



第7章 Java对话框

- 对话框(**JDialog**)
- 消息对话框(**JOptionPane**)
- 确认对话框
- 文件对话框(**JFileChooser**)
- 颜色对话框(**JColorChooser**)



对话框(JDialog)

- 对话框(JDialog)是与框架类似的可移动窗口，区别在于具有较少的修饰并且能够被设置为“模式(modal)”窗口
- 对话框的构造方法
 - ◆ JDialog(Frame owner) ——构造一个没有标题的非模式对话框
 - ◆ JDialog(Frame owner, boolean modal) ——构造一个没有标题的对话框，boolean型参数modal指定对话框是否为模式窗口
 - ◆ JDialog(Frame owner, String title) ——构造一个有标题的非模式对话框
 - ◆ JDialog(Frame owner, String title, boolean modal) ——构造一个有标题的对话框



对话框(JDialog)

- 对话框分为无模式和有模式两种。
 - ◆ 如果一个对话框是有模式的对话框，那么当这个对话框处于激活状态时，只让程序响应对话框内部的事件，程序不能再激活它所依赖的窗口或组件，而且它将堵塞当前线程的执行，即堵塞使得对话框处于激活状态的线程，直到该对话框消失不可见。
 - ◆ 无模式对话框处于激活状态时，程序仍能激活它所依赖的窗口或组件，它也不堵塞线程的执行。



对话框

- 构造一个标题为 “Dialog” 的模式对话框，该对话框为框架 **frame** 所拥有

```
JDialog dialog = new JDialog(frame, “Dialog”, true)
```

- 当对话框的拥有者被清除（**destroyed**）时，对话框也会被清除
- 对话框在显示时，如果其拥有者被最小化，对话框也将变为不可见，当其拥有者再次显示时，对话框会随之变为可见
- **setVisible(true)** 方法显示，**setVisible(false)** 方法可以将一个对话框隐藏起来



对话框

- 对话框是一种可以反复使用的资源，当某个对话框不需要显示时，不要立即将其清除，而是继续保留它，等待以后再用
- 对话框可对各种窗口事件进行监听，例如激活窗口、关闭窗口等
- 对话框也是顶层容器，我们可以向对话框的内容窗格中添加各种组件



程序1

- 例1构造了一个对话框，当用户按框架中的按钮时，对话框将被显示出来





```
■ import java.awt.*; import javax.swing.*;import java.awt.event.*;
■ public class MyJFrame extends JFrame implements ActionListener
■ {
■     JDialog dlgHello =new JDialog(this, "HelloIamDialog");
■     public MyJFrame (String str) {         super(str);     }
■     void init(){
■         JButton b1=new JButton("show Dialog");
■         Container con=this.getContentPane();
■         JPanel pan = new JPanel( );         pan.setSize(300,300);
■         b1.addActionListener(this);         pan.add(b1);
■         con.setLayout(null);                 con.add(pan);
■         this.setSize(300,300);                 this.setVisible(true);
■     }
■     public void actionPerformed( ActionEvent e ){
■         dlgHello.setBounds(100, 100, 30, 100);
■         dlgHello.add(new JLabel("Hello Dialog"));
■         dlgHello.setVisible(true);
■     }
■     public static void main(String args[]) {
■         MyJFrame x=new MyJFrame("Hello");     x.init();
■     }
```



```
■ package p1;

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

■ public class MyJFrame extends JFrame implements ActionListener {
■
■     JDialog dlgHello =new JDialog(this, "HelloIamDialog");

■
■     public MyJFrame (String str) {
■         super(str);
■     }
■
■     void init(){
■         JButton b1=new JButton("show Dialog");
■         Container con=this.getContentPane();
■
■         // 获得面板
■         this.setSize(300,300);
■         con.setBackground(Color.red);
■         JPanel pan = new JPanel( );
■         pan.setSize(100,100);
■         pan.setBackground(Color.yellow);
■         b1.addActionListener(this);
■         pan.add(b1);
■         con.setLayout(null);
■         con.add(pan);
■         this.setVisible(true);
■     }
■
■     public void actionPerformed( ActionEvent e ){
■         dlgHello.setBounds(100, 100, 30, 100);
■         dlgHello.setVisible(true);
■
■     }
■ }
```




标准对话框

- Swing中还提供了用于显示标准对话框（JOptionPane）的JOptionPane类
- 定义了多个showXxxDialog形式的静态方法
 - ◆ showConfirmDialog —— 确认对话框，显示问题，要求用户进行确认（yes/no/cancel）
 - ◆ showInputDialog —— 输入对话框，提示用户进行输入
 - ◆ showMessageDialog —— 信息对话框，显示信息，告知用户发生了什么情况
 - ◆ showOptionDialog —— 选项对话框，显示选项，要求用户进行选择



showXxxDialog方法的参数

■ Component parentComponent

- ◆ 对话框的父窗口对象，其屏幕坐标将决定对话框的显示位置；此参数也可以为null，表示采用缺省的Frame作为父窗口，此时对话框将设置在屏幕的正中

■ String title

- ◆ 对话框的标题

■ Object message

- ◆ 显示在对话框中的描述信息。该参数通常是一个String对象，但也可以是一个图标、一个组件或者一个对象数组



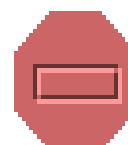
showXxxDialog方法的参数

■ int messageType

- ◆ 对话框所传递的信息类型。可以为以下常量

ERROR_MESSAGE
INFORMATION_MESSAGE
WARNING_MESSAGE
QUESTION_MESSAGE
PLAIN_MESSAGE

- ◆ 除PLAIN_MESSAGE之外，其他每种类型都对应于一个缺省图标





showXxxDialog方法的参数

■ int optionType

- ◆ 对话框上按钮的类型，可以为以下常量：

DEFAULT_OPTION

YES_NO_OPTION

YES_NO_CANCEL_OPTION

OK_CANCEL_OPTION

■ Object[] options

- ◆ 对话框上的选项。在输入对话框中，通常以组合框形式显示，在选项对话框中，则指按钮的选项类型。该参数通常是一个String数组，但也可以是图标或组件数组



showXxxDialog方法的参数

■ Icon icon

- ◆ 对话框上显示的装饰性图标，如果没有指定，则根据messageType参数显示缺省图标

■ Object initialValue

- ◆ 初始选项或输入值



实例

```
JOptionPane.showMessageDialog(frame, "File not found.",  
    "An error", JOptionPane.ERROR_MESSAGE);
```





实例

```
JOptionPane.showMessageDialog(frame, "Click OK to continue", "Warning",  
JOptionPane.DEFAULT_OPTION, JOptionPane.WARNING_MESSAGE,  
null, options, options[0]);
```





showXxxDialog()返回类型

- **showMessageDialog()**没有返回值
- **showConfirmDialog()**和**showOptionDialog()**方法返回**int**型数值，代表用户选择按钮的序号
 - ◆ （JOptionPane中定义了YES_OPTION、NO_OPTION、CANCEL_OPTION、OK_OPTION和CLOSED_OPTION等常量，分别代表用户选择了YES、NO、CANCEL、OK按钮以及未选择而直接关闭了对话框）
- **showInputDialog()**方法的返回值为**String**或**Object**，代表用户的输入或选项



JOptionPane的例子

- 例2
- 该程序创建了四个按钮和一个文本域，当用户单击某个按钮，屏幕上将会显示出对应的标准对话框，用户在确认、输入和选项对话框中的操作结果将显示在文本域中



文件对话框 (JFileChooser)

- 专门用于对文件（或目录）进行浏览和选择的对话框，常用的构造方法：
 - ◆ JFileChooser() —— 根据用户的缺省目录创建文件对话框
 - ◆ JFileChooser(File currentDirectory) —— 根据File型参数 currentDirectory指定的目录创建文件对话框
 - ◆ JFileChooser(String currentDirectoryPath) —— 根据String型参数 currentDirectoryPath指定的目录创建文件对话框



文件对话框显示

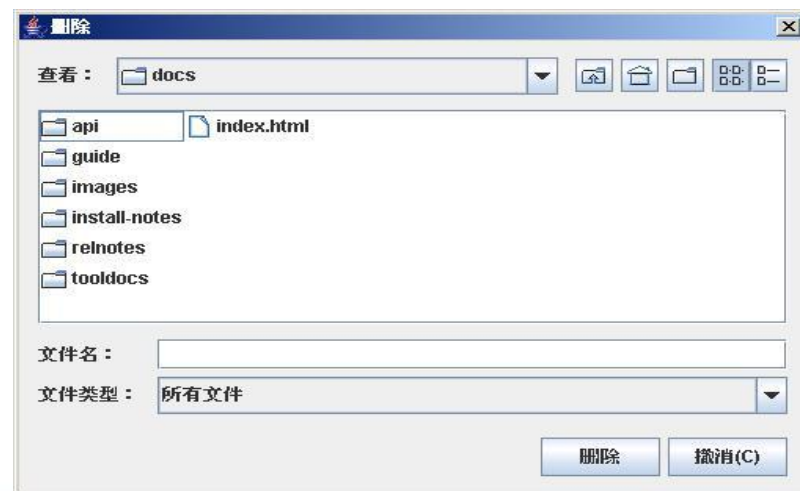
- **showOpenDialog(Component parent)**
 - ◆ 显示一个“打开”文件对话框
- **showSaveDialog(Component parent)**
 - ◆ 显示“保存”文件对话框
- **showDialog(Component parent, String approveButtonText)**
 - ◆ 显示一个自定义的文件对话框
 - ◆ 标题和按钮上的文本由String型参数approveButtonText指定
 - ◆ Component型参数，指定“父组件”，决定了文件对话框的显示位置，如果该参数为null，则文件对话框显示在屏幕正中
- 通过getSelectedFile()方法取得用户所选择的文件





文件对话框的例子

- 例3
- 当用户单击Open按钮时，“打开”文件对话框被显示出来；当用户单击Save按钮时，“保存”文件对话框被显示出来；当用户单击Delete按钮时，“删除”文件对话框被显示出来。用户进行选择之后，所选文件的路径和文件名将被显示在窗口中部的文本区内





文件对话框FileDialog类

- **Import java.awt.*;**
- **FileDialog类**

构造函数

FileDialog(Frame parent)

Creates a file dialog for loading a file.

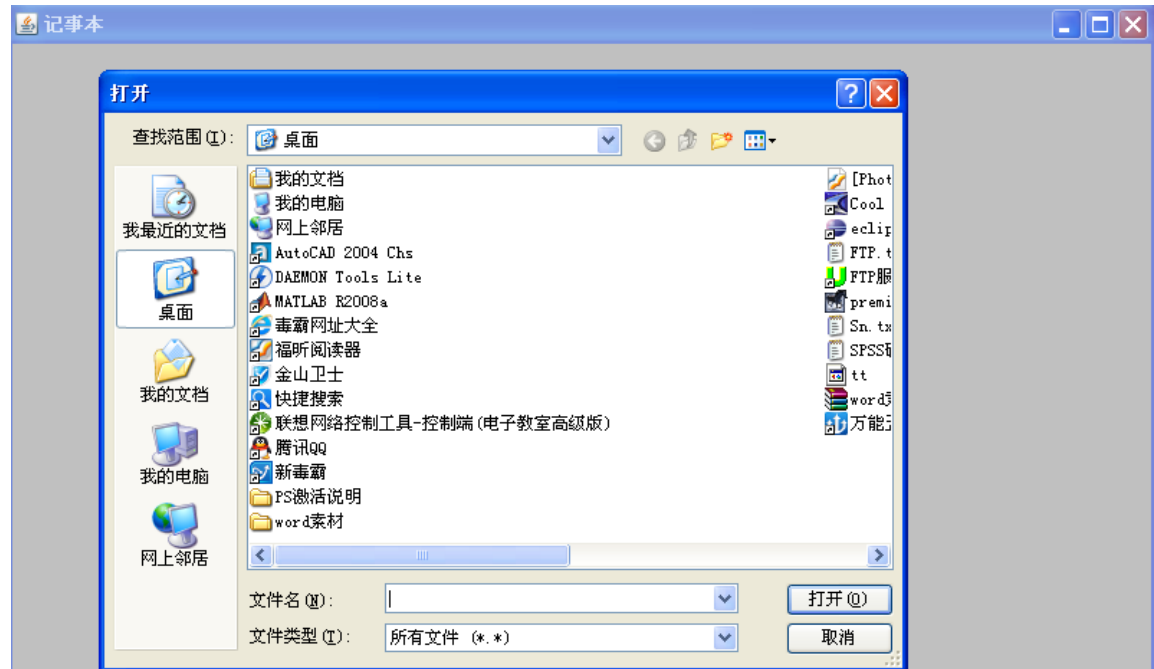
FileDialog(Frame parent, String title)

Creates a file dialog window with the specified title for loading a file.

FileDialog(Frame parent, String title, int mode)

Creates a file dialog window with the specified title for loading or saving a file.

- 举例:
- `JFrame f=new JFrame("记事本");`
- `FileDialog openFileDialog = new FileDialog(f, "打开",FileDialog.LOAD);`
- `openFileDialog.setVisible(true);`





- **FileDialog saveFileDialog = new FileDialog(f,
"保存",**
- **FileDialog.SAVE);**
- **saveFileDialog.setVisible(true);**



颜色对话框

- `javax.swing`包中的`JColorChooser`类的静态方法：
`public static Color showDialog(Component component,
String title, Color initialColor)`

创建一个颜色对话框

- 参数`component`指定对话框所依赖的组件，`title`指定对话框的标题；`initialColor` 指定对话框返回的初始颜色，即对话框消失后，返回的默认值。颜色对话框可根据用户在颜色对话框中选择的颜色返回一个颜色对象。

- 例4

