

源代码免费下载

商业开发

代码库系列



Java

案例开发集锦

(第二版)

袁 然 郑自国 来为国 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>



中国机械出版社
CHINA MACHINE PRESS

JAVA

案例开发集锦

【第二版】

清华大学出版社
Tsinghua University Press

商业开发代码库系列

Java案例开发集锦 (第二版)

袁 然 郑自国 来为国 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书沿袭第1版的风格,收录了来自实战中的六十多个完整的Java编程实例,并通过案例讲解Java的开发技巧。本书除了涵盖Java系统内建功能的开发以及与目前新技术相接合的应用开发外,还着重加强了Web应用的开发,涉及了J2EE、J2ME、安全、网络、EJB、XML等。在剖析比较案例的同时,也将各种经验性思想及理论性思想寓于其中。同时引入Java常用的设计模式,把软件架构设计理论融于案例讲解之中。读者不仅可以了解到网络中众多精彩纷呈的网络应用内幕,还可以体会到Java的强大功能,更可以借助这些案例方便地开发出自己的功能强大的应用系统,成为Java的思想者。

本书主要面向Java应用程序开发人员,同时也是Java编程爱好者的参考书,还可作为大中专院校学生学习的辅助材料。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Java案例开发集锦/袁然,郑自国,来为国编著. —2版. —北京:电子工业出版社,2008.9

(商业开发代码库系列)

ISBN 978-7-121-07112-6

I. J… II. ①袁… ②郑… ③来… III. Java语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2008)第106619号

责任编辑:戴 新

印 刷:北京天竺颖华印刷厂

装 订:三河市金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

北京市海淀区翠微东里甲2号 邮编:100036

开 本:787×1092 1/16 印张:22 字数:560千字

印 次:2008年9月第1次印刷

定 价:40.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至zltz@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

从Java诞生以来，许多人意识到Java不仅是一个了不起的概念，而且还蕴藏了了不起的思想。

有人问：“Java会灭亡吗？”

也有人说：“Java是明天的唯一吗？”

或许这些都没有答案，或许答案就在每个Java者的心中。

其实答案是什么并不重要，时间会说明一切。无论是没有可继承应用开发的缺陷，还是开放性和可移植性，都是Java不可阻挡的魅力。这些导致了Java大军越来越多，挥舞着Java的旗帜，在软件开发的高山上奋勇登攀。

本书沿袭了第1版的风格，从实际编程出发，在案例的选取上更倾向于新技术点和Web应用，也更注重知识点的应用与积累，于相同处寻找不同，于不同处寻求共性。加强了知识点的对比，着重于编程中遇到的问题处理，笔者的经验性思想也寓于案例的讲解中，以便读者更加自如地运用语言，提高Java编程的灵活性。本书共收集了62个案例，详细讲解了每个案例的实现步骤和制作要点，并提供了源代码。本书所有案例均通过调试运行，运行环境和配置在书中或者附带源码中都有说明。这些案例涉及的知识点仅仅是Java的冰山一角，笔者希望通过这些案例能使读者尽快投入到Java的实际应用中去。

本书选择的案例名称有几个与第一版中的相同，但开发的角度和内容有较大的差别，侧重的知识点也不同。本书的宗旨是通过案例的学习使读者尽快了解Java的开发思想，所以案例的实现内容并不重要，它仅仅是打开理解Java的一扇窗户，重要的是通过这扇窗，找到Java的应用方法。本书的案例强调制作要点和详细步骤的讲解，由于篇幅关系，案例中的源代码没有完全体现在书中，第10章的源代码更是被省略掉，感兴趣的读者可以到华信教育资源网下载。

全书共分10章，每一章都以Java的一个知识点或者一个应用范围为标题。

第1章 Java与Swing，收集了8个应用Swing技术的案例，从案例实现步骤的讲解中可以了解Swing丰富、灵活的功能和模块化组件，从而创建Java优雅的用户界面。

第2章 Java与线程，通过7个案例说明Java的线程和多线程的处理机制和重要性，通过线程的共享和同步可以增强程序的交互性，提供更好的GUI和更好的服务器功能，大大简化应用程序设计。

第3章 Java与I/O，7个难易程度不同的案例的输入输出流操作有助于读者了解Java的I/O体系概念，通过对字节流、文件、缓冲、管道应用的对比和共性的探讨提高I/O操作的灵活性。

第4章 Java与游戏，游戏的实现是对Java语言的综合应用，侧重于图像、声音、多线程、键盘和鼠标事件的处理，本章提供6个好玩的游戏，在轻松之余学习Java编程的方法和技巧。

第5章 Java与网络, Java是为网络而生的, 它在网络应用上的得心应手不是区区几个案例便能体现的, 希望这7个案例能带给读者一点Java在网络应用和Web上的功能的体会。

第6章 Java与数据库, Java与JDBC的结合使程序员可以只设计一次数据库应用软件后, 就能在各种数据库系统上运行。本章通过5个与不同数据库连接的案例, 突出了Java的跨平台性。

第7章 J2ME技术, 移动开发已经成为开发者社区最为引人注目的新技术, J2ME凭借其开放的特性占据了绝对的市场, 成为了移动开发领域的标准。本章通过5个案例简要介绍了J2ME的特性。

第8章 J2EE, J2EE是一套全然不同于传统应用开发的技术架构, 本章的6个案例只是对庞大的J2EE的简介而已, 希望借此能够使读者了解一点这个企业级应用开发平台的体系结构。

第9章 Web服务与其他, Java在Web服务上的便利性和扩展性, 是Java保持其魅力的原因之一, 本章提供了7个案例, 涉及到Web服务、XML、打印、安全等方面的内容。

第10章 Java综合案例, 分别是在图形界面设计、移动开发、Web应用和数据库方面的综合性的案例, 体现了Java知识在编程中的综合应用。

作为一种程序设计语言, Java有简单、面向对象、不依赖于机器的结构特点, 具有可移植性、安全性等, 并且提供了并发的机制, 具有很高的性能。本书案例中涉及了Java的多线程、异常处理、I/O流、Swing、Servlet、.NET、XML、Security、DB、File、JavaBean、Socket、JVM、MDlet、Web服务等内容, 同时也体现MVC的编程思想。

本书的分类并不严格, 每个案例都涉及到很多知识点, 笔者为了突出某一个知识点而硬性将它们归类, 不妥之处敬请读者谅解, 笔者全力想展现Java的博大精深与无穷魅力, 但这个重担不是意愿所能达到的, 期望与读者共勉, 共同提高。

本书在编写过程中得到了北京美迪亚电子信息有限公司、山东大学、山东建筑大学部分师生的大力协助及许多网友的支持, 在此表示衷心的感谢。参加本书编校的人员还有袁凯、王诚梅、陈圣琳、向小平、汤代禄、刘彬、李志勇等。由于时间仓促, 作者水平有限, 书中的错误与不足之处在所难免, 真诚希望广大读者提出宝贵意见。

谨向阅读本书的读者朋友们表示诚挚的敬意和谢意。

为了方便读者阅读, 本书配套资料请登录“华信教育资源网”(http://www.hxedu.com.cn), 在“资源下载”频道的“图书资源”栏目下载。

目 录

第1章 Java与Swing	1
案例1: 屏幕捕获工具	1
案例2: 文本阅读器	6
案例3: 简单的名片管理系统	11
案例4: 鼠标画线	17
案例5: 鼠标操作	21
案例6: 计算器程序	27
案例7: 数字时钟	32
案例8: 动画效果与颜色的控制	36
本章小结	39
第2章 Java与线程	40
案例1: 一个完整的线程池的实例	40
案例2: 鸭子凫水动画	44
案例3: 生产者-消费者模型的简单实现	49
案例4: 定时关机	53
案例5: 多线程TCP端口扫描程序	57
案例6: 一个简单的年历生成程序	62
案例7: 将GIF和JPG图像转换成VRML格式	67
本章小结	73
第3章 Java与I/O	74
案例1: 使用多线程删除指定目录及子目录下所有指定文件	74
案例2: 压缩文件	79
案例3: 解压缩Zip文件	85
案例4: 批量改名	90
案例5: 文件分割器	98
案例6: 管道流实现线程间的通信	106
案例7: 排序对象	111
本章小结	113

第4章 Java与游戏	114
案例1: Java扫雷	114
案例2: 黑白棋	122
案例3: 象棋游戏	127
案例4: 一个简单的弹球游戏	133
案例5: 找不同	139
案例6: 八皇后问题	146
本章小结	150
第5章 Java与网络	151
案例1: 简单的多线程服务器	151
案例2: 用Java实现的HTTP服务器端例程	155
案例3: 一个简单的HTML浏览器	161
案例4: 用JavaMail发送邮件	167
案例5: Java版MSN	174
案例6: Java实现HTTP队列下载	178
案例7: Java实现HTTP验证	181
本章小结	184
第6章 Java与数据库	185
案例1: Access数据库编程中查询结果的表格式输出	185
案例2: SQL Server数据库编程中查询结果的表格式输出	192
案例3: MySQL数据库编程中查询结果的表格式输出	197
案例4: Oracle OCI数据库编程	203
案例5: 网吧计费系统	206
本章小结	210
第7章 J2ME技术	211
案例1: 九宫格游戏	211
案例2: 五子棋游戏	220
案例3: 手机背单词	228
案例4: 用J2ME与ASP建立数据库连接	235
案例5: 利用J2ME开发联网程序实例	238
本章小结	244

第8章 J2EE技术	245
案例1: 一个用Servlet实现购物车的程序	245
案例2: 连接数据库的JavaBean	251
案例3: 测试安全性代码	257
案例4: 用EJB实现的用户消费信息登记系统	260
案例5: Fibonacci数列	272
案例6: 简单的图书信息管理系统	275
本章小结	285
第9章 Web服务与其他	286
案例1: 用Servlet生成图像验证码	286
案例2: 获取JVM系统属性	289
案例3: 密码生成器	293
案例4: 数据库数据转成XML文件	294
案例5: 网页计数器	297
案例6: Java打印程序	299
案例7: 用SunJCE进行文件的加密和解密	306
本章小结	312
第10章 Java综合案例	313
综合案例1: 多页面文本编辑器	313
综合案例2: “逃亡者”手机游戏	318
综合案例3: 网上CD销售系统	324
综合案例4: 航空查询订票系统	337
本章小结	343



第1章 Java与Swing

本章内容

- 案例1: 屏幕捕获工具
- 案例2: 文本阅读器
- 案例3: 简单的名片管理系统
- 案例4: 鼠标画线
- 案例5: 鼠标操作
- 案例6: 计算器程序
- 案例7: 数字时钟
- 案例8: 动画效果与颜色的控制
- 本章小结



案例1: 屏幕捕获工具

案例运行效果与操作

本案例的制作涉及Java图像处理方面的内容, 它使用Java扩展包中提供的JPEG编码器, 实现了抓取当前桌面图像并保存到指定文件夹的功能。程序运行后, 界面如图1-1所示。

单击“操作”菜单, 界面如图1-2所示。

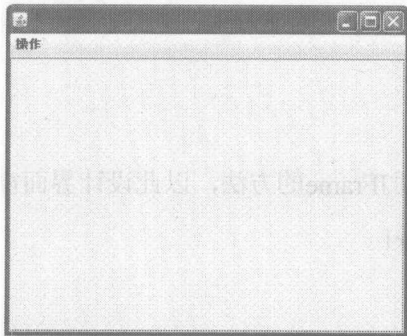


图1-1 运行界面

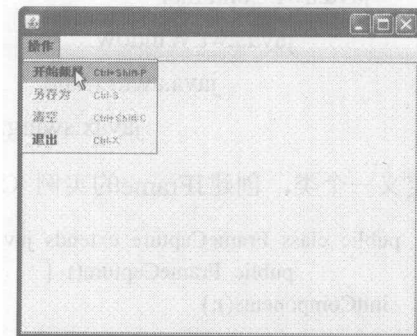
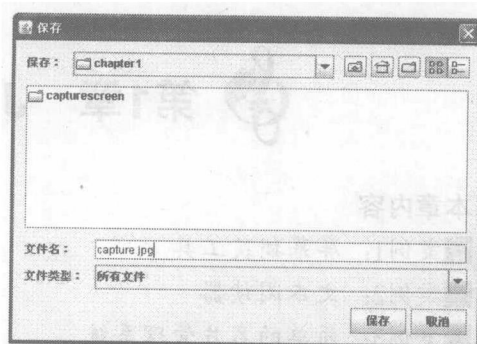
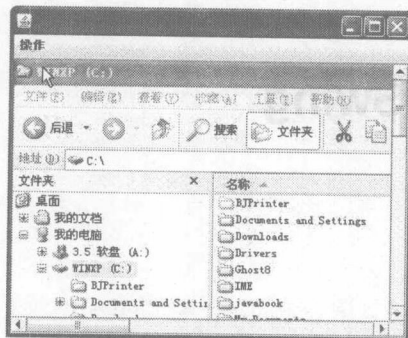


图1-2 “操作”菜单

单击“操作”菜单下的“开始截屏”菜单项, 或者按下抓图组合键Ctrl+Shift+P, 界面如图1-3所示。

按下Ctrl+S组合键保存当前窗口中的图片, 会弹出“保存”对话框, 如图1-4所示。

选择文件保存目录并填写文件名称后, 单击“保存”按钮, 完成抓取图像并保存的操作。按下Ctrl+Shift+C组合键则清空当前窗口中的内容, 返回到图1-1所示界面。按下Ctrl+X组合键则退出程序。



1. Image类的用法。
2. JPEG编码器的使用。
3. Swing菜单的应用技巧。

1. 初始化JFrame窗体：不同的开发环境操作步骤不同，本例使用NetBeans。设计程序的用户界面，笔者偏向于使用Swing，也有人喜欢用AWT，大多时候两者都用。

```
java.lang.Object
    java.awt.Component
        java.awt.Container
            java.awt.Window
                java.awt.Frame
                    javax.swing.JFrame
```

```
public class FrameCapture extends javax.swing.JFrame{
    public FrameCapture() {
        initComponents();}
}
```

2. 添加菜单控件: 添加JMenuBar控件并修改其属性, 属性修改界面如图1-6所示, 也可直接通过代码实现。

```
jMenuBar1 = new javax.swing.JMenuBar();
...
jMenuBar1.add(jMenuItem)
```

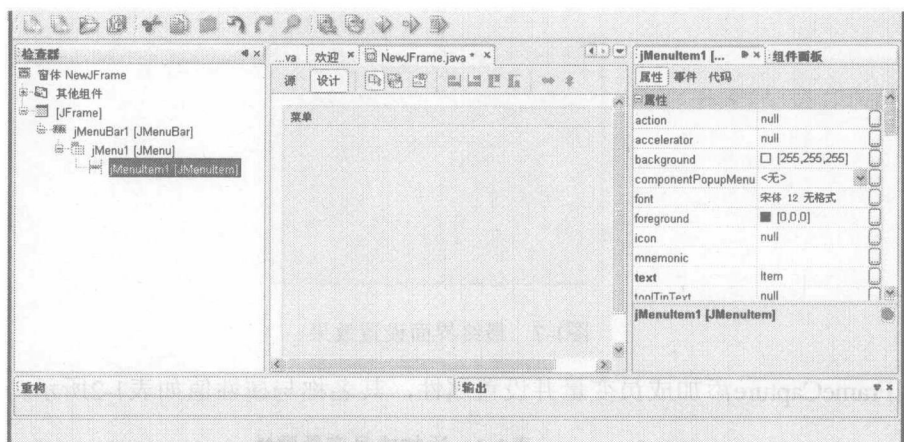


图1-5 GUI设计器

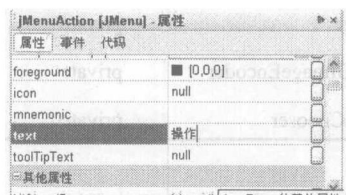


图1-6 在属性窗口中修改属性

添加JMenuItem “开始截屏”，即JMenuItemCapture并设置其属性。

```
jMenuItemCapture = new javax.swing.JMenuItem();
jMenuItemCapture.setText("\u5f00\u59cb\u622a\u5c4f");
```

3. 为JMenuItemCapture设置加速器。

```
jMenuItemCapture.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_P,
java.awt.event.InputEvent.SHIFT_MASK | java.awt.event.InputEvent.CTRL_MASK));
```

4. 用同样的方法再添加3个JMenuItem，并按照如表1-1所示修改属性。

表1-1 属性与值对照表

名称	text属性值	enable属性值	加速键
JMenuItemSaveAs	另存为	未选中	Ctrl+S
JMenuItemClear	清空	未选中	Shift+Ctrl+C
JMenuItemExit	退出	选中	Ctrl+X

5. 添加JScrollPane控件，将其名称修改为JScrollPaneGlobe。

```
jScrollPaneGlobe.setViewportViewView(jLabelShow)
```

6. 在JScrollPaneGlobe中添加一个JLabel，将其名称修改为JLabelShow，并将其text属性的值设置为空。

```
this.jLabelShow.setIcon(null);
```

7. 调整各个控件实例在窗体上的位置与大小，最终界面设置效果如图1-7所示。

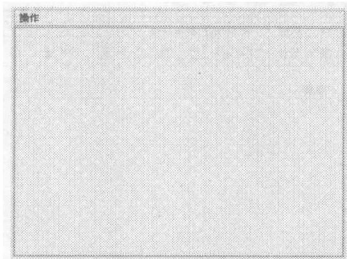


图1-7 最终界面设置效果

8. 向FrameCapture添加成员变量并设置属性，其名称与属性值如表1-2所示。

表1-2 添加成员变量属性

变量名称	变量类型	访问限制修饰符	初始值
tempImage	Image	private	null
encoder	JPEGImageEncoder	private	null
fileChooser	JFileChooser	private	new JFileChooser()

9. 添加createImage()方法及其代码，其返回类型为Image。createImage()是截图的方法，主要功能是获取屏幕大小、截取指定矩形区域内图像并返回。

10. 向FrameCapture添加saveImage()方法。saveImage()实现文件的保存输出，把输出流连接到编码器。

```
String fileName=new String(saveFileName.getPath()+".jpg");    //获取要保存文件的名字
FileOutputStream fileOutPut=new FileOutputStream(fileName);    //首先创建一个输出流
    //然后把输出流用JPEG编码器进行包裹，其实就是把输出流连接到编码器
encoder=JPEGCodec.createJPEGEncoder(fileOutPut);
encoder.encode((BufferedImage) tempImage);    //把BufferedImage对象进行编码
```

11. 为JMenuItemCapture菜单项添加ActionEvent事件的处理方法。

```
Image captureImage=this.createImage();    //截取当前屏幕图像
//将图像在JLabelShow中显示
jLabelShow.setIcon(new ImageIcon(tempImage=this.createImage()));
this.jMenuItemSaveAs.setEnabled(true);
this.jMenuItemClear.setEnabled(true);
this.jMenuItemCapture.setEnabled(false);
```

12. 为JMenuItemClear菜单项添加ActionEvent事件的处理方法。

```
this.jLabelShow.setIcon(null);    //将JLabelShow中的图像清空
this.jMenuItemSaveAs.setEnabled(false);
this.jMenuItemClear.setEnabled(false);
this.jMenuItemCapture.setEnabled(true);
```

程序源代码与解释

```
/** FrameCapture.java*/
public class FrameCapture extends javax.swing.JFrame {
```

```

    public FrameCapture() {
        initComponents();
    }
    /**在构造器内调用的方法，用于初始化表格    */
    private void initComponents() {
        //...代码略，请参见本书附带的代码，下同
    }
    private void jMenuItemClearActionPerformed(java.awt.event.ActionEvent evt) {
        //参见步骤详解12...    //将JLabelShow中的图像清空
    }
    private void jMenuItemSaveAsActionPerformed(java.awt.event.ActionEvent evt) {
        this.saveImage();    //调用saveImage()方法保存当前屏幕图像
    }
    private void jMenuItemCaptureActionPerformed(java.awt.event.ActionEvent evt) {
        //截取当前屏幕图像，将图像在JLabelShow中显示
        //参见步骤详解11
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new FrameCapture().setVisible(true);
            }
        });
    }
    private JPEGImageEncoder encoder = null;
    private JFileChooser fileChooser = new JFileChooser();
    private Image tempImage = null;
    private Image createImage() {
        try { //截图代码开始
            Image tempLocalImage=null;
            Robot robot=new Robot();
            Dimension dimension= Toolkit.getDefaultToolkit().getScreenSize();
            Rectangle scrRect=new Rectangle(0,0,dimension.width,dimension.height);
            tempLocalImage=robot.createScreenCapture(scrRect);
            return tempLocalImage;    //截图代码结束
        } catch (AWTException ex) {
            ex.printStackTrace();
        }
        return null;}
    public void saveImage() {
        try{
            int saved=fileChooser.showSaveDialog(this);
            if(saved==JFileChooser.APPROVE_OPTION) {
                File saveFileName=fileChooser.getSelectedFile();    //获取要设置的文件名（包括
                //参见步骤详解10
                fileOutPut.flush();
                fileOutPut.close();
            }
            catch(Exception ea) { ea.printStackTrace();}
        }
    }
}

```

路径)



案例2：文本阅读器

案例运行效果与操作

本案例是关于Java复合对话框方面的内容，它使用Swing中提供的各种对话框和文件过滤器等控件，实现针对用户不同操作出现不同的与用户交互对话框的功能。程序运行后，界面如图1-8所示。

单击“浏览”按钮，会弹出“打开”对话框，界面如图1-9所示。

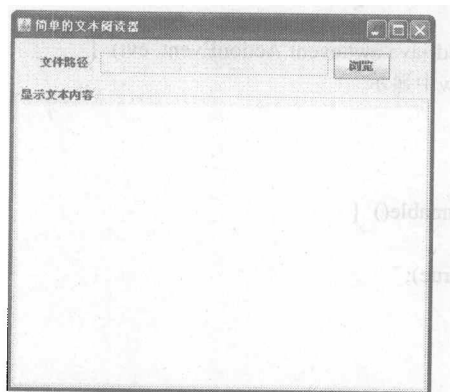


图1-8 运行界面

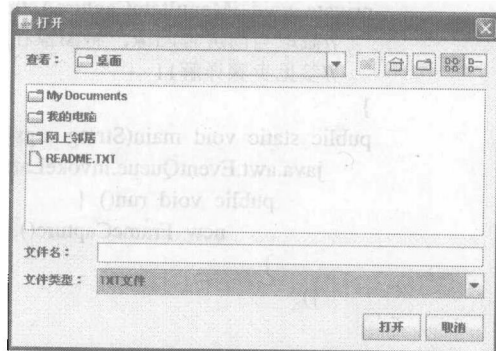


图1-9 “打开”对话框

如果未选择文件而单击“取消”按钮，则弹出提示对话框，界面如图1-10所示。
如果选择文件，并单击“打开”按钮，会弹出确认对话框，如图1-11所示。

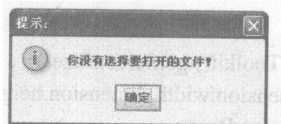


图1-10 未选择文件

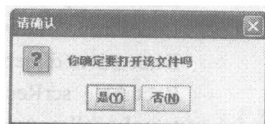


图1-11 确认打开文件

此时如果选择“否”，会弹出确认对话框，如图1-12所示。
选择“是”，则显示相应文件内容，如图1-13所示。

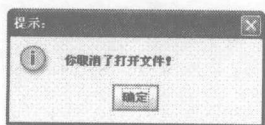


图1-12 取消打开

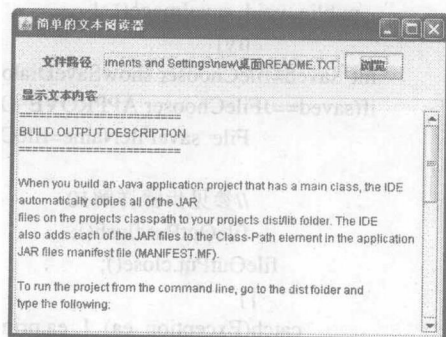


图1-13 显示文件内容

另外，通过文件过滤器的应用，使得该程序只能打开文本文件或者Java文件，如图1-14所示。

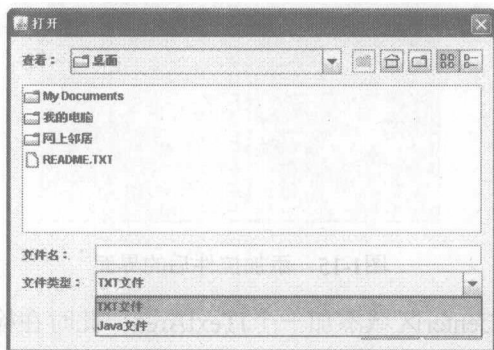


图1-14 只能打开文本文件或者Java文件

制作要点

1. Swing对话框的应用技巧。
2. JOptionPane类的使用方法。
3. 文件过滤器的使用。
4. 输入/输出流的使用。

步骤详解

1. JFrame设置：选择“设置布局” / “BorderLayout”选项，将FrameReader的布局管理器设置为BorderLayout。

```
public class FrameReader extends javax.swing.JFrame{
    ...
    initComponents();
    this.initFileChooser();
}
```

2. 添加JPanel控件，向窗体中添加一个JPanel控件JPanelNorth，属性设置为North。

```
jPanelNorth = new javax.swing.JPanel();
...
getContentPane().add(jPanelNorth, java.awt.BorderLayout.NORTH);
```

3. 向JPanelNorth中分别添加一个JLabel、JTextField和JButton，并按照表1-3所示修改属性。此时GUI设计器中的界面如图1-15所示。

表1-3 控件属性与值对照表

控件类型	控件名称	text属性值
JLabel	JLabelFilePath	文件路径
JTextField	JTextFieldFilePath	空
JButton	JButtonOpen	浏览

4. 在JTextFieldFilePath属性窗口中，将editable属性设置为非选中状态。

```
jTextFieldFilePath.setEditable(false)
```

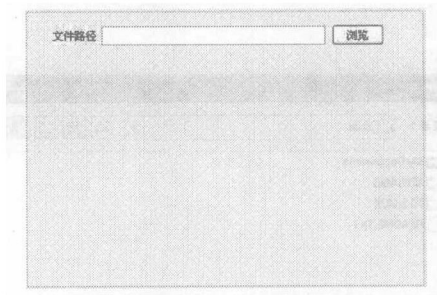


图1-15 添加控件后的界面

5. 在FrameReader的Center区域添加一个JTextArea, 此时在检查器窗口中会自动添加一个JScrollPane节点。

```
jTextAreaContent = new javax.swing.JTextArea();
```

JTextAreaContent和JScrollPaneCenter设置如下:

```
jTextAreaContent = new javax.swing.JTextArea()  
jScrollPaneCenter = new javax.swing.JScrollPane()
```

6. 调整各个控件实例在窗体上的位置与大小, 界面最终设置效果如图1-16所示。

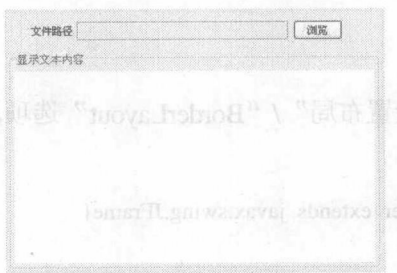


图1-16 界面最终设置效果

7. Swing中提供了JOptionPane类来实现类似Windows平台下的MessageBox的功能, 本例中复合对话框的实现使用了Java中的JOptionPane类。JOptionPane类中的方法可用各种标准的对话框实现显示信息、提出问题、警告、用户输入参数等功能。这些对话框都是模式对话框。其中常用的如下:

- **ConfirmDialog**——确认对话框, 提出问题, 然后由用户自己来确认 (单击 “Yes” 或 “No” 按钮)。
- **InputDialog**——提示输入文本。
- **MessageDialog**——显示信息。
- **OptionDialog**——组合其他三个对话框类型。

这四个对话框可以采用showXXXXDialog()来显示, 如showConfirmDialog()显示确认对话框、showInputDialog()显示输入文本对话框、showMessageDialog()显示信息对话框、showOptionDialog()显示选择性的对话框。本例中显示信息对话框:

```
JOptionPane.showMessageDialog(this,"你没有选择要打开的文件!", "提示: ",JOptionPane.  
.INFORMATION_MESSAGE)
```