import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class EditorJFrame extends JFrame implements ActionListener, MouseListener

{

   private JComboBox combox\_name, combox\_size;            //字体、字号组合框

   private JCheckBox checkb\_bold, checkb\_italic;          //粗体、斜体复选框

   private JRadioButton radiob\_color[];                   //颜色单选按钮

   private JTextArea text;                                //文本区

   private Color color;                                   //text的当前文本色

   private JPopupMenu popupmenu;                          //快捷菜单

   public EditorJFrame()

   {

       super("文本编辑器");                               //默认BorderLayout布局

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

       this.setBounds(dim.width/4,dim.height/4,dim.width/2,dim.height/2);  //窗口居中

       this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);      //窗口关闭时，程序结束

       text = new JTextArea("Welcome 欢迎");

       text.addMouseListener(this);                       //文本区注册鼠标事件监听器

       this.getContentPane().add(new JScrollPane(text));  //文本区添加到滚动窗格，滚动窗格添加到框架内容窗格中部

       JToolBar toolbar=new JToolBar();                   //创建工具栏，默认水平方向

       this.getContentPane().add(toolbar,"North");        //工具栏添加到框架内容窗格北部

       GraphicsEnvironment ge=GraphicsEnvironment.getLocalGraphicsEnvironment();

       String[] fontsName=ge.getAvailableFontFamilyNames(); //获得系统字体

       combox\_name = new JComboBox(fontsName);            //组合框显示系统字体

       combox\_name.addActionListener(this);               //组合框注册单击事件监听器

       toolbar.add(combox\_name);

       String sizestr[]={"20","30","40","50","60","70"};

       combox\_size = new JComboBox(sizestr);              //字号组合框

       combox\_size.setEditable(true);                     //设置组合框可编辑

       combox\_size.addActionListener(this);               //组合框注册单击事件监听器

       toolbar.add(combox\_size);

       checkb\_bold = new JCheckBox("粗体");               //字形复选框

       toolbar.add(checkb\_bold);

       checkb\_bold.addActionListener(this);               //复选框注册单击事件监听器

       checkb\_italic = new JCheckBox("斜体");

       toolbar.add(checkb\_italic);

       checkb\_italic.addActionListener(this);

       String colorstr[]={"红","绿","蓝"};

       ButtonGroup bgroup\_color = new ButtonGroup();      //按钮组

       radiob\_color = new JRadioButton[colorstr.length];  //颜色单选按钮数组

       for (int i=0; i<radiob\_color.length; i++)

       {

           radiob\_color[i]=new JRadioButton(colorstr[i]); //颜色单选按钮

           radiob\_color[i].addActionListener(this);

           bgroup\_color.add(radiob\_color[i]);             //单选按钮添加到按钮组

           toolbar.add(radiob\_color[i]);                  //单选按钮添加到工具栏

       }

       radiob\_color[0].setSelected(true);                 //设置单选按钮的选中状态

       this.addmyMenu();                                  //调用自定义方法，添加菜单

       this.setVisible(true);

//        String password = JOptionPane.showInputDialog(this, "密码");

//        System.out.println(password);                      //单击"撤消"按钮返回null

   }

   private void addmyMenu()                               //添加主菜单、快捷菜单、对话框

   {

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

       this.setJMenuBar(menubar);                         //框架上添加菜单栏

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

       JMenu menu[]=new JMenu[menustr.length];

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

       {

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

            menubar.add(menu[i]);                         //菜单栏中加入菜单

       }

       menu[0].add(new JMenuItem("打开"));                 //生成菜单项并加入到菜单

       menu[0].add(new JMenuItem("保存"));

       menu[0].addSeparator();                            //加分隔线

       JMenuItem menuitem\_exit = new JMenuItem("退出");

       menu[0].add(menuitem\_exit);

       menuitem\_exit.addActionListener(this);             //菜单项注册单击事件监听器

       JMenu menu\_style = new JMenu("字形");

       menu\_style.add(new JCheckBoxMenuItem("粗体"));

       menu\_style.add(new JCheckBoxMenuItem("斜体"));

/\*        String stylestr[]={"粗体", "斜体"};

       JCheckBoxMenuItem cbmenuitem\_style[]=new JCheckBoxMenuItem[stylestr.length];

       for (int i=0; i<cbmenuitem\_style.length; i++)

       {

           cbmenuitem\_style[i] = new JCheckBoxMenuItem(stylestr[i]);  //字形复选菜单项

           cbmenuitem\_style[i].addItemListener(this);

           menu\_style.add(cbmenuitem\_style[i]);

       }\*/

       menu[1].add(menu\_style);                           //菜单加入到菜单中成为二级菜单

       JMenu menu\_color = new JMenu("颜色");

       menu[1].add(menu\_color);

       ButtonGroup buttongroup = new ButtonGroup();       //按钮组

       String colorstr[]={"红","绿","蓝"};

       JRadioButtonMenuItem rbmi\_color[]=new JRadioButtonMenuItem[colorstr.length];

       for (int i=0; i<rbmi\_color.length; i++)

       {

           rbmi\_color[i] = new JRadioButtonMenuItem(colorstr[i]);  //单选菜单项

           buttongroup.add(rbmi\_color[i]);                //单选菜单项添加到按钮组

           rbmi\_color[i].addActionListener(this);

           menu\_color.add(rbmi\_color[i]);                 //单选菜单项添加到菜单

       }

       popupmenu = new JPopupMenu();                      //快捷菜单对象

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

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

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

       {

           popmenuitem[i] = new JMenuItem(menuitemstr[i]); //菜单项

           popupmenu.add(popmenuitem[i]);                  //快捷菜单加入菜单项

           popmenuitem[i].addActionListener(this);

       }

       popmenuitem[0].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_X, InputEvent.CTRL\_MASK));//设置快捷键Ctrl+X

       popmenuitem[1].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_C, InputEvent.CTRL\_MASK));//设置快捷键Ctrl+C

       popmenuitem[2].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_V, InputEvent.CTRL\_MASK));//设置快捷键Ctrl+V

       text.add(popupmenu);                               //文本区添加快捷菜单

   }

   public void actionPerformed(ActionEvent e)             //单击事件处理方法

   {

       if (e.getSource() instanceof JRadioButton)         //选择一个颜色复选框

       {

           if (e.getSource()==radiob\_color[0])

               color = new Color(255,0,0);

           if (e.getSource()==radiob\_color[1])

               color = new Color(0,255,0);

           if (e.getSource()==radiob\_color[2])

               color = new Color(0,0,255);

           text.setForeground(color);                     //设置文本区颜色

           return;

       }

       if (e.getSource() instanceof JMenuItem)            //单击菜单项

       {

           if (e.getActionCommand()=="退出")

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

                   System.exit(0);                        //单击确认对话框中的“是”按钮，结束程序运行

           if (e.getActionCommand()=="剪切")

               text.cut();                                //将选中文本剪切送系统剪贴板

           if (e.getActionCommand()=="复制")

               text.copy();                               //将选中文本复制送系统剪贴板

           if (e.getActionCommand()=="粘贴")

               text.paste();                              //将剪贴板的文本粘贴在当前位置

           return;

       }

       if (e.getSource() instanceof JComboBox || e.getSource() instanceof JCheckBox)

       {                                                  //组合框、复选框

           int size=0;

           try

           {

               String fontname = (String)combox\_name.getSelectedItem();//获得字体名

               size = Integer.parseInt((String)combox\_size.getSelectedItem());//获得字号

               if (size<4 || size>120)                    //字号超出指定范围时，抛出异常对象

                   throw new Exception("SizeException");

               java.awt.Font font = text.getFont();       //获得文本区的当前字体对象

               int style = font.getStyle();               //获得字形

               if (e.getSource()==checkb\_bold)            //粗体

                   style = style ^ 1;                     //整数的位运算，异或^

               if (e.getSource()==checkb\_italic)          //斜体

                   style = style ^ 2;

               text.setFont(new Font(fontname, style, size)); //设置文本区字体

           }

           catch(NumberFormatException nfe)

           {

               JOptionPane.showMessageDialog(this, "\""+(String)combox\_size.getSelectedItem()+"\" 不能转换成整数，请重新输入!");

           }

           catch(Exception ex)

           {

               if (ex.getMessage()=="SizeException")      //捕获自己抛出的异常对象

                   JOptionPane.showMessageDialog(this, size+" 字号不合适，请重新输入!");

           }

           finally{}

       }

       if (e.getSource()==combox\_size)                    //单击字号组合框

       {                    //将新输入的字号添加到组合框的下拉列表中，先要查找，不添加重复值

           String size = (String)combox\_size.getSelectedItem();//获得当前输入数据

           int i=0, n=combox\_size.getItemCount();         //n获得组合框的选项数

           while (i<n && size.compareTo((String)combox\_size.getItemAt(i))>=0)

           {                                              //size与组合框的选项比较

               if (size.compareTo((String)combox\_size.getItemAt(i))==0)

                   return ;                               //相同，不插入

               i++;

           }

           combox\_size.insertItemAt(size, i);             //将size插入在组合框的第i项

       }

   }

   public void mouseClicked(MouseEvent mec)               //鼠标事件处理方法，实现MouseListener接口

   {

       if (mec.getModifiers()==MouseEvent.BUTTON3\_MASK)   //单击的是鼠标右键

           popupmenu.show(text,mec.getX(),mec.getY());    //在鼠标单击处显示快捷菜单

   }

   public void mousePressed(MouseEvent mep) {}

   public void mouseReleased(MouseEvent mer) {}

   public void mouseEntered(MouseEvent mee) {}

   public void mouseExited(MouseEvent mex) {}

   public void mouseDragged(MouseEvent med) {}

   public static void main(String arg[])

   {

       new EditorJFrame();

   }

}