



- 13.1 Swing概述
- 13.2 Swing常用窗体
- 13.3 常用布局管理器
- 13.4 常用面板
- 13.5 按钮组件
- 13.6 列表组件
- 13.7 文本组件
- 13.8 表格组件
- 13.9 事件监听器



13.1 Swing概述

01

Java Swing 是在 Java 版本中 引入的,当时称为"Swing", 后来在 Java 版本中更名为

"Java Swing".

Swing 是 Java 版本中引入的 第一个 GUI 工具包, 它在 Java 版本中得到了进一步的 发展和完善,



02



13.1 Swing概述

Swing的出现极大地丰富了 Java 的图形用户界面, 使得 Java 应用程序可以拥有更加 美观和友好的用户界面,

03

● JFrame简介

• JFrame是Swing中最基本的顶层容器之一,代表一个窗口。它提供了窗口的所有功能,如打开、关闭、最小化、最大化等。JFrame可以包含其他组件,如按钮、文本框等,以构建复杂的用户界面。



- 创建JFrame窗体
 - 创建JFrame窗体首先需要导入Javax.swing.JFrame类。
 - 然后实例化一个JFrame对象,并调用setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)
 设置窗体的关闭操作。
 - 最后使用setSize(int width, int height)设置窗体的大小,并用setVisible(true)设置窗体为可见。

13.2.1 JFrame窗体

• 创建JFrame窗体代码示例



● JDialog简介

• Jdialog是Java Swing库中的一个类,用于创建对话框。对话框是一种特殊的窗口,用于与用户交互,

显示信息或获取输入。Jdialog是Swing库中用于创建对话框的主要类之一。



• JDialog的作用

JDialog可以用于显示提示信息,例如错误消息或确认操作的消息。它还可以用于与用户进行更复杂的交互,例如输入数据或选择选项Jdialog提供了多种预定义的对话框类型,如消息对话框、选项对话框和文件选择对话框等。

13.2.2 JDialog对话框

• 创建JDialog代码示例



- JOptionPane的功能
 - JOptionPane是Java Swing库中用于创建和管理对话框的类。 它可以用来显示各种类型的消息,如

警告、错误、提示信息等。 通过JOptionPane, 开发者可以轻松创建简单的用户交互界面。

• 组件的使用场景

• 在需要与用户进行简单交互时,可以使用JOptionPane显示对话框。例如,询问用户是否确定执行

某操作、显示系统提示信息或获取用户输入等。

- showMessageDialog方法
 - showMessageDialog方法用于显示带有指定消息和标题的对话框。参数包括对话框的父组件、消息

字符串、标题字符串以及对话框类型(如信息、警告、错误等)。



- showConfirmDialog方法
 - showConfirmDialog方法用于显示一个确认对话框、询问用户是否选择某个选项。参数包括对话框

的父组件、提示消息字符串、标题字符串以及可选的按钮标签数组。



- showInputDialog方法
 - showinputDialog方法用于显示一个输入对话框,允许用户输入文本。参数包括对话框的父组件、

提示消息字符串、标题字符串以及用于输入的文本域对象。



13.2.3 JOptionPane小型对话框

• 创建JOptionPane代码示例

1	pack	<pre>kage org.forhis.chapter_13;</pre>
2		
3	impo	ort javax.swing.JOptionPane;
4		
5 ▷	publ	.ic class c_13_demo_3 {
6		
7 ⊳ 🕈		<pre>public static void main(String[] args) {</pre>
8		
9		J0ptionPane optionPane_0 = new J0ptionPane();
10		JOptionPane. <i>showMessageDialog</i> (optionPane_0, message: "通知消息框");
11		
12		J0ptionPane optionPane_1 = new J0ptionPane();
13		JOptionPane. <i>showConfirmDialog</i> (optionPane_1, message: "确认消息框");
14		
15		J0ptionPane optionPane_2 = new J0ptionPane();
16		JOptionPane. <i>showInputDialog</i> (optionPane_2, message: "输入消息框");
17		}
18	}	
10		

• 绝对布局

- 绝对布局。绝对布局没有特定一个布局管理器类来表示,给容器的布局管理器设置为 null,就表示
 使用绝对布局,即通过设置组件的坐标和宽高来布置组件。
- 绝对布局需要明确指定每一个组件的坐标和宽高,否则不显示。

- FlowLayout流布局管理器
 - FlowLayout是Swing的默认布局管理器,适用于创建简单的单行界面。组件按照添加到容器的顺序
 从左到右排列,当容器的宽度不足时组件会换行。
 - FlowLayout的示例
 - 示例代码: JPanel panel = new JPanel(new FlowLayout())
 - 示例描述:创建一个采用FlowLayout布局管理器的面板。

🔮 Fl	ow Layout		
	Button One	Butto	n Two
	Butte	on Three	

13.3.3 BorderLayout边界布局管理器

- BorderLayout边界布局管理器
 - BorderLayout的特点
 - BorderLayout将容器分为五个区域:北、南、东、西、中。每个区域只能放置一个组件,当容器的大小改变时,组件会自动重新布局。
 Border Layout
 - BorderLayout的示例
 - 示例代码: JPanel panel= new JPanel(new BorderLayout());
 - 示例描述:创建一个采用BorderLayout布局管理器的面板。

🎂 Border I	Layout	_ 🗆 🗙	
	North		
West	Center	East	
South			

13.3.4 GridLayout网格布局管理器

- GridLayout网格布局管理器
 - GridLayout的特点
 - Gridlayout使容器中的组件呈网状排列,每个组件占据一行一列。适用于创建行数和列数固定的界面,当容器的大小改变时,组件会自动拉伸以填充空间。
 - GridLayout的示例
 - 示例代码: JPanel panel = new JPanel(new GridLayout(2,3));
 - 示例描述:创建一个采用GridLayout布局管理器的面板
 - 包含2行3列的组件。

ndBag		
One	Two	Three
Four	F	ive
	5	ŝix
	Seven	

13.4.1 JScrollPane滚动面板

- JScrollPane滚动面板
 - 实现滚动功能
 - JScrollPane 组件可以为其他组件提供滚动功能,如JTable、JList、JTextArea 等。
 - 当组件内容超出视口范围时, JScrollPane 会自动显示滚动条,以便用户能够查看全部内容。

• 淡井久松士宁山		JScrollPane 示例
●依幼家件以足削	这里是一些文本,	足够长以至于需要滚动条来查看
• JScrollPane 提供了多种滚动条样式,		
• 可以通过属性设置来调整滚动条的外观。		
• 用户可以通过设置滚动条的颜色、粗细等属性来定制滚动条的样式。		

13.4.1 JScrollPane滚动面板

• JScrollPane滚动面板代码示例



- 标签组件的基本概念
 - •标签(JLabel)是 Swing 中的一种常用组件,用于显示文本、图像或二者的组合。
 - •标签通常用于提供说明性文本或提示信息,不提供交互功能。

- 创建和定制标签
 - 使用 JLabel 类创建标签,可以设置文本、图像和horizontal alignment。
 - 标签的文本可以通过 setText(String text) 方法设置,图像可以通过 setIcon(lcon icon)
 方法设置
 - 通过布局管理器,可以控制标签在容器中的位置和大小。

13.4.3 JLabel标签

• 标签事件处理

- •标签本身不提供交互功能,但可以绑定事件监听器来响应用户的交互操作。
- 例如,可以监听鼠标事件,实现点击标签时触发特定操作。

13.4.4 图标的使用

- 创建和使用图标
 - Swing 提供了 Icon 接口,用于定义图标的行为。
 - 可以创建自定义图标类实现 Icon 接口,也可以使用 Swing 提供的内置图标。
 - 图标可以通过 JLabel 的 setIcon(Icon icon) 方法或 JButton 的 setIcon(Icon icon) 方法设置。

13.4.4 图标的使用

• 图标事件处理

- •图标本身不提供交互功能,但可以绑定事件监听器来响应用户的交互操作。
- 例如,可以监听鼠标事件,实现点击图标时触发特定操作。

- JButton的概念
 - JButton的定义
 - JButton是Java编程语言中Swing图形用户界面(GU)工具包的一部分。它代表一个按钮、用户可以点击它来 执行命令或触发事件。JButton是Java Swing组件库中用干创建桌面应用程序的基本元素之一。
 - Jbutton的作用
 - JButton允许用户在应用程序中执行各种操作、如提交表单、启动任务或调用其他功能。通过为JButton添加事件监听器、可以响应用户的点击动作并执行特定的代码。JButton通常用于图形用户界面中的命令操作,如启动或停止某个功能。

🕌 Button Example		_	×
	Click Here		

- JButton的特点
 - 外观定制
 - JButton支持多种外观定制选项、如字体、颜色、边框和背景。 可以通过设置不同的属性来调整按钮的大小
 - 、形状和视觉风格。 JButton的外观可以与应用程序的整体主题和风格保持一致。
 - 行为设置
 - JButton可以配置多种行为,如默认按钮、禁用状态和可见性。可以通过设置按钮的 Enabled 属性来控制 其是否可以响应用户操作JButton的 Text 属性用于设置按钮上显示的文本,以便用户明确了解其功能。

- 创建JButton
 - 导入Swing库
 - 在使用Jbutton之前,需要在Java代码中导入Swing库相关的类。导入语句为:import javax.swing.Jbuton;
 - 创建按钮实例
 - 使用 new 关键字和 Jbutton 类创建一个按钮实例。例如:Jbutton myButton =new Jbutton();
 - 设置按钮属性
 - 设置按钮的文本、大小、位置和其他属性。例如:
 - myButton.setBounds(50,50,100,50);
 - myButton.setFont(new Font(ArialFont.BOLD, 16));

- JRadioButton的定义
 - JRadioButton的概念
 - JRadioButon是Java Swing库中的一个组件,用于在GUI应用程序中创建单选按钮。它是 ButtonGroup组件的一部分,用于实现一组单选按钮,用户只能从中选择一个选项。
 - JRadioButton的作用
 - JRadioButton允许用户在多个选项中进行选择,适用于简化用户界面和提高用户体验。通过与 ActionListener接口结合使用,可以响应用户的点击事件,实现特定的功能。



13.5.2 JRadioButto按钮

13.5.2 JRadioButto按钮

- JRadioButton的基本使用
 - 创建JRadioButton
 - 首先需要导入Swing库和ActionListener接口。然后可以使用ButtonGroup来创建一个按钮组
 - ,用于管理多个单选按钮。 最后创建JRadioButton对象,并将其添加到按钮组中。
 - 为JRadioButton添加事件监听器
 - 通过实现Actionlistener接口的actionPerformed方法,可以为JRadioButton添加事件监听器
 - 。在事件监听器中,可以编写当用户点击JRadioButton时的具体操作。

- JCheckBox的定义与功能
 - 组件概述
 - JCheckBox是Java Swing库中的一个复选框组件
 - 用于在图形用户界面(GUI)中允许用户进行多项选择。
 - 安装与使用
 - JCheckBox是Swing的一部分,因此只需在项目中引入Swing库即可使用。在代码中,可以通过JCheckBox类创建一个复选框,并利用布局管理器将其添加到JFrame或JPanel等容器中。

🛃 JCheckbox 💷 💷 🔀
Happy Sad
Happy is unchecked

13.5.3 JCheckBox复选框

- JCheckBox的常用属性和方法
 - 常用属性
 - selected: 获取或设置复选框是否被选中。
 - text: 设置复选框旁边的文本。
 - actionCommand: 获取或设置与该复选框关联的动作命令字符串。
 - 主要方法
 - setSelected(boolean b): 设置复选框是否选中。
 - isSelected(): 获取复选框是否选中的状态。
 - addItemListener(ItemListener l): 为复洗框添加项监听器,以便在状态改变时作出响应。

- 创建一个简单的JCheckBox
 - 代码实现
 - 创建JFrame和JCheckBox实例。使用布局管理器(如BorderLayout或FlowLavout)将
 JCheckBox添加到JFrame中。为JCheckBox添加事件监听器,以便响应用户的选择操作。
 - 运行程序,显示包含复选框的GUI界面。用户可以点击复选框进行选择,程序会根据选择情况
 作出相应反馈。

- JComboBox的定义与功能
 - JComboBox的概述

13.6.1 JComboBox下拉列表框

🔬 Swing JComboBox Demo	
Effective Java	Select Remove
Effective Java	
Head First Java	
Thinking in Java	
Java Generics and Collections	
Beginnning Java 7	
Java I/O	

• JComboBox是Java Swing库中的一种组件,用于在图形用户界面(GU1)中提供下拉列表的控

件。通过JComboBox,用户可以从预定义的选项中选择一个值,或者输入新的值。

JComboBox通常用于简化用户的选择过程,并减少界面上的空间占用。

- JComboBox的用途
 - JComboBox广泛应用于各种应用程序中,如数据输入、配置选项、过滤条件等。它特别适合 于那些具有固定选项集的场景,如选择日期性别、国籍等。

13.6.1 JComboBox下拉列表框

- JComboBox的属性与方法
 - JComboBox的属性
 - JComboBox包含多种属性,如字体、颜色、边框等、可以自定义其外观和风格。可以通过设置属性来调整下拉列表的显示效果、以符合应用程序的整体风格。
 - JComboBox的方法
 - JComboBox提供了一系列的方法,用于控制其行为和功能、如添加选项、删除选项、获取选中

项等。这些方法可以灵活地操作JComboBox,以实现各种复杂的功能需求。

13.6.1 JComboBox下拉列表框

- 创建JComboBox
 - 导入Swing库
 - 在使用JComboBox之前,需要导入Java Swing库中的相关类。导入语句为: import javax.swing。
 - 创建JComboBox实例
 - 通过使用new关键字和JComboBox类的构造函数,可以创建一个JComboBox实例。
 - 例如: JComboBox<String> comboBox= new JComboBox<String>();

13.6.1 JComboBox下拉列表框

- 添加选项到JComboBox
 - 使用addltem方法
 - 通过调用JComboBox的addltem方法,可以向JComboBox中添加新的选项。
 - 例如: comboBox.addltem(选项1);
 - 使用addltemAt方法
 - 如果需要在特定的位置添加选项,可以使用addltemAt方法,并指定选项的索引。
 - 例如: comboBox.addltemAt(选项2, 1);

- JList 属性与方法
 - JList 属性
 - 数据模型: JList 需要一个数据模型来显示条目,通常使用 DefaultlistModel 类。
 - 添加条目: 使用 addlistDatalistener 方法来添加监听器,监听用户选择条目的事件。
 - 获取选中条目: 通过 getSelectedValue 方法可以获取用户选中的条目。



13.7 文本组件

• JTextField

- JTextField 是 Java Swing 库中的一个类,用于创建一个可以编辑的单行文本字段。用户可以在 JTextField 中输入文本,这些文本可以被程序读取和处理。JTextField 继承自 JTextComponent 类,并提供了许多方法来获取和设置文本字段的内容、字体、颜色等属性。
- 要创建一个 JTextField,可以直接使用其构造器。构造器可以不带参数,也可以接受一个字符串作 为初始文本。

```
// 创建一个空的 JTextField
JTextField textField = new JTextField();
// 创建一个包含初始文本的 JTextField
JTextField textFieldWithText = new JTextField("Hello, JTextField!");
```



- JPasswordField
 - JPasswordField 是 Java Swing 库中的一个组件,用于输入密码或敏感信息。与 JTextField 类似, JPasswordField 允许用户输入文本,但出于安全考虑,它不会直接显示用户输入的字符。相反,它
 会用占位符(通常是圆点或星号)来替换每个字符,从而隐藏输入的内容。

	실 Java Swing JPasswordField示例	-	\times
	Control in action: JPasswordField		
// 创建 JPasswordField 实例 JPasswordField passwordField = new JPasswordField(10); // 初始容量为10个字符	用户名: yiibal 密码 ~~~~~ 登录		
	用户名:yiibai, 密码: 123456		

13.7.3 JTextArea文本域

• JTextArea文本域

13.7 文本组件

- JTextArea 是 Java Swing 库中的一个组件,用于显示多行文本,并允许用户编辑这些文本。与 JTextField (只能显示单行文本)不同,JTextArea 提供了更广泛的文本编辑功能,包括滚动条 (如果文本超出了可视区域)、文本选择、复制和粘贴等。
- 要创建 JTextArea, 你可以直接使用其构造器,并可选地指定行数和列数(这些参数主要用于计算 初始的首选大小,并不限制用户可以输入的文本量)。





- JTable的概念
 - JTable的定义
 - JTable是Java Swing库中的一个类,用于在GUI应用程序中显示和编辑表格数据。JTable继承自 JTableModel,它是一个用于表示表格数据的模型。
 - JTable的作用
 - JTable可以用于显示数据列表,如数据库结果集或集合。
 - 用户可以通过JTable对数据进行排序、过滤和编辑。
 - 列属性
 - 列标题:用于显示表格中每列的名称。
 - 列类型:用于指定列中数据的类型,如整数、浮点数、字符串等。 列宽度:用于设置列的宽度,影响显示效果。
 - 行属性
 - 行数据:用于存储表格中的数据,每个单元格对应一行。行高度:用于设置行的 height,影响显示效果。

•••	JTable	示例	
姓名	年龄	城市	
张三	28	北京	
李四	22	上海	
王五	35	广州	

13.8.2 DefaultTableModel表格数据模型

- DefaultTableModel表格数据模型
 - DefaultTableModel 是 Java Swing 库中用于管理 JTable 数据的默认数据模型。它实现了 TableModel 接口,并提供了一种方便的方式来管理表格中的数据,包括添加、删除和修改行和列中 的数据。
 - 主要特点
 - 自动管理数据: DefaultTableModel 自动管理其内部的行和列数据,并提供了丰富的 API 来操 作这些数据。
 - 数据可编辑性:默认情况下,DefaultTableModel中的数据是可编辑的,但你可以通过调用 setEditable(false)方法来禁止编辑。
 - 列标题:你可以在创建 DefaultTableModel 时指定列标题,这些标题将自动显示在 JTable 的 列头上。

● 行和列的监听: DefaultTableModel 提供了对行和列变化的监听支持,允许你在数据变化时执 河南中医药大学信息技术学院(智能医疗行业学院)智能医疗教研室 / https://aitcm.hactcm.edu.cn 2025年2月16日星期日 第43页

13.8.2 DefaultTableModel表格数据模型

• DefaultTableModel示例



13.8.3 维护表格模型

- 维护表格模型
 - 使用getValueAt()方法获得表格中某一个单元格的值
 - 使用addRow()方法向表格中添加新的行
 - 使用setValueAt()方法修改表格中某一个单元格的值
 - 使用removeRow()方法从表格中删除指定行等

- ActionEvent动作事件
 - 在Java中,ActionEvent是用于监听用户界面(UI)组件(如按钮点击,菜单项选择等)的动作事件的一种事件。ActionEvent类是java.awt.event包的一部分,它包含有关发生的事件的信息,例如事件源,发生事件的时间以及事件的类型。
 - 当用户执行某种动作(如点击按钮)时,相应的组件会生成一个ActionEvent对象,然后调用其监听器的actionPerformed方法。这个方法是在ActionListener接口中定义的。

13.9.1 ActionEvent动作事件

• ActionEvent示例

1	<pre>import javax.swing.*;</pre>	
2		
3	<pre>import java.awt.*;</pre>	
4	<pre>import java.awt.event.ActionEvent;</pre>	
5	<pre>import java.awt.event.ActionListener;</pre>	
6		
7		
8	public class Main extends JFrame {	
9	private JButton jb=new JButton("我是按钮, 点击我");	
.0	<pre>public Main() {</pre>	
.1	<pre>setLayout(null);</pre>	_
.2	setSize(200,100);	· cp
.3	<pre>setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);</pre>	· · · ·
.4	<pre>Container cp=getContentPane();</pre>	
.5	cp.add(jb);	
.6	jb.setBounds(10,10,150,30);	
.7	<pre>jb.addActionListener(new jbAction());</pre>	
.8	<pre>setVisible(true);</pre>	
.9		
.0		
1	}	
2	class jbAction implements ActionListener{	
3	<pre>public void actionPerformed(ActionEvent arg0) {</pre>	
.4	jb.setText("我被点击了");	
.5	}	
:6	}	
!7	<pre>public static void main(String[] args) {</pre>	
8	new Main();	
.9		
0	}	
1		
2		





13.9.2 KeyEvent键盘事件监听器

- KeyEvent键盘事件监听器
 - 在Java中, KeyEvent是用于捕获键盘事件的一个类。
 - 当向文本框中输入内容时,会发出键盘事件。KeyEvent类负责捕获这些键盘事件,并且可以通过为组件添加实现了KeyListener接口的监听器类来处理相应的键盘事件。
 - KeyListener接口定义了三个抽象方法,它们分别在发生击键事件、按键被按下和释放时被 触发。具体地,当按下一个键时会调用KeyPressed处理器,当松开一个键时会调用 KeyReleased处理器,当输入一个统一编码时会调用KeyTyped处理器。
 - public void keyTyped(KeyEvent e): 发生击键事件时被触发
 - public void keyPressed(KeyEvent e): 按键被按下时被触发
 - public void keyReleased(KeyEvent e): 按键被释放时被触发

- MouseEvent鼠标事件
 - 所有组件都能发生鼠标事件, MouseEvent类负责捕捉鼠标事件, 可以通过为组件添加实现了 MouseListener接口的监听器类来处理相应的鼠标事件
 - MouseListener接口有5个抽象方法:
 - public void mouseEntered(MouseEvent e):光标移入组件时被触发
 - public void mousePressed(MouseEvent e):鼠标按键被按下时被触发
 - public void mouseReleased(MouseEvent e):鼠标按键被释放时被触发
 - public void mouseClicked(MouseEvent e):发生单机事件时被触发
 - public void mouseExited(MouseEvent e):光标移出组件时被触发



- 掌握以下Java Swing常用组件的概念及使用场景
 - 常用窗体 JFrame, JDialog, JOptionPane
 - •常用布局管理 null绝对布局, FlowLayout, BorderLayout, GridLayout 布局的区别
 - 常用面板 JScrollPane, JLabel
 - 按钮组件 JButton, JRadioButton, JCheckBox
 - 文本组件 JTextField, JPasswordField, JTextArea
- 能通过代码实现简单的表单并绑定按钮事件



- 使用JFrame组件创建一个窗体,设置窗体的高度和宽度,以及添加关闭、最小化、最大化按钮,并尝试设置窗体不同的Layout形式。
- 根据上一步骤JFrame设置的Layout,添加左右两个JPanel。
- 在左侧的JPanel中添加一个自定义表单,表单录入内容可以是患者的基础信息,并在表单最后添加一个提交按钮。
- 在右侧的JPanel中添加一个JTable,并初始化一些基础数据于JTable进行绑定显示。



河南中医药大学信息技术学院(智能医疗行业学院)智能医疗教研室 河南中医药大学医疗健康信息工程技术研究所