面向对象程序设计（Java）

实验报告

实验四 IO实验

|  |  |
| --- | --- |
| 学院名称 | 计算机科学与技术学院 |
| 专业班级 | 信安1802 |
| 学生姓名 | 侯天铭 |
| 学号 | 5120182377 |
| 指导教师 | 刘涌 |

2019年 4月

1. **实验目的**

练习输入/输出流的创建、转换及数据访问设计方法。

**二、实验内容**

设计一个类似Windows记事本的Application应用程序。

**三、概要设计**

1. **需求分析过程**
2. **用户对界面的需求**

通过对Windows自带的记事本程序的分析，用户对界面的需求主要可分为：

程序左上方显示当前打开的文件名，若是初始状态或点击新建后则显示“无标题-记事本”；

标题往下有一行菜单栏，包括了“文件、编辑、格式、查看、帮助”几个可选项；

最后是一大块用于输入数据的区域，它占据该程序窗体剩下的全部空间，并且应该具有滚动条，用于上下左右拖动查看文件内容。

1. **用户对使用过程的需求**

对于一个记事本程序，其应具备的同时也是用户最常使用的基本功能包括：新建，打开，保存，另存为。因此在分析用户对使用过程的需求时主要从这几个功能入手。

**新建：**若是直接打开记事本程序（不打开任何文件）点击“新建”将不会有任何变化；若是使用记事本打开了某个文件但不做任何操作，点击“新建”，则界面上的当前内容会清空，并且标题发生改变，即新建了一个新的空白的文件；若使用记事本打开某个文件并进行了修改或直接打开记事本进行了内容的输入，再点击“新建”，系统会提示“是否保存修改”。

**打开：**当用户点击“打开”后，程序会弹出一个对话框供用户选择需要打开的文件。若选择的文件不存在，则会提示“未找到选择的文件”；若选择文件存在，则在当前程序界面中会显示出用户选择打开的文件的内容。

**保存：**当用户点击“保存”后，程序会首先判断此时是否是在修改一个打开的文件，若是，则会直接将修改保存到当前打开的文件，而不弹出对话框。否则会弹出一个对话框供用户选择文件保存的路径。当用户选择好路径并输入了需要保存的文件名后，点击确定，则会在相应位置生成该文件，并将程序中的内容写入其中。当然若是点击取消，则程序什么也不会做。

**另存为：**用户在任何时候点击“另存为”都会弹出一个对话框来选择文件保存的路径。

最后简单说说“退出”，若用户在当前程序中进行了内容的修改操作，则点击“退出”时会提示是否保存修改，否则直接退出。

1. **总体设计方案**
2. **界面设计**

首先是刚刚打开程序，左上角显示默认标题“无标题-记事本”，往下是一行菜单栏，包括“文件、编辑、格式、查看、帮助”这几项Menu，其中文件栏下还设有“新建、打开、保存、另存为、退出”这几项MenuItem分别提供对应的功能。然后说窗体的核心部分：在中央面板处，首先是添加了一个容器Container：JScrollPane，目的是在界面中提供一个可以上下左右拉动的滚动条。接着为了使得用户能够在界面中输入多行数据，在该容器中添加了一个组件：JTextArea。至此，此记事本程序基本的界面设计就完成了，用户已经可以像使用Windows自带的记事本程序一样打开该程序写入一些文字了。

1. **事件设计**

与上次设计的计算器不同，本次的记事本程序没有Button组件，面板内只有一个输入框，事件包括菜单Menu下MenuItem对应的菜单actionperformed事件以及textArea对应的键盘typed事件。作为一个记事本来讲，该程序包括的基本功能是：新建，打开，保存，另存为，以及退出。而这些记事本具有且用户需要使用的功能就是我们需要设计的事件，只不过在此基础上，为了获取用户对程序的操作，我们还设置了键盘typed事件来捕获用户对键盘的操作，以便为相同的按键在程序运行的不同状态设计具有细微差别的事件。这些事件应实现的功能如下：

**新建：**清空显示内容。

**打开：**打开一个已有的文件。

**保存：**将编辑的内容保存到指定的文件中。

**另存为：**同保存。

**退出：**退出（终止）该程序。

**键盘：**改变status的值（后续会提）

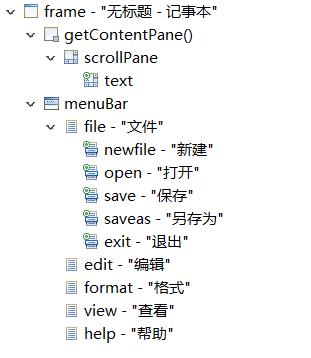
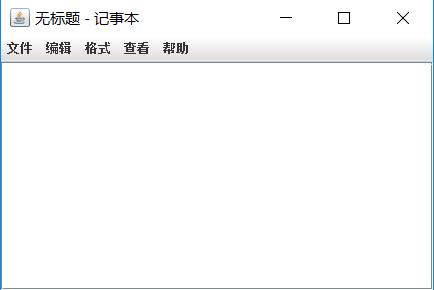
**四、详细设计**

**1、类的设计**

使用WindowBuilder创建类，输入类名Notepad，点击finish，自动生成的代码便创建了程序窗体的初始框架。随后，为了实现不同的功能，我们需要向该窗体中添加额外的要素。于是在Design窗口中将MenuBar、Menu、MenuItem三个菜单相关的类，JScrollPane这个容器以及JTextArea这一组件拖动到合适的位置填充在窗体中，形成一个记事本程序的外观。在这些操作之后，返回Source窗口，可以看到Notepad类中新增了很多代码，对应着我们对窗体的操作。随后在Source窗口中手动写入相关的事件代码，便完成了使用WindowBuilder创建整个Notepad类的过程。

**2、界面设计**

界面中层次结构：最外层为窗体frame，frame内包括了getContentPane面板，和menuBar菜单栏。其中getContentPane中包含有一个容器scrollPane（提供界面的滑动功能），在该容器中填充了一个textArea文本框组件，用于输入数据；对于menuBar而言，其下包含了文件、编辑、格式、查看、帮助5项menu，并且对于文件这一menu，其下还包含有新建、打开、保存、另存为、退出这5项menuItem。

如图：对应窗体为：

1. **事件设计**

本次设计的记事本程序所具有的事件，具体为菜单栏中文件菜单下的“新建、打开、保存、另存为、退出”这些菜单事件以及键盘typed事件。并且考虑到用户不管是用键盘还是鼠标都能使用该程序，因此将所有的菜单事件都设置为actionperformed事件。（由于能力及时间有限，某些事件的设计还存在一些不足之处）

在设计各个事件之前，考虑到用户操作记事本时在不同的状态按下相同的选项会产生不同的效果，因此首先定义了一个整型属性status，初始化为0，取值范围为0-3，依次代表：未保存未编辑，未保存已编辑，已保存未编辑，已保存已编辑。每一个菜单事件在不同的status下会有细微的不同。而键盘事件的设计就是为了捕获用户对键盘的操作，改变status的值。

1. **新建：**当用户点击新建时，首先判断此时的状态，若status=0或status=2，说明此时状态为：打开记事本程序，未做操作，或使用记事本打开了一个文件，但还没有进行任何修改。此时点击“新建”，直接将界面显示内容清空即可，即text.setText(“”)。若此时status=1或status=3，说明此时状态为：打开记事本程序，输入了一些数据，或使用记事本打开了一个文件，并做了修改操作。此时点击“新建”，将弹出一个对话框，询问用户是否保存已做的修改，使用一个变量ret储存JOptionPane.showOptionDialog()方法的返回值，判断用户的选择。若用户点击“否”或“取消”则程序不做任何操作；若用户点击“是”，则再次弹出一个对话框，供用户选择该文件存储的路径，以及输入此文件保存下来的文件名。同理，使用另一个变量存储showSaveDialog()方法的返回值，判断用户的选择，若此时用户点击取消，程序照样不做任何操作；若用户选择了保存路径并输入了文件名，然后点击“保存”则将该文件保存到指定位置；同时由于JFileChooser类本身提供的功能，用户若不输入文件名是无法进行保存的。最后，在保存完后，再进行“新建”操作，即将显示内容清空。最后的最后，将status置为2，表示已保存，是为了在用户已经选择了保存之后再点击新建不会再弹出对话框。
2. **打开：**当用户点击“打开”，首先还是得进行状态判断。若status=0或status=2，即文本处于未编辑的状态，此时点击“打开”，直接弹出打开选择对话框，用户选择需要打开的对象，点击“打开”，即可将目标文件内容读入到当前记事本文本框中，并且在最后将status置为2，表示已经保存。另外两种情况：status=1或status=3，均为已编辑的状态，即用户在原有文本的基础上作出了一些修改，那么当用户点击“打开”后，首先要弹出对话框询问用户是否保存更改，若用户点击取消，则不进行后续的操作；若用户点击不保存，则不弹出保存对话框，直接弹出打开对话框，用户选择需要打开的文件即可将文件内容显示在当前程序文本框中，同样需要将status置为2；当用户点击保存时，首先要弹出保存对话框，在保存对话框中若用户点击取消，那么之后将仍然不进行打开操作，若用户选择了保存路径且输入了保存文件名，点击保存后，则将该文件保存到指定的位置，当然最后还是得将status置为2。
3. **保存：**当用户点击“保存”后，照样先判断此时的状态，若status!=2即只有当status=2即打开一个文件但未进行任何操作时什么也不做。（实际上的记事本程序应该是将当前内容保存到当前文件，所以没弹出对话框，但是我没能实现）其余情况均弹出一个对话框，供用户选择文件保存的路径以及输入文件名，后续具体流程同**新建**中所述。最后需要将status置为2，表示已保存，那么比如说在用户使用记事本打开一个文件进行修改但是选择了保存之后再点击新建，便会直接清空显示内容，而不会弹出对话框。
4. **另存为：**对于另存为而言，不存在状态判断，用户在任何时刻点击另存为均会调用showSaveDialog方法弹出保存对话框进行保存操作。当然保存之后还是需要将status置为2。
5. **退出：**由于不知道如何设计窗体右上角“×”的事件，因此设计了这个“退出”事件。用户点击“退出”，首先判断状态。若status=1或status=3，说明此时用户对文本进行了编辑，并且并没有保存（若保存成功status会变成2），所以从人性化角度来考虑，用户对文本进行了编辑但是却忘记了保存就点击“退出”，那理所应当弹出一个对话框提醒用户“是否将更改保存”，若用户点击不保存，则程序直接退出；若用户点击取消，则关闭提示框，恢复原界面，不作任何操作；若点击保存，则再次弹出保存对话框，并在成功保存后将status置为2，并退出程序；若status=0或status=2，说明用户并没有对文本进行编辑，或是进行编辑后已经进行了保存，那也就没有必要再提示保存了，用户点击“退出”，程序直接结束即可。
6. **键盘：**键盘事件最为简单，设计目的是改变status的值。当用户按下键盘的时候，说明用户在进行输入操作，首先判断当前状态。若status=0，说明此时为一个空文本，当用户敲键盘输入数据之后，status变为1，即未保存已编辑状态；若status=2，说明此时是用记事本打开了一个文件，当用户敲键盘输入数据之后，status变为3，表示已保存已编辑（实际上并没有保存，只是说此时是打开了一个文件进行了编辑，不是创建的新文件）。然而这个状态变化存在一个缺陷：即若是用户先输入一些数据，然后又删除这些数据之后，status仍然是1或者3，也就是说在退出或新建的时候仍然会提示是否保存，点击保存也仍然会弹出保存框，但实际上文本内容根本没有更改，也不需要保存。

**五、实验结果**

本次实验的测试过程为对“新建、打开、保存、另存为、退出”这五大功能的测试：  
**1、新建：**

**（1）**当直接打开该程序不作任何操作时，点击“新建”，无明显变化；

**（2）**使用该程序打开某文件后，再次点击“新建”，文本框显示内容清空；

**（3）**当打开该程序并进行输入操作后，再次点击“新建”，弹出对话框询问是否保存更改，点击“取消”，则无变化；点击“否”，则直接将显示内容清空；点击“是”，则在进行了后续的保存操作保存成功后将显示内容清空。

**（4）**使用该程序打开某文件，进行修改操作后点击“新建”，提示是否保存修改，后续与上（3）相同。

**结论：**该程序基本实现了记事本的“新建”功能。

**2、打开：**

**（1）**直接打开程序不作任何操作，点击“打开”，直接弹出打开对话框，选择要打开的文件。

**（2）**使用该程序打开某文件后，再次点击“打开”，同样直接弹出打开对话框进行选择。

**（3）**打开程序，进行输入操作后，再次点击“打开”，首先提示是否保存修改，选择“取消”，无后续；点击“否”，弹出打开对话框；若点击“是”，则先弹出保存对话框，若此时取消，仍无后续，仅当选择保存后，弹出打开对话框。

**（4）**使用该程序打开某文件，进行修改操作后点击“打开”，提示是否保存修改，后续与上（3）相同。

**结论：**该程序基本实现了记事本的“打开”功能。

1. **保存：**
2. 使用该程序打开某文件后，点击“保存”，无变化。
3. 使用该程序打开某文件并进行修改后，点击“保存”，弹出保存选择框，进行保存操作。
4. 打开程序不作操作，即当文本框为空时点击“保存”，仍弹出保存选择框。

**结论：**该程序基本实现了记事本的“保存”功能。

**4、另存为：**

在任何时刻点击“另存为”，均弹出保存选择对话框，进行保存操作。

**结论：**该程序实现了记事本的“另存为”功能。

**5、退出：**

**（1）**打开程序或使用该程序打开一个文件，点击“退出”，程序直接退出。

**（2）**打开程序或使用该程序打开一个文件后进行修改操作，点击“退出”，提示是否保存，点击“取消”，则程序不结束；点击“否”，则不作保存便退出；点击“是”，则在进行了保存操作后退出程序。

**结论：**该程序基本实现了记事本“退出”的功能。

**六、讨论与分析**

**对课程知识点的理解：**在本次实验中，我们在使用WindowBuilder创建类的时候，首先是在面板内加入了菜单栏MenuBar，然后又在面板内添加了一个容器Containers，然后再在容器内添加了文本框TextArea。那么在一过程中，我对于窗体中的菜单，容器，以及各组件之间的层次关系认识得较之前更为清楚了。

**对设计过程的理解：**其实对于本次设计的记事本程序，其具有的“新建，打开，保存，另存为，退出”这些功能如果仅仅是要实验它们，其实是相当简单的。但是在这个过程总最难的也是最重要的是要考虑到用户在使用过程中的不同状态，即对于每一个事件，在程序不同的状态下，是有细小的差别的。因此在设计过程中，状态status的转换贯穿了每一个事件，非常重要。

**实验结果的分析：**本次设计的记事本程序大体上能够完成Windows记事本程序的各项基本功能，但缺陷还挺多，将在下一点叙述。

**需要引起注意的问题：**

1. **标题无法改变：**按道理来说，打开记事本程序的时候，左上角标题为“无标题-记事本”没错，但是当使用记事本打开一个文件后，或者是在将初始的文本修改保存后，左上角的标题都应该改变，分别为：打开的文件名，保存的文件名（实际上应该是在保存后，直接用记事本打开该文件，并改变标题名字）。但是因为窗体标题是frame的title，而在后面根本无法访问window.frame，所以不知道如何进行标题的修改。
2. **“保存”事件设计不完善：**“保存”事件设计的不完善主要是表现为：当使用记事本打开一个文件后，若不进行任何操作，点击“保存”，在我设计的程序里，在这种状态下是什么也没发生。只有当作出修改后点击“保存”才会弹出保存对话框。但是实际上，只要是打开了一个文件，无论做不做修改，当点击“保存”的时候都不应该弹出保存对话框，而是直接将当前内容保存到当前文件。但是这个过程，需要获取当前打开文件的路径，暂时未能实现。

**七、附录：关键代码**

本次实验关键代码有二：

1、状态status：

（1）判断状态的变量status：**private** **int** status = 0;//0未保存未编辑；1未保存已编辑；2已保存未编辑；3已保存已编辑。

（2）键盘事件对应的状态转换：**public** **void** keyTyped(KeyEvent e) {

//判断初始状态

**if**(status == 0)

status = 1;//转化状态

**else** **if**(status == 2)

status = 3;

}

2、输入\输出流的创建及使用：

（1）输入流：

File selfile = jfc.getSelectedFile(); //获取打开目标

**try** {

FileReader reader = **new** FileReader(selfile); //创建连接源的流（从哪儿读）

BufferedReader br = **new** BufferedReader(reader);//BufferedReader封装（升级）FileReader，使之能够读取字符串

String s = br.readLine();//每次读一行

text.setText("");//下次open的时候，先将上次内容清空

**while**(s != **null**) {//读到null说明字符串已读完，停止

text.append(s+"\n");//由于一次读一行，因此显示一行后应换行

s = br.readLine();

}

br.close();//关闭流

} **catch** (FileNotFoundException e1) {//捕获异常

// **TODO** Auto-generated catch block

JOptionPane.*showOptionDialog*(**null**,"未找到选择的文件！","提示",JOptionPane.***CANCEL\_OPTION***,JOptionPane.***WARNING\_MESSAGE***,**null**,**null**,**null**);

} **catch** (Exception e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

（2）输出流：

File curfile = jfc.getSelectedFile(); //获取打开目标

String s = text.getText(); //获取目标内容

**try** {

FileWriter fw = **new** FileWriter(curfile); //创建指向目标curfile的流（写出到哪儿）

fw.write(s); //把s写入目标curfile

fw.close(); //关闭流

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}