**《Java程序开发》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2020级计算机科学与技术卓越一班** | | | **姓名** | **孙莹莹** |
| **实验题目** | **文本编辑器系统** | | | | | |
| **实验时间** | **2021.11.11** | | **实验地点** | **DS1421** | | |
| **实验成绩** |  | | **实验性质** | **验证性 ■设计性 □综合性** | | |
| 教师评价： | | | | | | |
| 一、实验目的  利用Java输入输出系统、GUI程序设计等知识，实现文本编辑器的设计与实现。提升运用面向对象程序设计思想解决实际问题的能力。 | | | | | | |
| 二、实验项目内容  实现文本编辑的各项所需功能。主要功能包含以下内容：  （1）实现文件的新建、保存等基本功能。  （2）可以打开已保存的文件，读取文件的，实现查看，编辑等操作。  （3）显示、编辑修改文本信息等主要处理功能。可对相关信息进行复制，剪切，粘贴等操作。  （4）其他扩展功能可自由发挥。 | | | | | | |
| 1. 实验过程或算法（要求：写明设计思想、程序的结构、功能关系图、类的说明和类之间的关系图、程序主要执行流程图，最后是核心源代码。） 2. 设计思想及程序结构 3. 根据GUI的使用及设计方法，主要继承了JFrame类来实现窗口的设计，用ActionListener的接口来实现监听。在窗口中进行一层一层的从顶向下的设计，先是菜单栏menuBar（JMenuBar类）,其中包括了JMenu的四个菜单项File，Edit,Format,Help。而在File中有菜单项(JMenuItem类）new,new window,open,preserve,exit几个功能；Edit中有菜单项(JMenuItem类）cancel,recover,cur,copy,paste,delete几个功能；Format中有菜单项(JMenuItem类）wordformat一个功能，Help中有菜单项(JMenuItem类）about一个功能。 4. 在实现时则自底向上，先实现每一个菜单项中各个小菜单项的布局等，在将他们按一定的顺序及显示格式加入到窗口中，最后对窗口上的按钮等设置监听及功能实现。其中点击wordformat及about这两个菜单项会弹出新的窗口，这两个窗口也是继承自JFrame类实现的，为使代码清晰，分别写在同一个包下的另两个Java文件ShowFormat和ShowHelp中。ShowLine则用于实现在文本编辑器的左侧显示行号。   （二）功能关系图  413717A22AF82C4620B98EC9C01D48A7  （三）类的说明及类之间的关系图  TxtManager,ShowFormat,ShowHelp均继承自同一个父类JFrame。 TxtManger为最主要的类，用于实现整个文本编辑器的大窗口，菜单栏显示及具体功能。ShowFormat用于实现点击文本编辑器菜单栏Format的wordformat项时弹出的窗口，用于调整字体。ShowHelp则用于实现点击文本编辑器菜单栏Help的about项时弹出的窗口,显示about的内容。ShowLine则用于实现在文本编辑器的左侧显示行号。TxtManager中会调用ShowFormat,ShowHelp，ShowLine这几个类。  D2B26447559548F67C2D4485EF265B40   1. 程序主要执行流程图   CAD1269130C6BABF88DD05B705ACBE39   1. 核心源代码   主程序  **public** **class** TxtManager **extends** JFrame **implements** ActionListener{  **private** JMenuBar menuBar; //菜单栏    **private** JMenu me\_File,me\_Edit,me\_Help,me\_Format; //菜单项    **private** JMenuItem it\_new,it\_newwindow,it\_open,it\_save,it\_exit; //file菜单项内容    **private** JMenuItem it\_undo,it\_redo,it\_cut,it\_copy,it\_stick,it\_delete; //edit菜单项内容    **private** JMenuItem it\_about; //help菜单项内容    **private** JMenuItem it\_word\_format; //format菜单项内容    //文本区域  **private** **static** JTextArea *edit\_text\_area*; // JTestArea()建立一个JTextArea对象，使用预设的模式、空字符串、0行、0列。  //文本滚动条  **private** JScrollPane scroll\_bar;  //撤销管理  **private** UndoManager um;  //系统剪切板  **private** Clipboard clipboard;  //文件选择器  **private** JFileChooser fileChooser;    **public** TxtManager()  {  initMenuBar();  initEditText();  initListener();    **this**.setJMenuBar(menuBar);  **this**.setSize(700,500); //设置窗体大小  **this**.add(scroll\_bar);  **this**.setTitle("Serena's TXT.Manager"); //设置窗体标题  **this**.setVisible(**true**); //设置窗体是否可见，true：可见，false：不可见  **this**.setLocationRelativeTo(**null**); //窗体位于正中间位置  **this**.setDefaultCloseOperation(JFrame.***DISPOSE\_ON\_CLOSE***); // //设置窗体关闭方式  }    **private** **void** initMenuBar() //初始化菜单栏  {  menuBar = **new** JMenuBar();    //file菜单项  me\_File = **new** JMenu("File(F)");  me\_File.setMnemonic(KeyEvent.***VK\_F***); // 设置快捷键-Alt+f:打开  it\_new = **new** JMenuItem("new"); //新建  it\_newwindow = **new** JMenuItem("new window"); //新窗口  it\_open = **new** JMenuItem("open"); //打开  it\_save = **new** JMenuItem("preserve"); //保存  it\_exit = **new** JMenuItem("exit");  me\_File.add(it\_new); //将上述菜单项加入菜单中  me\_File.add(it\_newwindow);  me\_File.add(it\_open);  me\_File.add(it\_save);  me\_File.add(it\_exit);  //edit菜单项  me\_Edit = **new** JMenu("Edit(E)");  me\_Edit.setMnemonic('e'); // 设置快捷键-Alt+e:打开  it\_undo = **new** JMenuItem("cancel"); //撤销  it\_redo = **new** JMenuItem("recover"); //恢复  it\_cut = **new** JMenuItem("cut"); //剪切  it\_copy = **new** JMenuItem("copy"); //复制  it\_stick = **new** JMenuItem("paste"); //粘贴  it\_delete = **new** JMenuItem("delete"); //删除  me\_Edit.add(it\_undo);  me\_Edit.add(it\_redo);  me\_Edit.add(it\_cut);  me\_Edit.add(it\_copy);  me\_Edit.add(it\_stick);  me\_Edit.add(it\_delete);    //help菜单项  me\_Help = **new** JMenu("Help(H)"); //帮助  me\_Help.setMnemonic('h'); // 设置快捷键-Alt+h:打开  it\_about = **new** JMenuItem("about"); //关于  me\_Help.add(it\_about);  //format菜单项  me\_Format = **new** JMenu("Format(O)"); //格式  me\_Format.setMnemonic('o');  it\_word\_format = **new** JMenuItem("wordformat(F)"); // 字体 CTRL\_MASK = ctrl键  me\_Format.add(it\_word\_format);  //加入菜单栏  menuBar.add(me\_File);  menuBar.add(me\_Edit);  menuBar.add(me\_Format);  menuBar.add(me\_Help);  }  **private** **void** initEditText() {  //以下代码将文本区放置于滚动面板，滑动面板的滚动条能浏览文本区  *edit\_text\_area* = **new** JTextArea();  scroll\_bar = **new** JScrollPane(*edit\_text\_area*);    scroll\_bar.setRowHeaderView(**new** ShowLine()); //显示行号    scroll\_bar.setVerticalScrollBarPolicy(JScrollPane.***VERTICAL\_SCROLLBAR\_AS\_NEEDED***);  scroll\_bar.setHorizontalScrollBarPolicy(JScrollPane.***HORIZONTAL\_SCROLLBAR\_AS\_NEEDED***); //确定水平滚动条何时显示在滚动窗格上。选项有：ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_AS\_NEEDED 需要时出现    um = **new** UndoManager();  clipboard = **this**.getToolkit().getSystemClipboard();  }    **private** **void** saveFile()  {  File file =**null**;  **int** result; //int型值，showSaveDialog返回值用来判断用户是否选择了文件或路径。  fileChooser = **new** JFileChooser("C:\\Users\\62473\\Desktop\\"); //打开用户默认目录  result = fileChooser.showSaveDialog(rootPane); // 设置Dialog的根View根布局 显示对话框    **if**(result == JFileChooser.***APPROVE\_OPTION***) {  file = fileChooser.getSelectedFile(); // 若点击了确定按钮，给file填文件路径  }  **else** **return**; //按取消键后不会运行下面代码    **try**{  OutputStreamWriter write = **new** OutputStreamWriter(**new** FileOutputStream(file),"UTF-8"); // 对字符进行编码转换 输出流  BufferedWriter writer = **new** BufferedWriter(write);  **for** (String value : *edit\_text\_area*.getText().split("\n")) { // 实现在JTextArea多行如何转换到实际文本多行  writer.write(value);  writer.newLine();//换行  }  writer.close();  }  **catch**(IOException e) {  e.printStackTrace();  }  }    **private** **void** openFile() {    File file =**null**;  **int** result;  fileChooser =**new** JFileChooser("C:\\Users\\62473\\Desktop\\");  result = fileChooser.showOpenDialog(rootPane);    **if**(result == JFileChooser.***APPROVE\_OPTION***) {  file = fileChooser.getSelectedFile(); // 若点击了确定按钮，给file填文件路径  }  **else** **return**; //按取消键后不会运行下面代码    **if**(file.isFile() && file.exists()) {  **try** {  *edit\_text\_area*.setText(""); //清空目前文本内容  InputStreamReader inputStreamReader = **new** InputStreamReader(**new** FileInputStream(file),"UTF-8");  BufferedReader reader = **new** BufferedReader(inputStreamReader);  String readLine;  **while** ((readLine = reader.readLine()) != **null**) {  *edit\_text\_area*.append(readLine+'\n');  }  reader.close();  }  **catch** (IOException e) {  e.printStackTrace();  }  }    }    **public** **void** actionPerformed(ActionEvent e) {  //接口ActionListener里面定义的一个抽象方法，所有实现这个接口的类都要重写这个方法。  //getSource 返回发生事件的事件源对象的引用  **if**(e.getSource()==it\_about)  {  **new** ShowHelp();  }  **else** **if** (e.getSource()==it\_word\_format) {  **new** ShowFormat();  }  **else** **if** (e.getSource()==it\_new) {  **this**.saveFile(); //新建文件前保存原文件  **new** TxtManager();  **this**.dispose();  }  **else** **if** (e.getSource()==it\_newwindow) {  **new** TxtManager(); //添加新窗口，父窗口不会退出  }  **else** **if** (e.getSource()==it\_exit){  **this**.saveFile();  **this**.dispose(); //退出询问(目前有无更改都会弹出保存窗口)  }  **else** **if** (e.getSource()==it\_save) {  **this**.saveFile();  }  **else** **if** (e.getSource()==it\_open) {  **this**.openFile();  }  **else** **if** (e.getSource()==it\_undo&&um.canUndo()) { //撤销可以撤销是撤销上一步文本操作  um.undo();  }  **else** **if** (e.getSource()==it\_redo&&um.canRedo()) { //恢复就是恢复上一步文本操作(需要被撤销)  um.redo();  }  **else** **if** (e.getSource()==it\_copy) {  String temptext = *edit\_text\_area*.getSelectedText(); //得到选取文本  StringSelection editText = **new** StringSelection(temptext); //创建能传输指定 String 的 Transferable  clipboard.setContents(editText,**null**); // 将剪贴板的当前内容设置到指定的 transferable 对象，  // 并将指定的剪贴板所有者作为新内容的所有者注册。  }  **else** **if** (e.getSource()==it\_cut) {  String temptext = *edit\_text\_area*.getSelectedText();  StringSelection editText = **new** StringSelection(temptext); //构建String数据类型  clipboard.setContents(editText,**null**); //添加文本到系统剪切板  **int** startes = *edit\_text\_area*.getSelectionStart(); //复制+删除  **int** endes = *edit\_text\_area*.getSelectionEnd();  *edit\_text\_area*.replaceRange("",startes,endes); // 替换部分文本  }  **else** **if** (e.getSource()==it\_stick) {  Transferable contents = clipboard.getContents(**this**);  DataFlavor flavor= DataFlavor.***stringFlavor***;  **if**( contents.isDataFlavorSupported(flavor))  {  **try**  {  String str;  str = (String)contents.getTransferData(flavor);  *edit\_text\_area*.append(str); // 追加文本到文档末尾  }  **catch**(Exception ex)  {  ex.printStackTrace();  }  }  }  **else** **if** (e.getSource()==it\_delete) {  **int** start= *edit\_text\_area*.getSelectionStart(); //删除  **int** end = *edit\_text\_area*.getSelectionEnd();  *edit\_text\_area*.replaceRange("",start,end);  }  }    **public** **void** initListener()  {  // 菜单iter监听  it\_new.addActionListener(**this**);  it\_newwindow.addActionListener(**this**);  it\_open.addActionListener(**this**);  it\_save.addActionListener(**this**);  it\_exit.addActionListener(**this**);  it\_undo.addActionListener(**this**);  it\_redo.addActionListener(**this**);  it\_cut.addActionListener(**this**);  it\_copy.addActionListener(**this**);  it\_stick.addActionListener(**this**);  it\_delete.addActionListener(**this**);  it\_word\_format.addActionListener(**this**);  it\_about.addActionListener(**this**);    //注册撤销/恢复可编辑监听器  *edit\_text\_area*.getDocument().addUndoableEditListener(**new** UndoableEditListener(){  **public** **void** undoableEditHappened(UndoableEditEvent e) {  um.addEdit(e.getEdit());  }  });    }    **public** **static** JTextArea getEdit\_text\_area() {  //返回文本编辑区给其他窗口  **return** *edit\_text\_area*;  }    **public** **static** **void** main(String[] args) {    TxtManager createone = **new** TxtManager();  }}  其余三个类及详情请见附件源代码 | | | | | | |

|  |
| --- |
| 1. 实验结果及分析和（或）源程序调试过程（包含测试运行界面截图，总结体会等） 2. 实验结果 3. **创建一个文本编辑器**      1. **保存文件**        1. **打开文件**   **65P%9(CE1@SNGI31(~Z50$M**   1. **改变字体**        1. **打开about**     **（2）调试过程**  **上网查阅资料学习GUI各个组件的使用方法，将他们组合到一起以实现功能。** |