为0
CardLayout(int hgap,vgap)	建立指定卡片之间的垂直和水平间隔的对象
void first(Container container)	显示容器中的第一个对象
void last(Container container)	显示容器中的最后一个对象
void next(Container container)	显示下一个对象
void previous(Container container)	显示上一个对象
int getHgap()	返回窗口对象间的水平间隔
int setHgap(int hgap)	设定窗口对象间的水平间隔
int getVgap()	返回窗口对象间的垂直间隔
void setVgap(int vgap)	设定窗口对象间的垂直间隔
void show(Container container, String name)	显示名称是name的对象

➤ 例8_15 采用CardLayout布局管理器，通过鼠标点击可以实现卡片之间的切换TestCardLayout.java

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 196

例8_15

```

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

class CardLayoutFrame extends JFrame implements ActionListener {
    JPanel deck;
    CardLayout cardManager; // CardLayout对象
    JButton buttons[];
    String names[] = {"First","Next","Previous","Last"};

    public CardLayoutFrame() { // testCardLayout的构造函数
        super("testCardLayout");
        deck = new JPanel();
        cardManager = new CardLayout(); // 定义一个布局管理器
        deck.setLayout(cardManager); // 设置内容格的布局

        JPanel card1 = new JPanel(); // 定义第1张卡片
        JLabel lab1 = new JLabel("卡片一"); // 上面仅有一个标签
        card1.add(lab1);
        deck.add(lab1.getText(), card1); // 将第1张卡片加入deck面板
    }
}

```



网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 197

例8_15

```

JPanel card2 = new JPanel(); // 定义第2张卡片
JTextField field = new JTextField("卡片二");
card2.add(field);
deck.add(field.getText(), card2); // 将第2张卡片加入deck面板

JPanel card3 = new JPanel(); // 定义第3张卡片
JLabel lab3 = new JLabel("卡片三");
card3.setLayout(new BorderLayout());
card3.add(new JButton("北"), BorderLayout.NORTH);
card3.add(new JButton("南"), BorderLayout.SOUTH);
card3.add(new JButton("西"), BorderLayout.WEST);
card3.add(new JButton("东"), BorderLayout.EAST);
card3.add(lab3, BorderLayout.CENTER);
deck.add(lab3.getText(), card3); // 将第3张卡片加入deck面板

// 创建控制面板
JPanel controls = new JPanel();
controls.setLayout(new GridLayout(2,2));
buttons = new JButton(names.length);
for(int i=0; i<names.length; i++) {
    buttons[i] = new JButton(names[i]);
    buttons[i].addActionListener(this);
    controls.add(buttons[i]); // 将按钮加入controls面板
}

```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 198

例8_15

```
// 将controls 和 deck加入到内容格
getContentPane().add(controls, BorderLayout.WEST);
getContentPane().add(deck, BorderLayout.EAST);
setSize(350, 200);
setVisible(true);
}

public void actionPerformed(ActionEvent evt){
    if(evt.getSource() == buttons[0])           // 显示第一张卡片
        cardManager.first(deck);
    else if(evt.getSource() == buttons[1])       // 显示下一张卡片
        cardManager.next(deck);
    else if(evt.getSource() == buttons[2])       // 显示前一张卡片
        cardManager.previous(deck);
    else if(evt.getSource() == buttons[3])       // 显示最后一张卡片
        cardManager.last(deck);
}

}

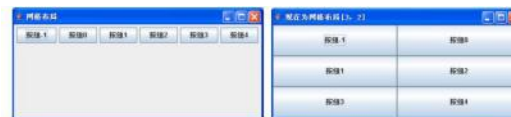
public class TestCardLayout {
    public static void main(String args[]) {
        new CardLayoutFrame().setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

运行结果演示

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 199

8.8.4 java.awt.GridLayout: 网格布局

- GridLayout 布局是将容器空间划分为若干行和若干列的网格区域, 并将组件放置在由行和列组成的网格中
- 每个组件占据一个网格单元的所有空间, 而且每个单元空间具有相同的尺寸
- 如果用户改变 GridLayout 所有管理的窗口的尺寸, 用户将会看到 GridLayout 改变了单元空间的尺寸, 使它们在容器的可用空间中尽可能的大
- 非常适合数量庞大的组件



运行结果

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 200

GridLayout类的主要方法

方法	说明
GridLayout()	建立一个行数为1的GridLayout布局管理器对象
GridLayout(int rows, int cols)	建立rows行cols列的对象
GridLayout(int rows, int cols, int hgap, int vgap)	建立指定行和列数及水平和垂直间隔的对象
int getColumns()	返回设定的水平网格数
int setColumns(int cols)	设置水平的网格数
int getRows()	返回垂直的网格数
int setRows(int rows)	设置垂直的网格数
int getHgap()	返回水平间隔
int setHgap(int hgap)	设置水平间隔
int getVgap()	返回垂直间隔
int setVgap(int vgap)	设置垂直间隔
void layoutContainer(Container container)	设置 container 容器的布局管理为 GridLayout

►例8_16 采用GridLayout布局管理器, 每隔1秒重新设置组件布局TestGridLayout.java

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 201

例8_16

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

class GridLayoutFrame extends JFrame {
    Container contentPane;
    GridLayout grid1, grid2;           // 定义两个GridLayout引用
    JButton buttons[];
    String names[] = {"one", "two", "three", "four", "five", "six"};
    boolean change;                    // 改变布局的标记

    public GridLayoutFrame() {
        super("GridLayout举例");
        contentPane=getContentPane();
        grid1=new GridLayout(2,3,5,5); // 定义一个2行3列的布局管理器
        grid2=new GridLayout(3,2);     // 定义一个3行2列的布局管理器
        contentPane.setLayout(grid1);  // 设置内容格的布局
        fillComponent(contentPane);    // 向内容格中添加组件
        change=true;                   // 设置改变标记为true
        setSize(220,200);
        setVisible(true);
    }
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 202

例8_16

```
private void fillComponent(Container c) {
    buttons=new JButton[names.length];

    for(int i=0;i<names.length;i++) {
        buttons[i]=new JButton(names[i]);
        c.add(buttons[i]);           // 将按钮加入容器
    }
}

public void testGrid() {
    while(true) {
        try { Thread.sleep(1000); } // 睡眠1秒
        catch(Exception e) { e.printStackTrace(); }

        if(change) contentPane.setLayout(grid1); // 采用grid1设置内容格的布局
        else contentPane.setLayout(grid2); // 采用grid2设置内容格的布局
        change = !change; // 反转标记
        contentPane.validate(); // 使改变后的布局有效
    }
}
}
```

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 203

例8_16

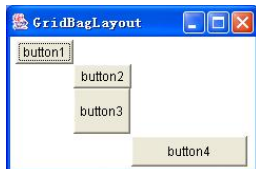
```
public class TestGridLayout {
    public static void main(String args[]) {
        GridLayoutFrame obj=new GridLayoutFrame();
        obj.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        obj.testGrid();
    }
}
```

运行结果演示

网络安全与网络工程系系东平 jsxhbc@163.com Java语言与网络编程 2022年12月28日 3时38分 204

8.8.5 java.awt.GridBagLayout: 网格包布局管理器

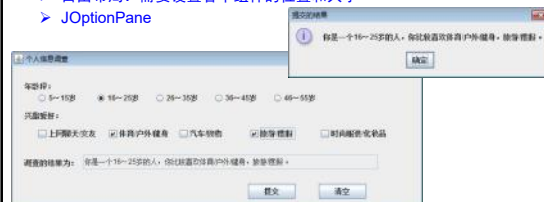
- GridBagLayout 是最灵活、最复杂的布局管理器
 - 1) 网格布局中的每个网格的宽度和高度可以不一样
 - 2) 每个组件可以占据一个或是多个网格
 - 3) 可以指定组件在网格中的停靠位置
- 当将一个 GUI 组件添加到使用了网格包布局的容器中时, 需要指定该组件的位置、大小以及缩放等一系列约束条件
 - 用 java.awt.GridBagConstraints 类型的对象来存储这些约束条件



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 205

8.9 图形用户界面综合案例

- 例8_17 综合示例checkAndRadio.java
 - JCheckBox
 - JRadioButton
 - JLabel
 - JButton
 - JTextField
 - 自由布局: 需要设置各个组件的位置和大小
 - JOptionPane



网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 206

例8_17

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

public class checkAndRadio extends JFrame implements ActionListener {
    private JPanel jp=new JPanel(); //创建JPanel对象

    //创建复选框数组
    private JCheckBox[] jcbArray = {
        new JCheckBox("上网聊天/交友"), new JCheckBox("汽车/购物"),
        new JCheckBox("旅游/度假"), new JCheckBox("时尚服装/化妆品")
    };

    //创建单选按钮数组
    private JRadioButton[] jrbArray = {
        new JRadioButton("5~15岁"), new JRadioButton("26~35岁"),
        new JRadioButton("16~25岁", true), new JRadioButton("46~55岁")
    };

    //创建按钮数组
    private JButton[] jbArray={ new JButton("提交"), new JButton("清空")};

    //创建标签数组
    private JLabel[] jlArray= { new JLabel("年龄段: "), new JLabel("兴趣爱好: ")
    };
}
```

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 207

例8_17

```
private JTextField jtf=new JTextField(); //创建文本框
private ButtonGroup bg=new ButtonGroup(); //创建按钮组

public checkAndRadio() { //设置JPanel布局管理器
    jp.setLayout(null); //自由布局
    for(int i=0;i<5;i++) { //对各个控件进行设置
        //自由布局需要设置单选按钮与复选框的大小位置
        jrbArray[i].setBounds(40+i*100,40,80,30); //计算组件位置和大小
        jcbArray[i].setBounds(40+i*120,100,120,30); //计算组件位置和大小
        //将单选按钮与复选框添加到JPanel中
        jp.add(jrbArray[i]); jp.add(jcbArray[i]);
        //为单选按钮与复选框注册动作事件监听器
        jrbArray[i].addActionListener(this); jcbArray[i].addActionListener(this);
        //将单选按钮添加到按钮组中
        bg.add(jrbArray[i]);
        if(i>1) continue;
        //设置标签与普通按钮的大小位置
        jlArray[i].setBounds(20,20+i*50,80,30);
        jbArray[i].setBounds(400+i*120,200,80,26);
        //将标签与普通按钮添加到JPanel中
        jp.add(jlArray[i]); jp.add(jbArray[i]);
        jlArray[i].addActionListener(this); //为普通按钮注册动作事件监听器
    }
}
```

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 208

例8_17

```
//设置调查结果标签的大小位置, 并将其添加到JPanel中
jlArray[2].setBounds(20,150,120,30);
jp.add(jlArray[2]);

//设置调查结果文本框的大小位置, 并将其添加到JPanel中
jtf.setBounds(120,150,500,26);
jp.add(jtf);
jtf.setEditable(false); //设置显示调查结果的文本框为不可编辑状态
this.add(jp); //将JPanel添加进窗体

//设置窗体的标题、大小位置以及可见性等
this.setTitle("个人信息调查");
this.setBounds(100,100,700,280);
this.setVisible(true);
this.setResizable(false); //运行过程中不允许改变窗体大小
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
```

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 209

例8_17

```
public void actionPerformed(ActionEvent e) {
    if(e.getSource()==jbArray[1]) { //清空按钮执行的动作
        bg.clearSelection();
        for(int i=0;i<jcbArray.length;i++) jcbArray[i].setSelected(false);
        jtf.setText("");
    } else if(e.getSource()==jbArray[0]) { //提交按钮,用通用对话框显示选择结果
        JOptionPane.showMessageDialog(this, jtf.getText(),
            "提交的结果", JOptionPane.INFORMATION_MESSAGE);
    } else { //其他按钮执行的动作
        //创建两个临时字符串
        StringBuffer temp1=new StringBuffer("你是一个");
        StringBuffer temp2=new StringBuffer();
        for(int i=0; i<5; i++) { //获取年龄段的选中值
            if(jrbArray[i].isSelected()) { temp1.append(jrbArray[i].getText()); }
            //获取爱好的选中值
            if(jcbArray[i].isSelected()) { temp2.append(jcbArray[i].getText()+" "); }
        }
    }
}
```

网络安全与网络工程系系平 jsxhbc@163.com Java语言与网络编程 2022年12月28日3时38分 210

例8_17

```
//在文本域中显示结果
if(temp2.length()==0) { //如果没有选取爱好
    jtf.setText("兴趣爱好选项不能为空!!!");
} else { //选取了爱好
    temp1.append("的人，你比较喜欢");
    temp1.append(temp2.substring(0,temp2.length()-1));
    jtf.setText(temp1.append(".").toString());
}
}
}

public static void main(String[] args) { //创建checkAndRadio窗体对象
    new checkAndRadio();
}
}
```

运行结果演示

小结

✿ 本章介绍了 Java 中图形用户界面 Swing 的编写。通过本章的学习，熟练掌握 Java 提供的图形组件的使用，才能开发出丰富的图形用户界面