

浙江大学实验报告

课程名称: JAVA 应用技术 指导老师: 翁恺 姓名: 吴雅芳

实验名称: Markdown 实验类型: 编程实现 学号: 3120104690

一、实验目的和要求

目的:

要求:

做一个 markdown 编辑器, 有左右两部分界面, 左边编辑, 右边实时显示对应的 HTML 的效果。在 MS Windows 下, 可以参考 MarkdownPad, 在 MacOS 下, 可以参考 MOU。

二、实验内容和原理 (必填)

JFrame 实现一个框架。

在框架里装入一个 JSplitPane, 左边为 EditPane, 右边为 ViewPane。

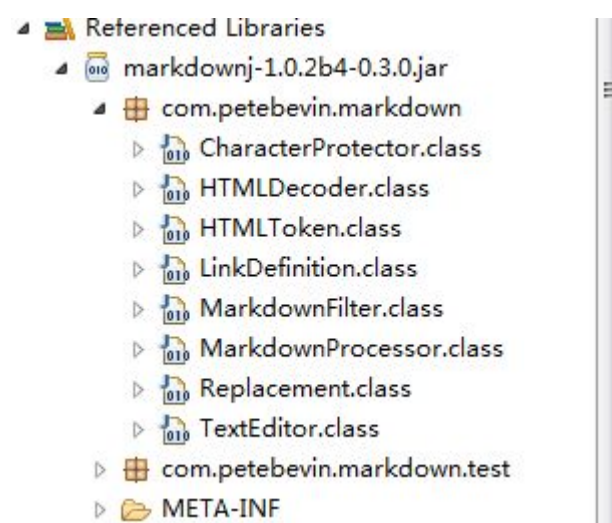
```
editPane.addObserver(viewPane);
```

添加观察者后, 当 editpane 内的 textarea 键盘按键 release 的时候判断文本是否发生改变, 通知给 viewpane 做 update 更新操作。更新操作则是把 editpane 中得到的字符通过 Markdown 的函数进行转化, 再显示在 viewpane 的文本域内。

从而实现左边编辑, 右边实时显示对的 HTML 的效果。

```
String content = (String)data;  
MarkdownProcessor processor = new MarkdownProcessor();  
//写文本  
previewLabel.setText(String.format("<html>%s</html>",  
processor.markdown(content)).replaceAll("src=\"", "src=\"file:"));
```

引入下列包来实现 markdown 语法。



三、主要仪器设备 (必填)

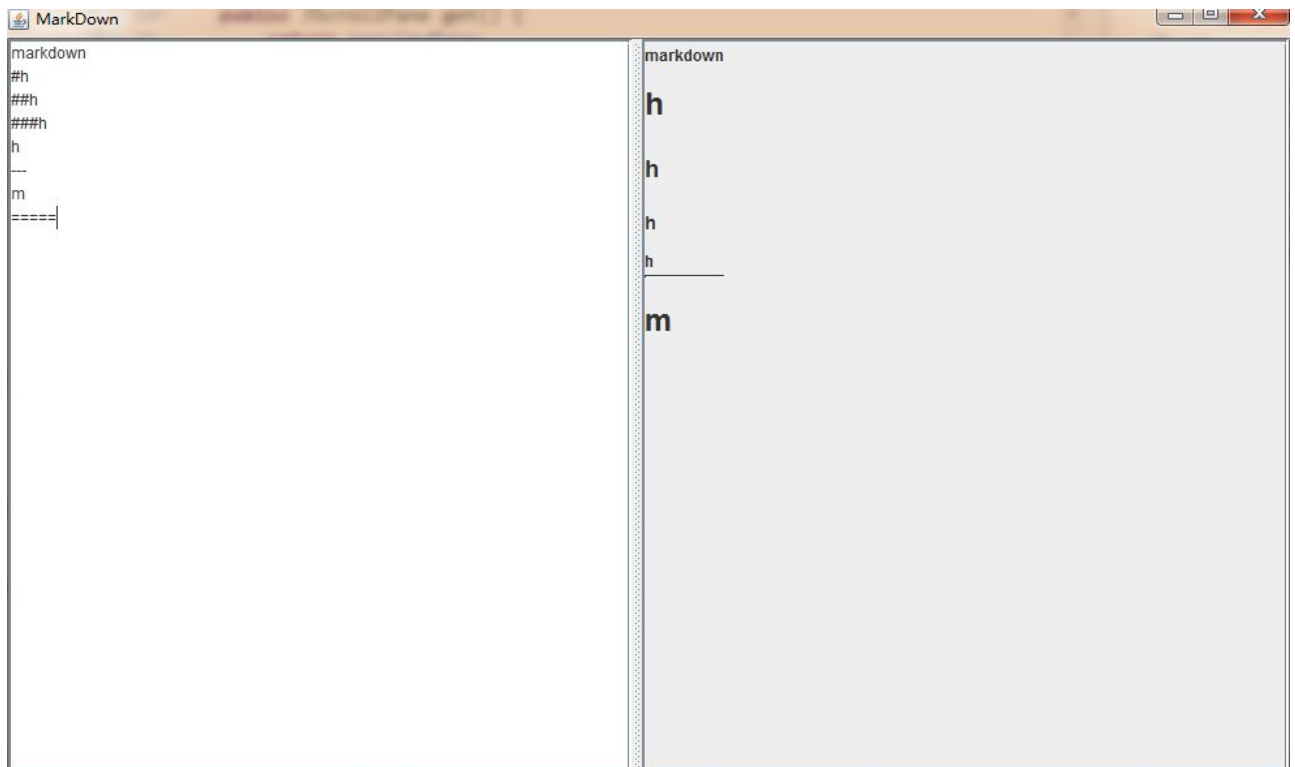
Eclipse

四、操作方法和实验步骤

- 1.复习 GUI 基本知识。
- 2.编写程序代码
- 3.调试运行
- 4.测试

五、实验数据记录和处理

运行结果如图：



六、实验结果与分析（必填）

运行正确。

七、讨论、心得

1.Swing 中的 JSplitPane(分割面板)

JSplitPane 的层次结构图:

```
--java.awt.Component
    --java.awt.Container
        --javax.swing.JComponent
            --javax.swing.JSplitPane
```

Split Pane(分割面板)一次可将两个组件同时显示在两个显示区中，若你想要同时显示在多个显示区显示组件，你必须同时使用多个 Split Pane。JSplitPane 提供两个常数让你设置到底是要水平分割还是垂直分割。这两个常数分别是：HORIZONTAL_SPLIT, VERTICAL_SPLIT.

```
JSplitPane splitPane = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,
    editPane.get(), viewPane.get());
```

2. 观察者方法

决定何时以及如何发起更新（updating）的那个“被观察者对象”被命名为 `Observable`。

`Observable` 类用一个标志（**flag**）来指示它自己是否改变。对于比较简单的设计来说，不用 `flag` 也是可以的；如果有变化，就通知所有的观察者。如果用 `flag` 的话，你可以使通知的时间延迟，并且由你来决定只在合适的时候通知观察者。但是，请注意，控制 `flag` 状态的方法是受保护的（`protected`），也就是说，只有（`Observable` 类的）派生类可以决定哪些东西可以构成一个变化，而不是由 `Observer` 派生类的最终用户来决定。

大多数工作是在 `notifyObservers()` 这个方法里完成的。如果没有将 `flag` 置为“已改变”，那 `notifyObservers()` 什么也不做；否则，它先清除 `flag` 的“已改变”状态，从而避免重复调用 `notifyObservers()` 的时候浪费时间。这些要在通知观察者之前完成，为了避免对于 `update()` 的调用有可能引起被观察对象的一个反向的改变。

然后 `notifyObservers()` 方法就遍历它所保存的观察者序列，并且调用每个观察者的 `update()` 成员函数。

初看起来，似乎可以用一个普通的 `Observable` 对象来管理更新。但是实际上办不到；为了达到这个效果，你必须继承 `Observable` 类，并且在派生类的代码里调用 `setChanged()` 方法。它就是用来将 `flag` 置为“已改变”的那个成员函数，这么一来，当你调用 `notifyObservers()` 的时候所有的观察者都会被准确无误的通知道。在什么地方调用 `setChanged()`，这取决于你程序的逻辑结构。