**方块苗文信息处理系统源程序**

1. ex

package com.Miaowen;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class ex {

public static void main(String[] args)

{

new Test1();

}

}

1. Test1

package com.Miaowen;

import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

import java.io.\*;

import java.awt.print.PageFormat;

import java.awt.print.Printable;

import java.awt.print.PrinterException;

import java.awt.print.PrinterJob;

import javax.jws.soap.SOAPBinding.Style;

import javax.swing.\*;

import javax.swing.event.CaretEvent;

import javax.swing.event.CaretListener;

import javax.swing.filechooser.FileFilter;

import javax.swing.text.BadLocationException;

import javax.swing.text.SimpleAttributeSet;

import javax.swing.text.StyleConstants;

import javax.swing.text.StyleContext;

import javax.swing.text.StyledDocument;

import javax.swing.undo.UndoManager;

import javax.xml.stream.events.StartDocument;

import org.w3c.dom.Document;

import java.awt.Dialog.ModalExclusionType;

public class Test1 extends JFrame implements ActionListener ,MouseListener,CaretListener

{

private static final long serialVersionUID = 1L;

private static final Object button = null;

private File file;

private JTextPane text;

private JMenu menu[];

private JFileChooser fchooser;

private PrinterJob prtMe;

private JCheckBoxMenuItem checklinewrapmenuitem;

private JPopupMenu popupmenu;

private UndoManager myundo = new UndoManager();

//UndoManager 管理 UndoableEdit 列表，提供撤消或恢复适当编辑的方法。

//FindJFrame findnext;

//Clipboard clipbd=getToolkit().getSystemClipboard();

private JPanel panelstatus;

private JTextField text\_status;

private JCheckBoxMenuItem checkstatusmenuitem;

private JButton find;

Dimension dim = getToolkit().getScreenSize();//获得屏幕分辨率

int Width=dim.width;

int Height=dim.height;

FindJFrame findjframe;

public Test1()

{

super("方块苗文信息处理系统");

this.setExtendedState(JFrame.MAXIMIZED\_BOTH);

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

System.out.println(Width+"A"+Height);

text = new JTextPane();

text.setFont(new Font("方块苗文楷体", Font.PLAIN, 22));

text.getDocument().addUndoableEditListener(myundo);

text.addCaretListener(this);

getContentPane().setLayout(null);

getContentPane().setLayout(null);

JScrollPane scrollPane = new JScrollPane(text);

scrollPane.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_ALWAYS);

int x1=(int) (Width\*0.6588);

int y1=(int) (Height\*0.8164);

System.out.println(x1+"+1+"+y1);

scrollPane.setBounds(226, 27,x1, y1);

this.getContentPane().add(scrollPane); //JTextField加滚动条

JMenuBar menubar = new JMenuBar(); //加菜单栏

menubar.setFont(new Font("微软雅黑", Font.PLAIN, 14));

this.setJMenuBar(menubar);

String[] menustr = {"文件","编辑","格式","查看","帮助"};

menu = new JMenu[menustr.length];

for(int i=0;i<menustr.length;i++)

{

menu[i] = new JMenu(menustr[i]);

menubar.add(menu[i]);

}

Filemenu();

Editmenu();

Formatmenu();

watchmenu();

helpmenu();

//加弹出菜单

text.addMouseListener(this); //JTextField中加鼠标监听器

mypopupmenu();

text.add(popupmenu);

MyToolbar(); //加工具栏

MyStatusbar(); //添加装态栏

this.setVisible(true);

this.file=null;

this.fchooser=new JFileChooser(new File(".","")); //文件对话框的初始路径是当前路径

// this.fchooser.setFileFilter(new FileFilter("文本文件(\*.txt)","txt")); //设置文件过滤器

prtMe = PrinterJob.getPrinterJob();

//PrinterJob 类是控制打印的主要类。应用程序调用此类中的方法设置作业、（可选地）调用与用户的打印对话框，然后打印作业的页面。

}

public Test1(File file)

{

this();

if(file!=null)

{

this.file=file;

this.text.setText(this.file.getName());

this.setTitle(this.file.getName());

}

}

public Test1(String filename)

{

this(new File(filename));

}

private void MyStatusbar() //状态栏

{

panelstatus=new JPanel();

int x2=(int) (Width\*0.5856);

int y2=(int) (Height\*0.0527);

System.out.println(x2+"+2+"+y2);

panelstatus.setBounds(63, 652, x2, y2);

this.getContentPane().add(panelstatus);

panelstatus.setLayout(null);

text\_status=new JTextField(25);

text\_status.setBounds(461, 0, 339, 33);

text\_status.setHorizontalAlignment(SwingConstants.CENTER);

text\_status.setFont(new Font("微软雅黑", Font.PLAIN, 15));

text\_status.setForeground(Color.BLACK);

text\_status.setBackground(Color.WHITE);

panelstatus.add(text\_status);

panelstatus.setVisible(true);

}

private void MyToolbar()

{

JToolBar toolbar=new JToolBar();

toolbar.setFloatable(false);

int x3=(int) (Width\*0.9970);

int y3=(int) (Height\*0.0351);

System.out.println(x3+"+3+"+y3);

toolbar.setBounds(0, 0, x3, y3);

this.getContentPane().add(toolbar);

JButton bcreatfile =new JButton("新建");

bcreatfile.setHorizontalAlignment(SwingConstants.LEFT);

bcreatfile.setIcon(new ImageIcon(Test1.class.getResource("/javax/swing/plaf/metal/icons/ocean/directory.gif")));

bcreatfile.addActionListener(this);

toolbar.add(bcreatfile);

JButton bopen=new JButton("打开");

bopen.setIcon(new ImageIcon(Test1.class.getResource("/com/sun/javafx/scene/web/skin/Copy\_16x16\_JFX.png")));

bopen.addActionListener(this);

toolbar.add(bopen);

JButton bsave=new JButton("保存");

bsave.setIcon(new ImageIcon(Test1.class.getResource("/javax/swing/plaf/metal/icons/ocean/floppy.gif")));

bsave.addActionListener(this);

toolbar.add(bsave);

JButton bundo=new JButton("撤消");

bundo.setIcon(new ImageIcon(Test1.class.getResource("/javax/swing/plaf/metal/icons/ocean/minimize-pressed.gif")));

bundo.addActionListener(this);

toolbar.add(bundo);

JButton bredo=new JButton("恢复");

bredo.setIcon(new ImageIcon(Test1.class.getResource("/com/sun/javafx/scene/web/skin/Undo\_16x16\_JFX.png")));

bredo.addActionListener(this);

toolbar.add(bredo);

find = new JButton("\u67E5\u627E");

find.setIcon(new ImageIcon(Test1.class.getResource("/com/sun/javafx/scene/web/skin/Paste\_16x16\_JFX.png")));

find.addActionListener(this);

// find.addActionListener(new ActionListener() {

//public void actionPerformed(ActionEvent arg0) {if (arg0.getSource()==find){

// new FindJFrame(text);}

//}

//

// });

toolbar.add(find);

JButton button\_1 = new JButton("");

button\_1.setToolTipText("字体");

button\_1.setIcon(new ImageIcon(Test1.class.getResource("/com/sun/javafx/scene/web/skin/FontColor\_16x16\_JFX.png")));

button\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {if (arg0.getSource()==button\_1){new WordStyle(text);}

}

});

toolbar.add(button\_1);

StyledDocument doc = text.getStyledDocument();

JButton button\_2 = new JButton("");

button\_2.setToolTipText("左对齐");

button\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) { if(arg0.getSource()==button\_2) {

SimpleAttributeSet aSet = new SimpleAttributeSet();

StyleConstants.setAlignment(aSet, StyleConstants.ALIGN\_LEFT);

doc.setParagraphAttributes(text.getSelectionStart(), text.getSelectionEnd()-text.getSelectionStart(), aSet, false);

} //左对齐

}

});

button\_2.setIcon(new ImageIcon(Test1.class.getResource("/com/sun/javafx/scene/control/skin/modena/HTMLEditor-Bullets-Black.png")));

toolbar.add(button\_2);

JButton button\_3 = new JButton("");

button\_3.setToolTipText("右对齐");

button\_3.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) { SimpleAttributeSet bSet = new SimpleAttributeSet();

StyleConstants.setAlignment(bSet, StyleConstants.ALIGN\_RIGHT);

doc.setParagraphAttributes(text.getSelectionStart(), text.getSelectionEnd()-text.getSelectionStart(), bSet, false);

}

});

button\_3.setIcon(new ImageIcon(Test1.class.getResource("/com/sun/javafx/scene/control/skin/modena/HTMLEditor-Bullets-Black-rtl.png")));

toolbar.add(button\_3);

JButton button\_4 = new JButton("");

button\_4.setToolTipText("居中");

button\_4.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {SimpleAttributeSet cSet = new SimpleAttributeSet();

StyleConstants.setAlignment(cSet, StyleConstants.ALIGN\_CENTER);

doc.setParagraphAttributes(text.getSelectionStart(), text.getSelectionEnd()-text.getSelectionStart(), cSet, false);

}

});

button\_4.setIcon(new ImageIcon(Test1.class.getResource("/com/sun/javafx/scene/control/skin/modena/HTMLEditor-Center-Black.png")));

toolbar.add(button\_4);

JButton button\_5 = new JButton("");

button\_5.setToolTipText("加粗");

button\_5.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {SimpleAttributeSet dSet = new SimpleAttributeSet();

StyleConstants.setBold(dSet,true);

doc.setParagraphAttributes(text.getSelectionStart(), text.getSelectionEnd()-text.getSelectionStart(), dSet, false);

}

});

button\_5.setIcon(new ImageIcon(Test1.class.getResource("/com/sun/javafx/scene/web/skin/Bold\_16x16\_JFX.png")));

toolbar.add(button\_5);

}

private void mypopupmenu() //弹出菜单

{

popupmenu=new JPopupMenu();

String menuitemstr[]={"撤消","剪切","复制","粘贴","全选"};

JMenuItem popupmenuitem[]=new JMenuItem[menuitemstr.length];

for(int i=0;i<menuitemstr.length;i++)

{

popupmenuitem[i]=new JMenuItem(menuitemstr[i]);

if(i==1||i==4||i==5)

popupmenu.addSeparator();

popupmenu.add(popupmenuitem[i]);

popupmenuitem[i].addActionListener(this);

}

popupmenuitem[0].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_Z,InputEvent.CTRL\_MASK));

popupmenuitem[1].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_X,InputEvent.CTRL\_MASK)); //剪切快捷方式ctrl+X

popupmenuitem[2].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_C,InputEvent.CTRL\_MASK));

popupmenuitem[3].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_V,InputEvent.CTRL\_MASK));

popupmenuitem[4].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_A,InputEvent.CTRL\_MASK));

}

private void Filemenu()

{

String menuitemstr[] = {"新建" ,"打开","保存","另存为","打印","打印设置","退出"};

JMenuItem menuitem[]=new JMenuItem[menuitemstr.length];

for(int i=0;i<menuitemstr.length;i++)

{

menuitem[i]= new JMenuItem(menuitemstr[i]);

menuitem[i].addActionListener(this);

menu[0].add(menuitem[i]);

if(i==3||i==5)

menu[0].addSeparator(); //加菜单分隔符

}

}

private void Editmenu()

{

String menuitemstr[]={"撤消","恢复","剪切","复制","粘贴","删除","查找","替换","全选","时间/日期"};

JMenuItem menuitem[]=new JMenuItem[menuitemstr.length];

for(int i=0;i<menuitemstr.length;i++)

{

menuitem[i]=new JMenuItem(menuitemstr[i]);

menuitem[i].addActionListener(this);

menu[1].add(menuitem[i]);

if(i==5||i==7)

menu[1].addSeparator();

}

// menuitem[9].setEnabled(false);//////"转到"为阴影,没有作用

// menuitem[9].removeActionListener(this);

}

private void Formatmenu()

{

checklinewrapmenuitem = new JCheckBoxMenuItem("自动换行",true); //true表示默认状态可以换行

checklinewrapmenuitem.setEnabled(true);

menu[2].add(checklinewrapmenuitem);

JMenuItem menuitem = new JMenuItem("字体");

menuitem.addActionListener(this);

menu[2].add(menuitem);

}

private void watchmenu()

{

checkstatusmenuitem = new JCheckBoxMenuItem("查看状态栏",true);

checklinewrapmenuitem.setEnabled(false);

menu[3].add(checkstatusmenuitem);

checkstatusmenuitem.addActionListener(this);//状态栏监听器

}

private void helpmenu()

{

JMenuItem menuitem1 = new JMenuItem("帮助主题");

menuitem1.addActionListener(this);

menu[4].add(menuitem1);

menu[4].addSeparator();

JMenuItem menuitem2 = new JMenuItem("关于记事本");

menuitem2.addActionListener(this);

menu[4].add(menuitem2);

}

public void writeToFile(String lines) //将字符串lines写入到当件文本文件中

{

try

{

FileWriter fout=new FileWriter(this.file);

fout.write(lines+"\n");

fout.close();

}

catch(IOException ioex)

{

JOptionPane.showMessageDialog(this,"有IO错，写入"+file.getName()+"文件不成功");

}

}

public String readFromFile() //将当前文本文件中的内容读出，以lines返回

{

char lines[]=null;

try

{

FileReader fin=new FileReader(this.file);

lines=new char[(int)this.file.length()];

fin.read(lines);

fin.close();

}

catch(FileNotFoundException fe)

{

//JOptionPane.showInternalConfirmDialog(this,"""+this.file.getName()+""文件不存在", defaultCloseOperation);

}

catch(IOException ioex)

{

JOptionPane.showMessageDialog(this,"IO错，读入"+file.getName()+"文件不成功");

}

finally

{

return new String(lines);

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getActionCommand()=="新建")

{

this.file=null;

this.setTitle("未命名");

this.text.setText("");

}

else if(e.getActionCommand()=="打开"&&fchooser.showOpenDialog(this)==JFileChooser.APPROVE\_OPTION)

//点击打开，显示打开对话框，选中文件后选择确定(确定以JFileChooser.APPROVE\_OPTION(int类型)返回

{

this.file=fchooser.getSelectedFile();

this.setTitle(this.file.getName());

this.text.setText(this.readFromFile());

}

else if(e.getActionCommand()=="保存"&&this.file!=null) //保存非空文件，不显示保存文件对话框

this.writeToFile(this.text.getText());

else if((e.getActionCommand()=="保存"&&file==null||e.getActionCommand()=="另存为")

&&fchooser.showSaveDialog(this)==JFileChooser.APPROVE\_OPTION)

//显示保存（空文件）或点击另存为按扭，出现保存对话框，点击确定（点击确定后会返回JFileChooser.APPROVE\_OPTION

{

this.file=fchooser.getSelectedFile(); //file为当前选中的文件

if(!file.getName().endsWith(".txt"));

this.file=new File(file.getAbsolutePath()+".txt");

this.writeToFile(this.text.getText());

this.setTitle(this.file.getName());

}

else if(e.getActionCommand()=="打印设置")

prtMe.printDialog(); //显示找印对话框，供用户更改打印属性

else if(e.getActionCommand()=="打印")

{

try{

// text.print(null);

prtMe.print();//打印一组页面

}

catch(Exception ew)

{

JOptionPane.showMessageDialog(this, "找不到打印机");

}

}

else if(e.getActionCommand()=="退出")

{

if(JOptionPane.showConfirmDialog(this,"终止当前程序运行?")==0)

System.exit(0);

}

else if(e.getActionCommand()=="撤消"||e.getActionCommand()=="恢复")

{

if(e.getActionCommand()=="撤消")

{

if(myundo.canUndo())

myundo.undo();

else JOptionPane.showMessageDialog(this,"无法撤消");

}

else

{

if(myundo.canRedo())

myundo.redo();

else JOptionPane.showMessageDialog(this,"无法恢得");

}

}

else if(e.getActionCommand()=="复制"){

text.copy();

}

else if(e.getActionCommand()=="剪切")

{

text.cut();

}

else if(e.getActionCommand()=="粘贴")

{

text.paste();

}

else if(e.getActionCommand()=="删除") //replaceRange(String,start,end)用指定替换文本

text.replaceSelection("");//以空字符串代串选中的范围

else if(e.getActionCommand()=="查找")

{

new FindJFrame(text);

}

else if(e.getActionCommand()=="替换")

{

new ReplaceJFrame(text);

}

else if(e.getActionCommand()=="全选")

text.selectAll();

else if(e.getActionCommand()=="时间/日期") //在JTextArea后追加显示系统当前时间

{

Calendar time =Calendar.getInstance();

String str=" "+time.HOUR\_OF\_DAY+":"+time.MINUTE+" "+time.YEAR+"-"+time.MONTH+"-"+time.DAY\_OF\_MONTH+"\r";

javax.swing.text.Document doc = text.getDocument();

if (doc != null)

{

try

{

// void insertString(int offset, String str, AttributeSet a)throws BadLocationException插入内容字符串。

doc.insertString(doc.getLength(), str, null);

}

catch (BadLocationException e1)

{

}

}

}

else if(e.getActionCommand()=="字体")

{

new WordStyle(text);

}

else if(e.getActionCommand()=="查看状态栏")

{

panelstatus.setVisible(checkstatusmenuitem.isSelected());

}

else if(e.getActionCommand()=="帮助主题")

{

JOptionPane.showMessageDialog(this,"do it by yourself！");

}

else if(e.getActionCommand()=="关于记事本")

{

JOptionPane.showMessageDialog(this,"方块苗文编辑器");

}

}

public void caretUpdate(CaretEvent e) //状态栏上显示光标所在的行号和列号

{

try

{

int pos = text.getCaretPosition();

int posLine;

int y = 0;

int X=0;

try

{

//public Rectangle modelToView(int pos)

// throws BadLocationException将模型中给定位置转换为视图坐标系统中的位置。

Rectangle caretCoords = text.modelToView(pos);

y = (int) caretCoords.getY();

X = (int) caretCoords.getX();

}

catch (BadLocationException ex)

{

}

//public FontMetrics getFontMetrics(Font font)

//FontMetrics 类定义字体规格对象

//int getHeight() 获取此 Font 中文本行的标准 height

int lineHeight = text.getFontMetrics(text.getFont()).getHeight();

posLine = (y / lineHeight) + 1;

int col=text.viewToModel(new Point(X,lineHeight));

text\_status.setCaretColor(Color.red);

text\_status.setText("字符总数:"+pos+" 当前位置 Line:"+posLine+" 当前行的字数:"+col);

}

catch(Exception ex)

{

text\_status.setText( " 无法获得当前光标位置 ");

}

}

public void mouseClicked(MouseEvent mec)

{

if(mec.getModifiers()==MouseEvent.BUTTON3\_MASK)

popupmenu.show(text,mec.getX(),mec.getY()); //在鼠标点击的地方显示弹出菜单

}

public void mousePressed(MouseEvent mep) {}

public void mouseReleased(MouseEvent mre) {}

public void mouseEntered(MouseEvent med) {}

public void mouseExited(MouseEvent me) {}

public void mouseDragged(MouseEvent mdg) {}

}

1. FindJFrame

package com.Miaowen;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.text.Caret;

import javax.swing.text.SimpleAttributeSet;

import javax.swing.text.StyleConstants;

import java.awt.Dialog.ModalExclusionType;

import java.awt.Window.Type;

public class FindJFrame extends JFrame implements ActionListener{

private JTextPane text;

private JTextField findstr;

private JButton btn\_findnext,btn\_cannel,radioButton,radioButton\_1, find\_fast;

private int start=0,end=0;

private final int FIND=1; //找到标志

private int flag=0; //标志是否找到

int sum=0;

int x =0; //计算查找的字数

int kdw=0;

String str;

public FindJFrame()

{

super("查找");

setType(Type.UTILITY);

setBackground(Color.WHITE);

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

setAlwaysOnTop(true);

this.setBounds(200,300,369,230);

getContentPane().setLayout(new GridLayout(2,1));

Font f1=new Font("方块苗文楷体",Font.PLAIN,15);

Font f2=new Font("微软雅黑",Font.PLAIN,15);

Panel panel = new Panel();

panel.setLayout(null);

getContentPane().add(panel);

Label label\_2 = new Label("是否查找方块苗文");

label\_2.setBounds(25, 41, 110, 24);

panel.add(label\_2);

JRadioButton radioButton = new JRadioButton("是",true);

radioButton.setBounds(147, 41, 62, 23);

panel.add(radioButton);

JRadioButton radioButton\_1 = new JRadioButton("否", false);

radioButton\_1.setBounds(251, 42, 62, 23);

panel.add(radioButton\_1);

ButtonGroup bgroup1 = new ButtonGroup();

bgroup1.add(radioButton);

bgroup1.add(radioButton\_1);

radioButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {if (arg0.getSource()==radioButton){findstr.setFont(f1);}

}

});

radioButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {if (arg0.getSource()==radioButton\_1){findstr.setFont(f2);}

}

});

Panel panel2=new Panel();

this.getContentPane().add(panel2);

panel2.setLayout(null);

Label label = new Label("查找内容");

label.setBounds(25, 29, 54, 24);

panel2.add(label);

findstr=new JTextField(15);

findstr.setToolTipText("");

findstr.setForeground(Color.BLACK);

findstr.setHorizontalAlignment(SwingConstants.LEFT);

findstr.setFont(new Font("方块苗文楷体",Font.PLAIN, 15));

findstr.setBounds(85, 29, 115, 24);

panel2.add(findstr);

btn\_findnext=new JButton("查找下一个");

btn\_findnext.setBounds(222, 29, 100, 23);

panel2.add(btn\_findnext);

//find\_fast = new JButton("从当前位置查找");

// find\_fast.setBounds(188, 54, 128, 23);

// panel2.add(find\_fast);

btn\_findnext.addActionListener(this);

//find\_fast.addActionListener(this);

this.setVisible(true);////////////////////////////

}

public FindJFrame(JTextPane text)

{

this();

this.text=text;

end=text.getText().length();

}

public void setfindstr(String str) //设置要查找的字符串

{

this.findstr.setText(str);

}

public int getStart() //获得选中文本的起始位置

{

return start;

}

public int getEnd() //选中文本的终止位置

{

return end;

}

public void setTitle(String name) //设置文本标题

{

super.setTitle(name);

}

public void setStart(int start) //设置鼠标起始位置

{

if(start>=0)

//this.start=0;

this.start=start;

else start=0;

}

public void setEnd(int end) //设置鼠标终止位置

{

if(end>=0)

this.end=end;

else end=text.getText().length();

}

public boolean isFind() //判断是否找到

{

if(flag==FIND)

return true;

else return false;

}

public void find()

{

flag=0;

if(start<0)

{

JOptionPane.showMessageDialog(this,"位置超出范围");

return ;

}

setEnd(text.getText().length());

str=text.getText().substring(start,end);

int findstart=str.indexOf(findstr.getText()); //没有找到findstr的话返加-1

if(findstart!=-1)//找到

{flag=FIND;sum++;

int k=str.lastIndexOf("\r",findstart-1);

while(k!=-1){

kdw++;

k=str.lastIndexOf("\r",k-1);

}

System.out.println(kdw);

}

else

{

text.setSelectionStart(end);

text.setSelectionEnd(end);

return ;

}

text.setSelectionStart(start+findstart-kdw);

int findend=start+findstart+findstr.getText().length();

text.setSelectionEnd(findend-kdw);

start=findend;

}

//从当前位置查找

/\* public void findfast(){

setStart(text.getCaretPosition());

find();

}\*/

public void actionPerformed(ActionEvent e) //点击"从头查找"

{

if(e.getSource()==btn\_findnext)

{

text.requestFocus(true);

text.setCaretPosition(0);

find();

if(!isFind())

{

getStart();

getEnd();

setStart(0);

// setEnd(text.getText().length());

JOptionPane.showMessageDialog(this,"查找结束，总共查找到："+sum+"个");

sum=0;

kdw=0;

}

}

/\* if(e.getSource()==find\_fast){ //从当前位置查找响应

// text.requestFocus(true);

findfast();

if(!isFind())

{

getStart();

getEnd();

setStart(0);

// setEnd(text.getText().length());

JOptionPane.showMessageDialog(this,"查找结束，总共查找到："+sum+"个");

sum=0;

kdw=0;

}

}\*/

}

private void FindJFrame(JTextPane text2) {

// TODO 自动生成的方法存根

}

}

1. WordStyle

package com.Miaowen;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GraphicsEnvironment;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

import javax.swing.text.SimpleAttributeSet;

import javax.swing.text.StyleConstants;

import javax.swing.text.StyledDocument;

public class WordStyle extends JFrame implements ActionListener{

private JComboBox wordname,wordstyle,wordsize; //组合框

private JRadioButton fontleft,fontcenter,fontright; //单选按钮

private JTextField showfrecolor,showbackcolor; //显示当前选择的前景色背景色预浏

private JButton frecolorchoose,backcolorchoose;

private JTextPane text,present;

private JCheckBox underline;

public WordStyle()

{

super("字体");

this.setBounds(200,200,500,500);

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

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

JPanel panel1=new JPanel();

this.getContentPane().add(panel1);

panel1.setSize(500,200);

panel1.add(new JLabel("字体"));

GraphicsEnvironment ge=GraphicsEnvironment.getLocalGraphicsEnvironment();

String[] fontName=ge.getAvailableFontFamilyNames(); //获取系统字体

wordname=new JComboBox(fontName);

wordname.addActionListener((ActionListener) this);

panel1.add(wordname);

panel1.add(new JLabel(" 字形"));

String[] fontstyle={"常规","加粗","倾斜","加粗 倾斜"};

wordstyle=new JComboBox(fontstyle);

wordstyle.addActionListener(this);

panel1.add(wordstyle);

panel1.add(new JLabel(" 字号")); //本想加数字,但不知道怎么弄

String[] fontsize={"5","5.5","6.5","7.5","8","9","10","11","12","13","14","15","16","17","18","19","20",

"24","28","30","40","50","60","70","80","90","100"};

wordsize=new JComboBox(fontsize);

wordsize.setSelectedItem("16");

wordsize.addActionListener(this);

panel1.add(wordsize);

JPanel panel2=new JPanel();

this.getContentPane().add(panel2);

panel2.setLayout(new GridLayout(1,2));

panel2.setSize(500,100);

JPanel panel21=new JPanel(); //前景颜色选择面板

panel2.add(panel21);

panel21.setLayout(new FlowLayout());

frecolorchoose = new JButton("前景色");

frecolorchoose.addActionListener(this);

panel21.add(frecolorchoose);

showfrecolor=new JTextField(10);

showfrecolor.setBackground(Color.black);

panel21.add(showfrecolor);

JPanel panel22=new JPanel(); /////背景字体对齐面板

panel2.add(panel22);

panel22.setLayout(new FlowLayout());

backcolorchoose = new JButton("背景色");

backcolorchoose.addActionListener(this);

panel22.add(backcolorchoose);

showbackcolor=new JTextField(10);

panel22.add(showbackcolor);

JPanel panel3 =new JPanel();

this.getContentPane().add(panel3);

panel3.setSize(400,100);

ButtonGroup group=new ButtonGroup();

fontleft =new JRadioButton("左对齐",true);

group.add(fontleft);

panel3.add(fontleft);

fontleft.addActionListener(this);

fontcenter =new JRadioButton("居中");

group.add(fontcenter);

panel3.add(fontcenter);

fontcenter.addActionListener(this);

fontright =new JRadioButton("右对齐");

group.add(fontright);

panel3.add(fontright);

fontright.addActionListener(this);

underline=new JCheckBox("下划线");

panel3.add(underline);

underline.addActionListener(this);

JPanel panel4 =new JPanel();

this.getContentPane().add(panel4);

panel4.setSize(400,200);

present=new JTextPane();

panel4.add(present);

present.setEditable(false);

this.setResizable(false);

this.setVisible(true);

}

public WordStyle(JTextPane text)

{

this();

this.text=text;

}

public void actionPerformed(ActionEvent e)

{

int size=0;

if(e.getSource()==underline)

{

SimpleAttributeSet bSet = new SimpleAttributeSet();

StyleConstants.setUnderline(bSet, true); //设置下划线

StyledDocument doc = text.getStyledDocument();

doc.setParagraphAttributes(text.getSelectionStart(), text.getSelectionEnd()-text.getSelectionStart(), bSet, false);

}

if(e.getSource() instanceof JComboBox)

{

try

{

String name=(String)wordname.getSelectedItem();

SimpleAttributeSet aSet = new SimpleAttributeSet();

//public class StyleConstantsextends Object 一个已知的或常见的属性键和方法的集合

StyleConstants.setFontFamily(aSet, name);

text.setFont(new Font(name, Font.PLAIN, 22));

System.out.println(name);

int style=this.text.getFont().getStyle();

if(style==0) //常规

{

StyleConstants.setBold(aSet,false);

StyleConstants.setItalic(aSet,false);

}

else if(style==1) //粗体

{

StyleConstants.setBold(aSet,true);

}

else if(style==2) //斜体

{

StyleConstants.setItalic(aSet, false);

}

else if(style==3) //粗斜

{

StyleConstants.setBold(aSet,true);

StyleConstants.setItalic(aSet, true);

}

size=Integer.parseInt((String)(wordsize.getSelectedItem()));

if(size<5||size>100)

throw new Exception("the size is error");

StyleConstants.setFontSize(aSet, size);

StyledDocument doc = text.getStyledDocument();

doc.setCharacterAttributes(text.getSelectionStart(), text.getSelectionEnd()-text.getSelectionStart(), aSet, false);

}

catch(NumberFormatException nfe)

{

JOptionPane.showMessageDialog(this,(String)wordsize.getSelectedItem()+"不能转化成整数");

}

catch(Exception ec)

{

if(ec.getMessage()=="fontsizeException")

JOptionPane.showMessageDialog(this,size+" 字号不合适(5,100)");

}

}

if(e.getSource() instanceof JRadioButton)

{

SimpleAttributeSet bSet = new SimpleAttributeSet();

if(e.getSource()==fontleft) //左对齐

StyleConstants.setAlignment(bSet, StyleConstants.ALIGN\_LEFT);

if(e.getSource()==fontcenter) //居中

StyleConstants.setAlignment(bSet, StyleConstants.ALIGN\_CENTER);

if(e.getSource()==fontright) //右对齐

StyleConstants.setAlignment(bSet, StyleConstants.ALIGN\_RIGHT);

StyledDocument doc = text.getStyledDocument();

doc.setParagraphAttributes(text.getSelectionStart(), text.getSelectionEnd()-text.getSelectionStart(), bSet, false);

}

if(e.getSource() instanceof JButton)

{

Color color;

if(e.getSource()==frecolorchoose)

{

color=JColorChooser.showDialog(this,"前景色选择",Color.black);

showfrecolor.setBackground(color);

this.present.setForeground(color);

SimpleAttributeSet aSet = new SimpleAttributeSet();

StyleConstants.setForeground(aSet, color);

StyledDocument doc=text.getStyledDocument();

doc.setCharacterAttributes(text.getSelectionStart(),text.getSelectionEnd()-text.getSelectionStart(), aSet, false);

}

else

{

color=JColorChooser.showDialog(this,"背景色选择",Color.white);

showbackcolor.setBackground(color);

this.present.setBackground(color);

SimpleAttributeSet aSet = new SimpleAttributeSet();

StyleConstants.setBackground(aSet, color);

StyledDocument doc=text.getStyledDocument();

doc.setCharacterAttributes(text.getSelectionStart(),text.getSelectionEnd()-text.getSelectionStart(), aSet, false);

}

}

}

}

1. ReplaceJFrame

package com.Miaowen;

import java.awt.\*;

import java.awt.event.\*; //Action

import javax.swing.\*;

import javax.swing.event.\*; //Caret

public class ReplaceJFrame extends JFrame implements ActionListener,CaretListener{

private static final long serialVersionUID = 1L;

private JTextPane text;

private JTextField text\_find,text\_replace;

private JButton btn\_findnext , btn\_replace,btn\_replaceall;

private int x; //替换的个数

private int y; //查找到的个数

FindJFrame findjframe ;

public ReplaceJFrame()

{

super("替换");

this.setBounds(200,200,522,329);

// this.setLayout(new GridLayout(5,1));

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

Font f1=new Font("方块苗文楷体",Font.PLAIN,15);

Font f2=new Font("微软雅黑",Font.PLAIN,15);

Container container=this.getContentPane();

getContentPane().setLayout(null);

JPanel panel1=new JPanel(); //Panel默认情况下是FlowLayout

panel1.setBounds(41, 94, 277, 34);

container.add(panel1);

panel1.setLayout(null);

JLabel lbln = new JLabel("\u67E5\u627E\u5185\u5BB9\uFF08N\uFF09: ");

lbln.setBounds(5, 9, 96, 15);

panel1.add(lbln);

text\_find=new JTextField("",20);

text\_find.setBounds(106, 5, 166, 24);

text\_find.setFont(new Font("方块苗文楷体", Font.PLAIN, 15));

panel1.add(text\_find);

JPanel panel2=new JPanel();

panel2.setBounds(323, 94, 127, 33);

container.add(panel2);

btn\_findnext=new JButton("查找下一个");

btn\_findnext.setBounds(5, 5, 112, 23);

btn\_findnext.addActionListener(this);

panel2.setLayout(null);

panel2.add(btn\_findnext);

JPanel panel3=new JPanel();

panel3.setBounds(41, 184, 277, 34);

container.add(panel3);

panel3.setLayout(null);

JLabel lblp = new JLabel("\u66FF\u6362\u4E3A\uFF08P): ");

lblp.setBounds(8, 9, 90, 15);

panel3.add(lblp);

text\_replace=new JTextField("",20);

text\_replace.setFont(new Font("方块苗文楷体", Font.PLAIN, 15));

text\_replace.setBounds(103, 5, 166, 24);

panel3.add(text\_replace);

JPanel panel4=new JPanel();

panel4.setBounds(323, 185, 83, 33);

container.add(panel4);

btn\_replace=new JButton("替换");

btn\_replace.setBounds(5, 5, 76, 23);

btn\_replace.addActionListener(this);

panel4.setLayout(null);

panel4.add(btn\_replace);

btn\_replaceall=new JButton("全部替换");

btn\_replaceall.setFont(new Font("微软雅黑", Font.PLAIN, 15));

btn\_replaceall.setBounds(202, 253, 111, 23);

getContentPane().add(btn\_replaceall);

JLabel label\_2 = new JLabel("\u662F\u5426\u67E5\u627E\u65B9\u5757\u82D7\u6587");

label\_2.setFont(new Font("微软雅黑", Font.PLAIN, 12));

label\_2.setBounds(45, 44, 103, 33);

getContentPane().add(label\_2);

JLabel label\_3 = new JLabel("\u662F\u5426\u66FF\u6362\u65B9\u5757\u82D7\u6587");

label\_3.setFont(new Font("微软雅黑", Font.PLAIN, 12));

label\_3.setBounds(41, 141, 103, 33);

getContentPane().add(label\_3);

JRadioButton radioButton = new JRadioButton("是");

radioButton.setSelected(true);

radioButton.setBounds(190, 49, 65, 23);

getContentPane().add(radioButton);

JRadioButton radioButton\_1 = new JRadioButton("否");

radioButton\_1.setBounds(323, 49, 65, 23);

getContentPane().add(radioButton\_1);

ButtonGroup bgroup1 = new ButtonGroup();

bgroup1.add(radioButton);

bgroup1.add(radioButton\_1);

radioButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {if (arg0.getSource()==radioButton){text\_find.setFont(f1);}

}

});

radioButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {if (arg0.getSource()==radioButton\_1){text\_find.setFont(f2);}

}

});

JRadioButton radioButton\_2 = new JRadioButton("是");

radioButton\_2.setSelected(true);

radioButton\_2.setBounds(190, 146, 65, 23);

getContentPane().add(radioButton\_2);

JRadioButton radioButton\_3 = new JRadioButton("否");

radioButton\_3.setBounds(323, 146, 65, 23);

getContentPane().add(radioButton\_3);

btn\_replaceall.addActionListener(this);

ButtonGroup bgroup2 = new ButtonGroup();

bgroup2.add(radioButton\_2);

bgroup2.add(radioButton\_3);

radioButton\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {if (arg0.getSource()==radioButton\_2){text\_replace.setFont(f1);}

}

});

radioButton\_3.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {if (arg0.getSource()==radioButton\_3){text\_replace.setFont(f2);}

}

});

this.setResizable(false);

this.setVisible(true);

}

public ReplaceJFrame(JTextPane text)

{

this();

this.text=text;

findjframe = new FindJFrame(this.text);

findjframe.setVisible(false);

findjframe.setStart(0);

findjframe.setEnd(text.getText().length());

}

public void caretUpdate(CaretEvent e)

{

findjframe.setfindstr(text\_find.getText());

}

public void actionPerformed(ActionEvent e) {

{

findjframe.setfindstr(text\_find.getText());

if(e.getSource()==btn\_findnext)

{

findjframe.find();

y=findjframe.sum;

if(!findjframe.isFind())

{JOptionPane.showMessageDialog(this,"查找结束，总共查找到"+y+"个;"+"替换了"+x+"个");

findjframe.getStart();

findjframe.getEnd();

findjframe.setStart(0);

findjframe.setEnd(text.getText().length());}

}

else if(e.getSource()==btn\_replace)

{

if(text\_replace.getText()=="")

{

JOptionPane.showMessageDialog(this,"请输入要替换的内容");

return ;

}

if(findjframe.isFind()) //调用find后,会把找到的字符串选中

{

String tempstr=text.getText().substring(text.getSelectionStart(), text.getSelectionEnd());

if(tempstr.equals(text\_find.getText())) //找到了 用text\_replace替换

{text.replaceSelection(text\_replace.getText()); x++;}

}

}

else if(e.getSource()==btn\_replaceall) //从文本开头,替换所有

{

if(text\_replace.getText()=="")

{

JOptionPane.showMessageDialog(this,"请输入要替换的内容");

return ;

}

int len=text.getText().length();

findjframe.setStart(text.getSelectionStart());

findjframe.setEnd(len);

findjframe.find();

while(findjframe.isFind())

{

String tempstr=text.getText().substring(text.getSelectionStart(), text.getSelectionEnd());

if(tempstr.equals(text\_find.getText()))

{

text.replaceSelection(text\_replace.getText());

}

findjframe.find();x++;

}

y=findjframe.sum;

JOptionPane.showMessageDialog(this,"查找结束，总共查找到"+y+"个;"+"替换了"+x+"个");

x=0;y=0;

}

}

}

}