**Java Exercise 5：**

**18342138 郑卓民**

**目录：**

**1. 必做题两道**

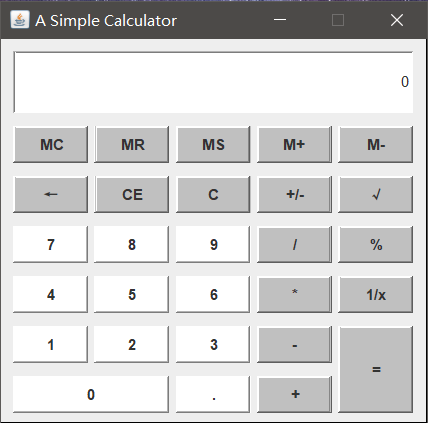
**2. 附加题一道**

完整代码不展示在本文档中，下文只阐述题目分析和结果截图分析。

**题目一：一个简单的计算器**

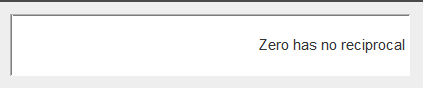
首先要查好某些按键的作用（M\*；CE；C；%），然后就是先init布局，然后写好对应的事件处理函数。

UI效果图：

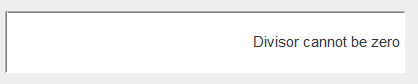


测试命令：

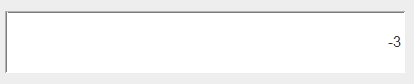
1. 1/0：



2. 3/0：



3. 1+2+3-9：



4. 测试M\*指令：

MS：存当前显示的数值

MR：读取存储器中的数值，并显示出来

MC：清除已存的数据

M-：用已存的数值减去当前显示的数值后，再将结果保存

M+：用已存的数值加上当前显示的数值后，再将结果保存

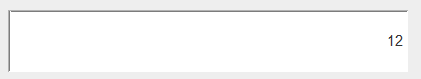
命令：

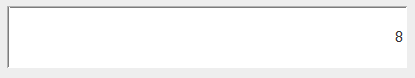
计算3\*4+2\*4=20：

先算3\*4，然后MS；

然后算2\*4，然后M+；

然后MR；







**题目二：一个简单的多人对话**

创建两个类分别为Server和Client：

对于Server：

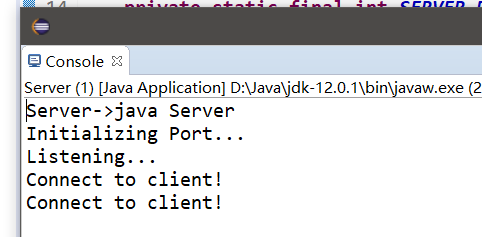
我们要处理两个问题：

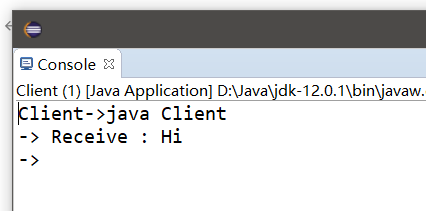
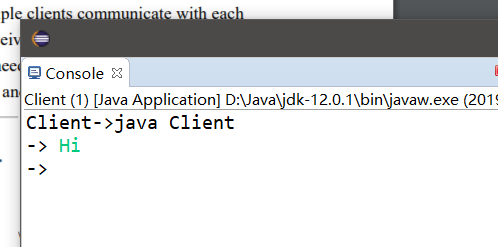
1. 利用一个监听线程来对于每个连接过来的Client，创建一个线程去分别处理通话内容；

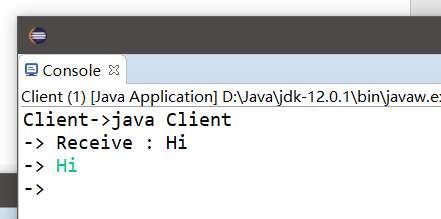
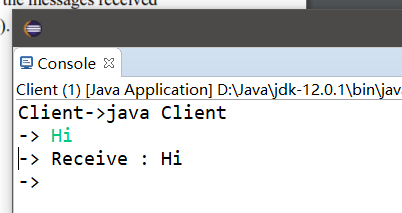
2. 利用一个发送信息线程来对于收到某个Client的信息后，将信息转发到其他Client中（此处的处理为：将1中创建的每个Client对应的线程存在一个list中，然后每次转发前，判断那个信息的来源是否不等于list中对应线程的name，若不等说明是不同Client，可发送）

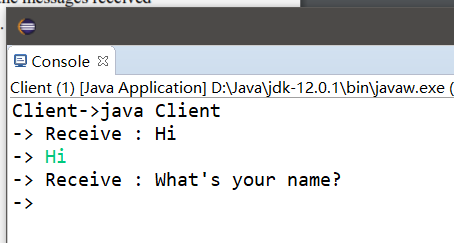
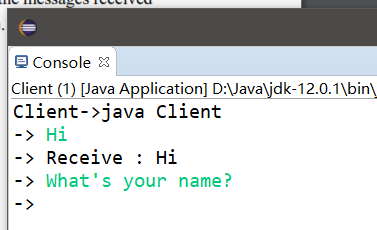
对于Client：  
主要就是要创建一个read线程来处理从server中接收到的信息；

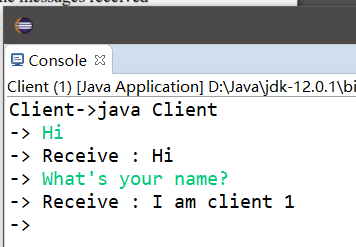
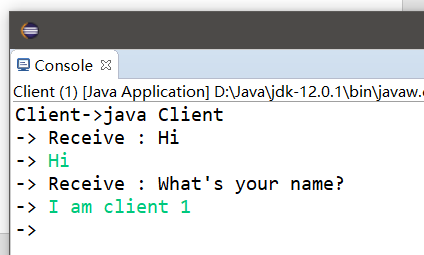
效果图：

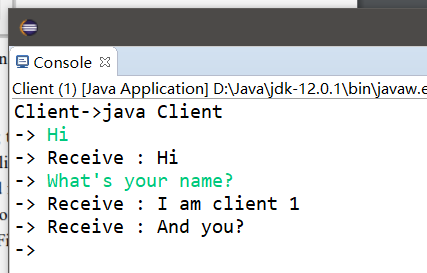
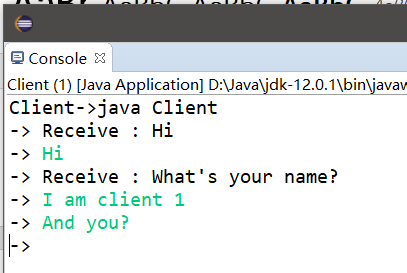


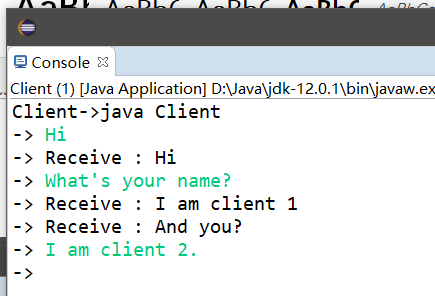
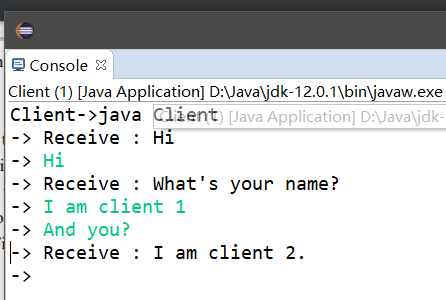












**附加题：**

**题目二：一个简单的radio按钮控制红绿灯**

核心思想就是通过一个标识（flag）来控制对应的圈圈（灯），要不要涂某个颜色（亮灯），然后每次点击radio button，都修改flag并且repaint一次。

