

# 基于Java的记事本程序的设计与实现

张会

(攀枝花学院 数学与计算机学院, 四川 攀枝花 617000)

**摘要:**该文利用Java Swing, 输入输出流, 事件处理等技术实现了一个记事本小程序。记事本具有新建, 保存, 另存为, 设置字体, 设置颜色及退出等功能。记事本是一个集Java Swing、输入输出流、事件处理等知识综合应用的实例程序。

**关键词:**输入输出流; 事件处理; 记事本; Java Swing

**中图分类号:** TP311 **文献标识码:** A **文章编号:** 1009-3044(2017)30-0088-03

**DOI:** 10.14004/j.cnki.ckt.2017.3343

Notepad(记事本)是Windows系统中一款小巧的纯文本编辑器, 常用做源程序编辑器, 具有打开速度快, 文件小等特点, 若只保存为纯文本文件最好使用记事本。文中的记事本采用Java Swing, 输入输出流、事件处理等相关技术进行设计及开发, 使初学者快速掌握及使用Java中的Java Swing、输入输出流、事件处理等相关技术进行Java程序开发有较大帮助, 并对《Java程序设计》课程相关知识点的综合实践教学具有一定指导作用。

## 1 记事本设计

### 1.1 功能设计

记事本能够实现以下功能:

1) 保存: 对于新建文件, 单击保存选项, 则弹出保存对话框, 实现对记事本中的内容命名并保存; 对于已命名文件, 单击保存选项, 不会弹出保存对话框, 直接实现原名保存。

2) 另存为: 实现换名或换位置保存。

3) 打开: 实现打开一个纯文本文件, 如\*.txt, \*.cpp, \*.java等纯文本文件。

4) 保存提示: 对新建的文件输入了内容或对打开的文件作了修改并直接退出记事本时, 会提示是否保存修改。

5) 设置字体颜色: Windows系统中的记事本, 未实现对文本内容设置字体颜色功能, 本程序实现了对记事本中的文字设置字体颜色。

6) 设置字体、字形、字号: 实现了对文本设置字体、字形、字号功能。

### 1.2 界面设计

Swing是Java的GUI(图形用户界面)工具包, 包括了GUI组件, 如: 菜单栏、菜单项、按钮、面板、文本框等组件。本设计利用了Swing中的相关组件实现了如图1, 图2所示的记事本。其中使用JFrame做为顶层窗体容器, 在其上添加了JMenuBar(菜单栏), JMenu(记事本菜单)及菜单下的JMenuItem(菜单项)组件, 另外在JFrame容器中添加了一个JPanel面板, 再在JPanel中添加了一个JTextArea组件, JTextArea对象用来显示或编辑记事本文件中的内容。



图1 记事本界面1

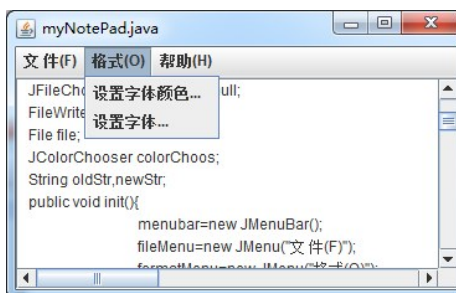


图2 记事本界面2

## 2 记事功能实现

### 2.1 记事各功能事件处理实现

本记事本具有打开文件, 新建文件, 保存、另存为、字体设置、颜色设置等功能, 当鼠标单击这些选项时, 将会对应不同的功能操作。这些操作都是通过Java中的事件处理技术实现。当单击菜单中某菜单选项时, 会产生一个ActionEvent事件, 然后Java虚拟机检查是否有与该菜单项相关联的事件处理器, 事件处理器实质是一个方法, 如actionPerformed()方法, 如果没有, 不做任何处理; 若有则将事件对象传递给与之关联的事件处理器方法作为事件处理方法参数, 并自动调用事件处理方法。Java虚拟机通过传递过来的ActionEvent对象, 获得事件发生时与该事件及事件源相关联的信息。Java组件通过形如addXxxListener()的方法添加事件监听器。

一个事件源可以添加多个监听器, 而本记事本中的事件源

收稿日期: 2017-09-02

作者简介: 张会(1980—), 女, 四川射洪人, 副教授, 硕士, 主要研究方向为计算机应用, 应用程序开发。

是菜单中的各菜单项,在菜单项上发生鼠标单击事件,因此在每个菜单项上添加addActionListener()方法添加事件监听器,并实现事件处理actionPerformed()方法给出具体的功能实现。

为记事本中各菜单项添加事件监听器代码如下:

```
newfile.addActionListener(this);
open.addActionListener(this);
save.addActionListener(this);
exit.addActionListener(this);
saveas.addActionListener(this);
fontColor.addActionListener(this);
font.addActionListener(this);
事件处理 actionPerformed()方法代码如下:
public void actionPerformed(ActionEvent e) {
    JMenuItem item=(JMenuItem)e.getSource();//获得事件源
    if(item==open){
        openFile(); //打开文件
    }else if(item==save){
        saveFile(); //保存文件
    }else if(item==saveas){
        saveasFile(); //另存为文件
    }else if(item==newfile){
        newFile(); //新建文件
    }else if(item==exit){
        exitNotepad(); //退出记事本
    }else if(item==fontColor){
        setFontColor(); //字体颜色设置
    }else if(item==font){
        setFont(); //字体设置
    }
}
```

## 2.2 打开文件

实现记事本的打开功能,实质是把要打开的文件内容读入到内存再显示在记事本中。在Java中,是通过流来实现数据的输入输出。Java中的流有两种类型,分别是字节流和字符流,其中字符流只处理文本化的数据,而本记事本程序读写都是文本文件,因此本程序输入和输出操作都是针对字符数据流操作,而FileReader,FileWriter是用来读写字符流的主要类。因此本程序使用FileReader,FileWrite两个类提供的read(),write()方法实现文件打开和保存操作。

打开文件代码如下:

```
public void openFile(){
    area.setText(null);
    fileChooser=new JFileChooser();//创建一个不可见的文件选择对话框
    //显示文件选择对话框并确定打开所选择文件
    if(fileChooser.showOpenDialog(this) ==JFileChooser.APPROVE_OPTION){
        file=fileChooser.getSelectedFile();
        char content[]=new char[30];
        int length;
        try {reader=new FileReader(file);
            while((length=reader.read(content))!=-1){
                String str=new String(content, 0, length);
                this.textArea.append(str);
            }
        }
    }
}
```

```
}
reader.close();
this.setTitle(file.getName());
oldStr=textArea.getText();
} catch (Exception e) { e.printStackTrace();}
}
```

在实现打开功能时,通过JFileChooser类的showOpenDialog()方法显示打开文件对话框,当选择了要打开的文件并确认打开操作后,该方法的返回值与JFileChooser.APPROVE\_OPTION相等时,使用JFileChooser对象的getSelectedFile()获得打开文件file对象,再通过FileReader对象与打开的文件关联,建立输入流对象reader,通过输入流对象reader调用read()方法循环读入文件内容至content数组中,再把数组内容追加在记事本中用于显示和编辑的textArea组件中。从而达到把要打开的文件内容显示在记事本中。

## 2.3 另存为功能

当单击另存为选项时,通过JFileChooser类的showSaveDialog()方法来显示保存文件对话框,当设置了文件保存位置、文件名并确认保存操作后,该方法的返回值与JFileChooser.APPROVE\_OPTION相等,使用JFileChooser对象的getSelectedFile()获得保存文件对象file,再通过FileWriter类对象与保存的文件关联建立输出流对象writer,通过输出流对象writer调用write()方法将textArea.getText()中需要保存的所有文本内容,写到输出流对象writer关联的文件中,从而实现文件的另存为功能。

```
public void saveasFile(){
    fileChooser=new JFileChooser();
    if(fileChooser.showSaveDialog(this) ==JFileChooser.APPROVE_OPTION){
        file=fileChooser.getSelectedFile();
        String content=textArea.getText();
        try {
            writer=new FileWriter(file);
            writer.write(content);
            writer.close();
            this.setTitle(file.getName());
            oldStr=textArea.getText();
        } catch (IOException e) {e.printStackTrace();}
    }
}
```

## 2.4 保存功能

保存功能类似于另存为功能,但其包含两种情况,第一种情况是文件不存在即指新文件第一次保存时,需要调用上节所实现的另存为功能对应的saveasFile()方法完成文件位置选择及文件命名保存;第二种情况是对已存在文件进行保存操作,此时只需要把新增内容写入原文件中保存即可。

保存功能代码如下:

```
public void saveFile(){
    if(file!=null){//旧文件保存时,把textArea中的内容直接写到原文件中即可
        String text=textArea.getText();
        char content[]=text.toCharArray();
        try {
            writer=new FileWriter(file);
        }
    }
}
```

```

writer.write(text);
writer.close();
this.setTitle(file.getName());
oldStr=textArea.getText();
} catch (IOException e) { e.printStackTrace();}
}else{
    saveasFile();} //新文件第一次保存,会弹出保存对话框,进行命名

```

## 2.5 新建文件功能

只要对文本编辑区里的内容做了修改,选择文件菜单下的新建功能时,会提示是否保存文件的修改内容。如果需要保存,则调上节实现的保存功能对应的 saveFile() 方法来处理新文件保存或者原名原位置文件保存;如果放弃保存,则直接新建文件。新建文件实质是把 file 对象置为 null,同时把 textArea 文本区内容置为 null。

新建功能代码如下:

```

public void newFile() {
    newStr = textArea.getText();
    if (!oldStr.trim().equals(newStr.trim())){//文本有修改,并保存修改
        int status = JOptionPane.showConfirmDialog(this, "是否保存文件的修改?", "确认修改", JOptionPane.YES_NO_CANCEL_OPTION);
        if (status == JOptionPane.YES_OPTION) {
            saveFile();
            file = null;
            textArea.setText(null);
            this.setTitle("新建文件");
        }
    }
}

```

## 2.6 设置字体颜色

本文使用 Java 类库所提供的 JColorChooser 来创建一个颜色选择器对象,并利用该对象调用 showDialog 来显示颜色选择对话框并返回选择的 Color 对象,再通过 JTextArea 对象的 setForeground(color) 将记事本的文本区设置成所选颜色。实现现代代码如下:

```

public void setFontColor(){
    colorChoos=new JColorChooser();
    Color color= colorChoos.showDialog(this, "调色板", Color.black);
}

```

```

textArea.setForeground(color);

```

## 2.7 字体设置功能

代码中的 FontChooser 是自定义的一个字体选择器类,用来仿 Windows 记事本中的字体选择控件,FontChooser 类的定义和操作与文件选择器 JFileChooser 基本相同,由于篇幅关系,FontChooser 类的实现在此不再赘述。利用自定义 FontChooser 类设置记事本中字形,字号,字体选择设置代码如下:

```

public void setFont(){
    FontChooser fontChooser=new FontChooser(textArea.getFont()); //构造选择器对象
    int returnValue = fontChooser.showFontDialog(this); // 打开字体选择器对话框
    if (returnValue == FontChooser.APPROVE_OPTION) { //按下确定按钮
        Font font = fontChooser.getSelectedFont(); // 获取选择的字体
        textArea.setFont(font); // 将 textArea 中文字设置为选中字体
    }
}

```

## 3 结束语

本记事本通过使用 Java Swing 呈现出友好的人机交互界面,采用字符输入输出流对象实现文件打开,保存功能,采用事件处理技术实现各菜单项相应操作功能,另外通过 JColorChooser 类及自定义 FontChooser 类实现了文本字体颜色,字体设置等功能。文中所实现的记事本与 Windows 系统所提供的记事本功能大致相同,但扩展并实现了 Windows 记事本没有的字体颜色设置功能。本程序为 Java 初学者掌握事件处理,输入输出流,Java Swing 等知识提供了一个综合而有用的实例,便于初学者理解及掌握,并且对《Java 程序设计》课程教学有一定的参考作用。

## 参考文献:

- [1] 王振飞,孙媛. Java 语言程序设计[M]. 广州: 华南理工大学出版社,2015.
- [2] 传智博客. Java 基础入门[M]. 北京: 清华大学出版社,2014.
- [3] 欧阳桂秀. Java 中几个文件输入流和文件输出流的研究[J]. 福建电脑,2014(10):117-118.
- [4] 马巧梅. 基于 Java 打砖块游戏的设计与实现[J]. 信息技术,2016(7):16-19.
- [5] 孙佳美. 基于 Java 飞机大战的设计与实现[J]. 无线互联科技,2016(1):54-55.