**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

2

package edu.njau.demo;

3

import javax.swing.\*;

4

import java.awt.\*;

5

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.FocusEvent;

import java.awt.event.FocusListener;

import java.io.\*;

6

7

8

9

10

11 public class FileSplitTool extends JFrame implements ActionListener,

FocusListener {

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

//定义窗口所需组件

private JFileChooser jfc;

private JButton jButton;

private JButton jButton1;

private JTextField jTextField1;

//定义一些变量

private File selectFile=null;

public FileSplitTool(){

//初始化组件

jfc=new JFileChooser();

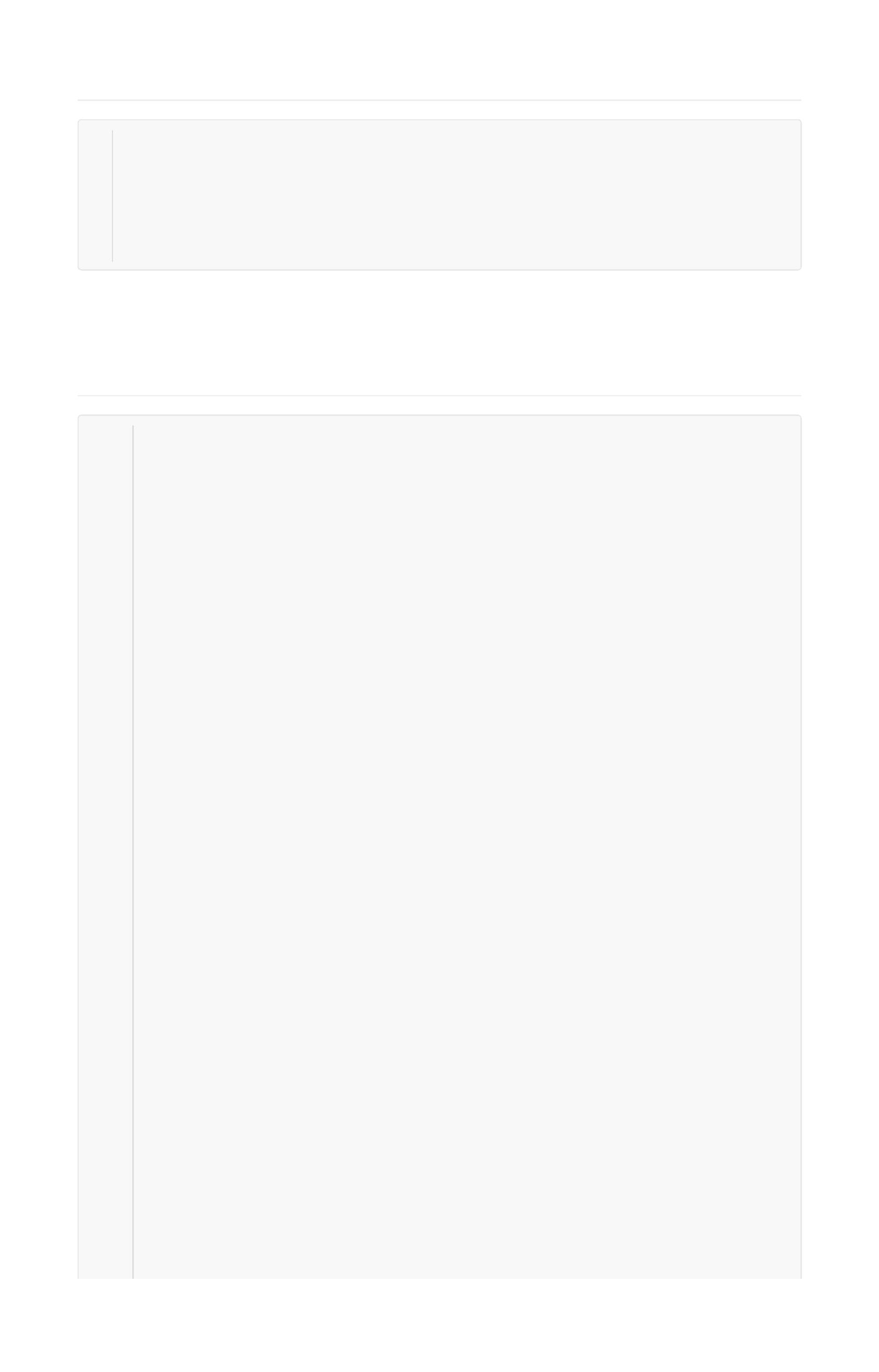
jButton=new JButton("选择文件");

jTextField1=new JTextField();

jTextField1.setText("请输入需要分割数目");

jTextField1.setForeground(new Color(204,204,204));

jButton1=new JButton("进行分割");



38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

//添加组件事件

jButton.addActionListener(this);

jButton1.addActionListener(this);

jTextField1.addFocusListener(this);

//大面板添加内部组件

this.setLayout(new GridLayout(1,4));

this.add(jButton);

this.add(jTextField1);

this.add(jButton1);

//设置大面板属性

this.setSize(1200, 300);

this.setTitle("文件分割GUI");

this.setVisible(true);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public static void main(String[] args) {

new FileSplitTool();

}

public static void splitFile(File file,int splitNum) throws

IOException {

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

String fileName = file.getName();

System.out.println(fileName);

//获得文件名和文件扩展名

String[] splits = fileName.split("\\.");

String pre=splits[0];

String suf=splits[1];

//获得文件父目录名

String parentPath = file.getParentFile().getAbsolutePath();

System.out.println(file.length());

int splitSize= (int) (file.length()/splitNum);

byte [] bytes=new byte[splitSize];

BufferedInputStream bfin=new BufferedInputStream(new

FileInputStream(file));

85

86

87

88

89

90

91

92

93

int i=1;

while(true)

{

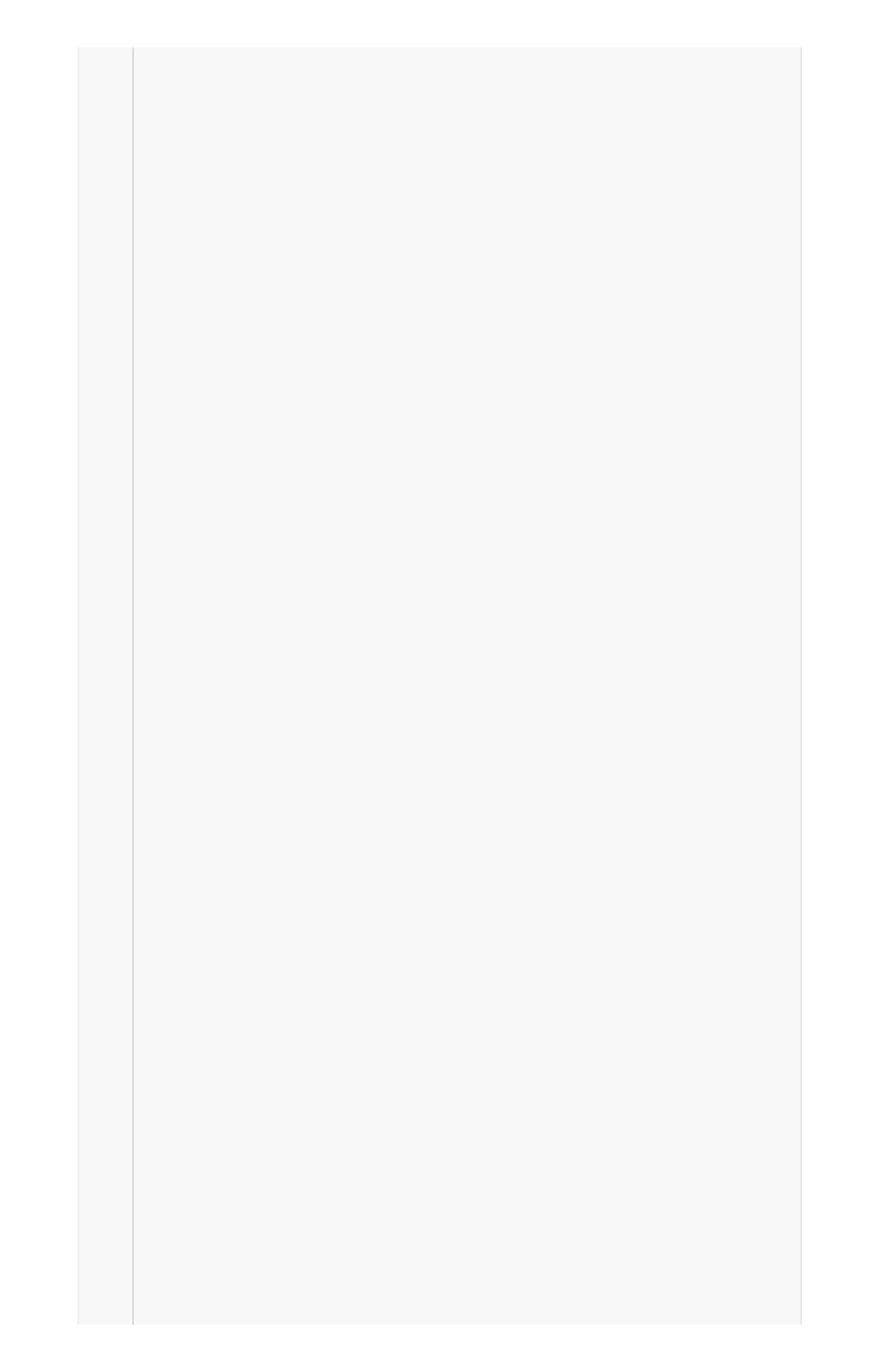
File splitFile=new File(parentPath+"\\"+pre+"-"+i+"."+suf);

int length=bfin.read(bytes);

if(length!=-1)

{

System.out.println(splitFile.getName());



94

BufferedOutputStream bfout=new BufferedOutputStream(new

FileOutputStream(splitFile));

95

96

bfout.write(bytes,0,length);

bfout.close();

97

}

98

else

{

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

break;

}

i++;

}

bfin.close();

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==this.jButton)

{

System.out.println(e);

jfc.showOpenDialog(this);

File file = jfc.getSelectedFile();//获取打开的文件

if(file!=null)

{

System.out.println(file.getAbsolutePath());

this.jButton.setText("当前选中的文

件:"+file.getAbsolutePath());

System.out.println(

128

file.getParentFile().getAbsolutePath());

this.selectFile=file;

129

130

131

132

133

134

135

136

137

138

139

}

}

else if(e.getSource()==this.jButton1)

{

if(selectFile!=null)

{

try {

BufferedInputStream bf=new BufferedInputStream(new

FileInputStream(selectFile));

System.out.println( bf.available());

140

141

142

143

try {

int

splitNum=Integer.valueOf(this.jTextField1.getText());

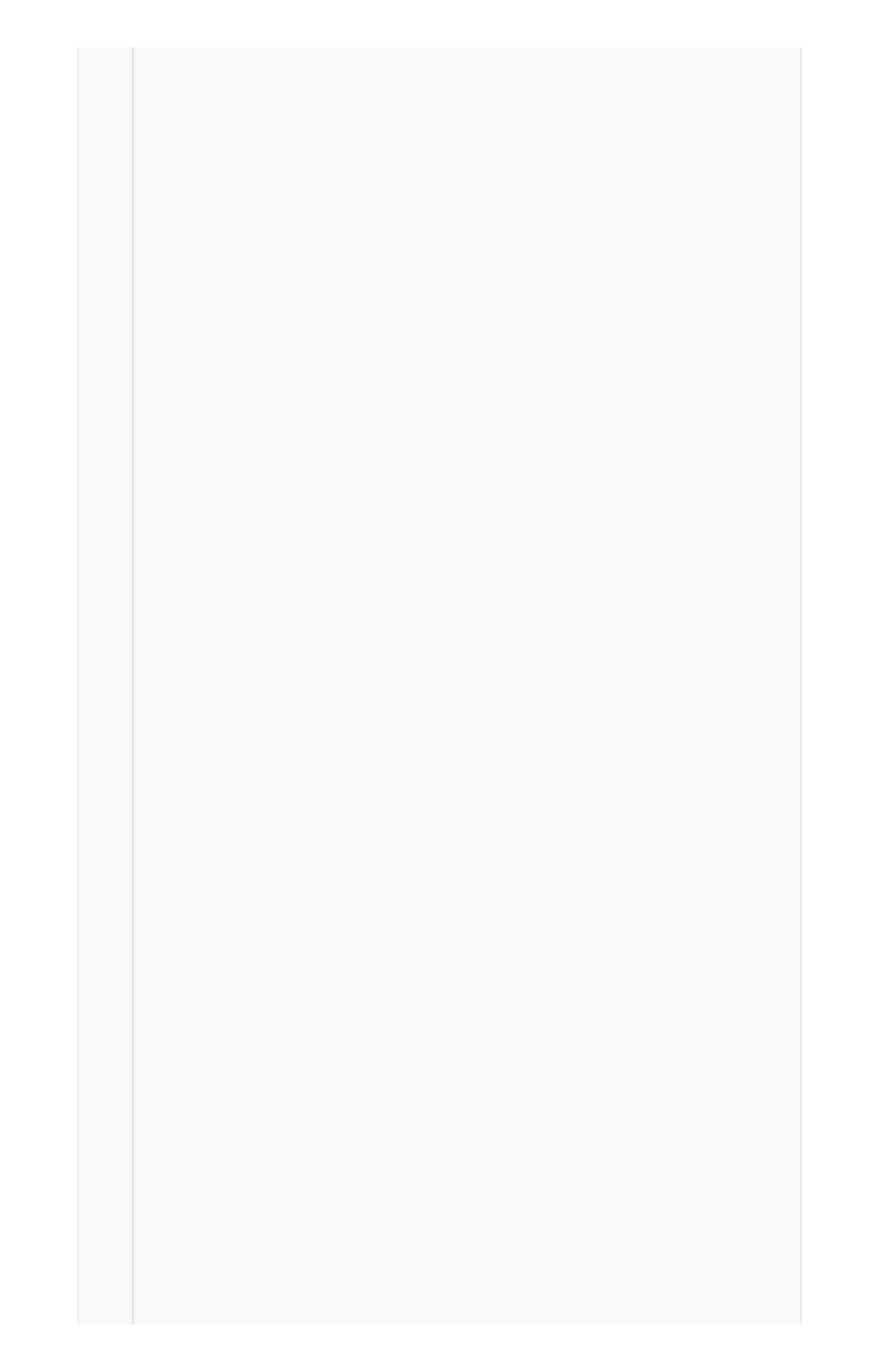
144

145

146

splitFile(selectFile,splitNum);

JOptionPane.showMessageDialog(this,



147

148

149

150

151

"文件成功分割",

"消息提示",

JOptionPane.PLAIN\_MESSAGE);

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(this, "请输入分割的数

目", "！！",JOptionPane.WARNING\_MESSAGE);

152

153

154

155

156

157

158

159

160

161

162

163

}

} catch (FileNotFoundException ex) {

ex.printStackTrace();

} catch (IOException ex) {

ex.printStackTrace();

}

}

else{

JOptionPane.showMessageDialog(this, "请选择文件", "没有选择文

件！！",JOptionPane.WARNING\_MESSAGE);

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

}

}

}

@Override

public void focusGained(FocusEvent e) {

//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。

if(jTextField1.getText().equals("请输入需要分割数目")){

jTextField1.setText("");

jTextField1.setForeground(Color.BLACK);

}

}

@Override

public void focusLost(FocusEvent e) {

//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

if(jTextField1.getText().length()<1){

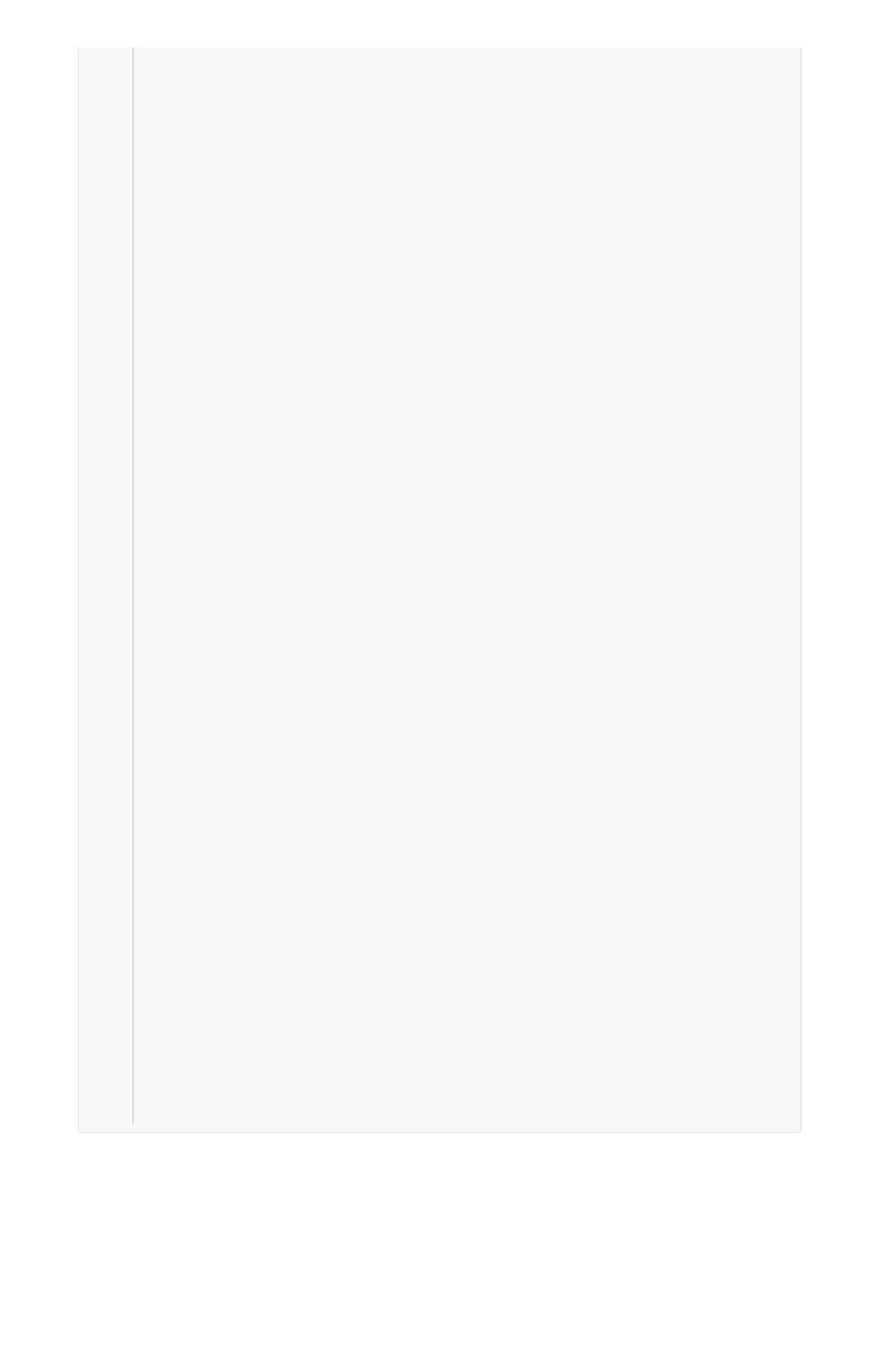
jTextField1.setText("请输入需要分割数目");

jTextField1.setForeground(new Color(204,204,204));

}

}

}



**一些要点**

1

2

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。

3

4

5

6

7

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

8

9

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

10

11 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

12

13 jButton.addActionListener(this);

14

15

16

17

18 3. IO包

19 IO就是input output 即输入 输出

20

21 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

