

java简单案例

- 1 目前java的主要应用方向是后端开发，当然大数据什么的都是可以做的(大家以为python是搞大数据的，其实java才是鼻祖)
- 2
- 3
- 4
- 5 今天带来用java开发的一个实用性的小工具

使用java的swing包和IO包 编写一个文件分割的可视化程序

```
1 package edu.njau.demo;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7 import java.awt.event.FocusEvent;
8 import java.awt.event.FocusListener;
9 import java.io.*;
10
11 public class FileSplitTool extends JFrame implements ActionListener,
    FocusListener {
12
13     //定义窗口所需组件
14     private JFileChooser jfc;
15     private JButton jButton;
16     private JButton jButton1;
17     private JTextField jTextField1;
18
19
20
21     //定义一些变量
22     private File selectFile=null;
23
24     public FileSplitTool(){
25
26
27
28         //初始化组件
29         jfc=new JFileChooser();
30         jButton=new JButton("选择文件");
31         jTextField1=new JTextField();
32         jTextField1.setText("请输入需要分割数目");
33         jTextField1.setForeground(new Color(204,204,204));
34
35
36         jButton1=new JButton("进行分割");
37
```

```

38
39
40 //添加组件事件
41 jButton.addActionListener(this);
42 jButton1.addActionListener(this);
43 jTextField1.addFocusListener(this);
44
45
46
47 //大面板添加内部组件
48 this.setLayout(new GridLayout(1,4));
49 this.add(jButton);
50 this.add(jTextField1);
51 this.add(jButton1);
52
53
54
55 //设置大面板属性
56 this.setSize(1200, 300);
57 this.setTitle("文件分割GUI");
58 this.setVisible(true);
59 this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
60 }
61 public static void main(String[] args) {
62
63     new FileSplitTool();
64 }
65
66
67 public static void splitFile(File file,int splitNum) throws
IOException {
68
69     String fileName = file.getName();
70     System.out.println(fileName);
71     //获得文件名和文件扩展名
72     String[] splits = fileName.split("\\.");
73     String pre=splits[0];
74     String suf=splits[1];
75
76
77     //获得文件父目录名
78     String parentPath = file.getParentFile().getAbsolutePath();
79     System.out.println(file.length());
80     int splitSize= (int) (file.length()/splitNum);
81     byte [] bytes=new byte[splitSize];
82
83
84     BufferedInputStream bfin=new BufferedInputStream(new
FileInputStream(file));
85     int i=1;
86     while(true)
87     {
88
89         File splitFile=new File(parentPath+"\\ "+pre+"-"+i+"."+suf);
90         int length=bfin.read(bytes);
91         if(length!=-1)
92         {
93             System.out.println(splitFile.getName());

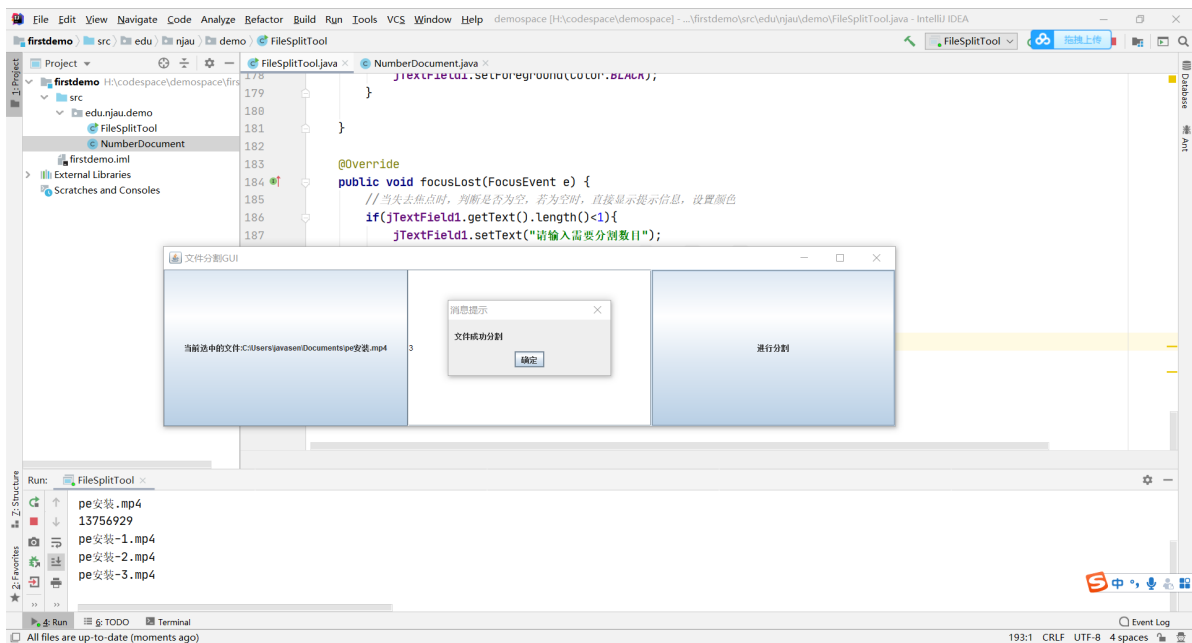
```



```

147         "文件成功分割",
148         "消息提示",
149         JOptionPane.PLAIN_MESSAGE);
150     } catch (NumberFormatException ex) {
151         JOptionPane.showMessageDialog(this, "请输入分割的数
152         目", "!!", JOptionPane.WARNING_MESSAGE);
153     }
154
155     } catch (FileNotFoundException ex) {
156         ex.printStackTrace();
157     } catch (IOException ex) {
158         ex.printStackTrace();
159     }
160
161     }
162     else{
163         JOptionPane.showMessageDialog(this, "请选择文件", "没有选择文
164         件!!", JOptionPane.WARNING_MESSAGE);
165     }
166
167     }
168
169
170
171     }
172
173     @Override
174     public void focusGained(FocusEvent e) {
175         //当点击输入框时，里面的内容为提示信息时，清空内容，将其字体颜色设置为正常黑色。
176         if(jTextField1.getText().equals("请输入需要分割数目")){
177             jTextField1.setText("");
178             jTextField1.setForeground(Color.BLACK);
179         }
180
181     }
182
183     @Override
184     public void focusLost(FocusEvent e) {
185         //当失去焦点时，判断是否为空，若为空时，直接显示提示信息，设置颜色
186         if(jTextField1.getText().length()<1){
187             jTextField1.setText("请输入需要分割数目");
188             jTextField1.setForeground(new Color(204,204,204));
189         }
190
191     }
192 }
193

```



一些要点

1. **JFrame**类的默认布局是**BorderLayout**(边界布局)
- 这种布局管理器分为东、南、西、北、中心五个方位。北和南的组件可以在水平方向上拉伸；而东和西的组件可以在垂直方向上拉伸；中心的组件可同时在水平和垂直方向上同时拉伸，从而填充所有剩余空间。在使用**BorderLayout**的时候，如果容器的大小发生变化，其变化规律为：组件的相对位置不变，大小发生变化。例如容器变高了，则**North**、**South** 区域不变，**West**、**Center**、**East**区域变高；如果容器变宽了，**West**、**East**区域不变，**North**、**Center**、**South**区域变宽。不一定所有的区域都有组件，如果四周区域（**West**、**East**、**North**、**South**区域）没有组件，则由**Center**区域去补充，但是如果 **Center**区域没有组件，则保持空白。
2. 事件监听
- 按钮，文本框等组件是事件源，这些事件源可以添加事件监听者，事件监听者就是一些实现了特点接口的**java**类
- 当事件源添加了事件监听者后，事件监听者监听到事件源发生的相应事件就会执行对应的代码
- 一个监听者可以监听多个事件源
- 代码中 **jButton**添加事件监听者 **this**， **this**就是指 **FilesplitTool**类产生的对象， **FilesplitTool**类实现了 **ActionListener**接口可以作为事件监听者，他可以监听到 **jButton**的点击事件，就会执行相应的代码
- jButton.addActionListener(this);**
3. **IO**包
- IO**就是**input output** 即输入 输出
- 这里使用**java**的**InputStream**和**OutputStream**

```
22  InputStream作用就是将硬盘中的数据读取到内存中，所谓的读取到内存中就存放在了一个java的变  
    量中，因为变量是在内存中的  
23  OutputStream作用就是将内存中的数据写入到硬盘中，也就是将java变量中存放的数据写出  
24  代码中使用byte数组来存放从硬盘读入的数据  
25  
26
```