**说明文档**

姓名：陈前

学号：2015302580211

**一、概述**

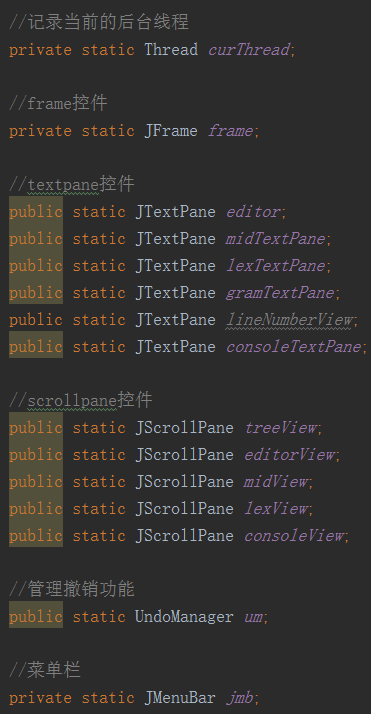
本次CMM解释器实验，我主要负责GUI部分，其中包括：

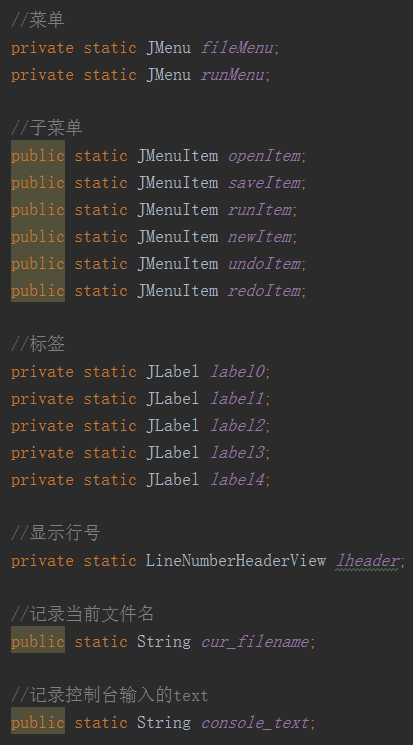
1. GUI的界面设计；
2. 界面显示行号；
3. 关键词高亮；
4. 新建、打开、保存文本文件功能，撤销和恢复文本功能，解决编码问题，统一使用GBK编码；
5. 运行功能；
6. 使用多线程分离复杂任务；

**二、具体内容**

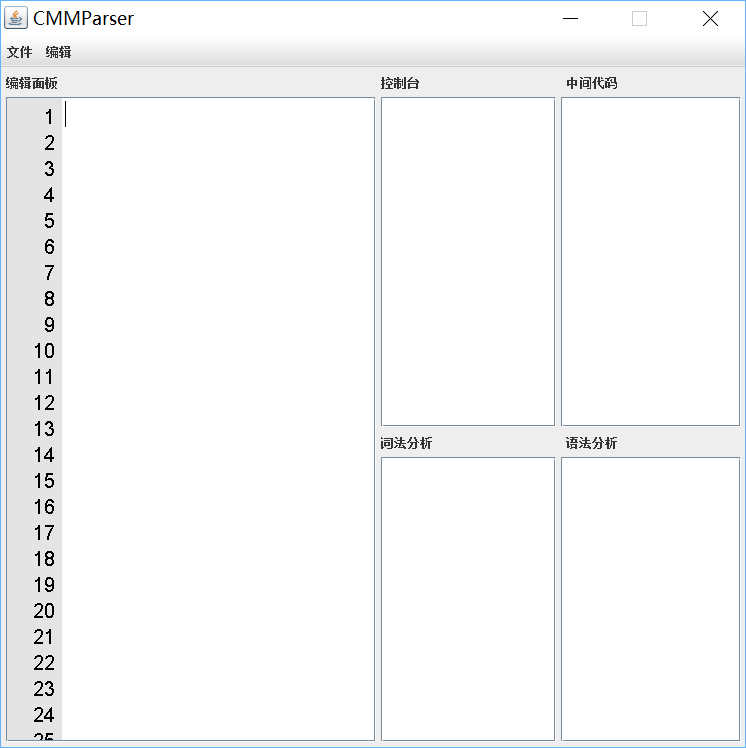
1. **GUI的界面设计**

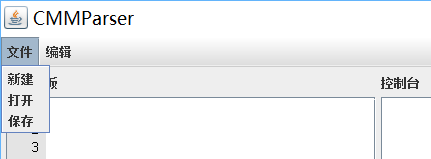
此次的GUI界面设计没有使用IDE提供的界面拖拽功能，完全由代码实现。新建一个MainForm类，类中包括以下属性：

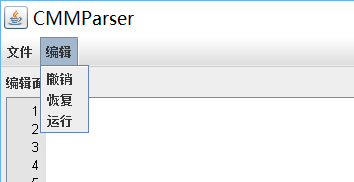




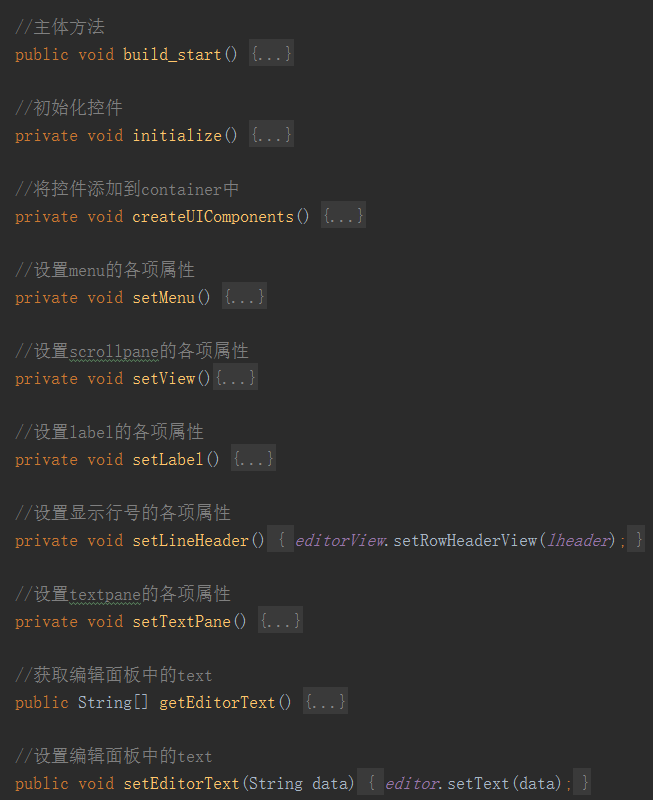
大部分为界面设计用到的控件，暂不详述。设计界面如下：







MainForm中方法如下：





其中：

1. 使用默认构造函数；
2. 主体方法为build\_start；
3. 在build\_start函数中调用initialize（初始化控件）和createUIComponents（将各个控件添加到container中）；
4. createUIComponents在添加控件之前调用setMenu，setView，setTextPane，setLineHeader，setLabel等函数具体初始化各个控件。
5. getEditorText获取编辑面板的内容，传出一个处理后的String数组；
6. setEditorText，setMidText，setLexText，setGramText用来设置编辑面板、中间代码、词法分析、语法分析面板的内容，包括添加Listener，设置位置大小等；
7. 语义分析遇到read时，调用readFunc函数阻塞后台线程，使用户在前端输入数据，监控用户输入，输入结束后（按下回车键）把输入内容传递给后台，继续运行后台线程；
8. 语义分析遇到write时，调用writeFunc函数在控制台输出结果；
9. CMM解析任一阶段出现问题，调用reportErr函数，在控制台输出结果；

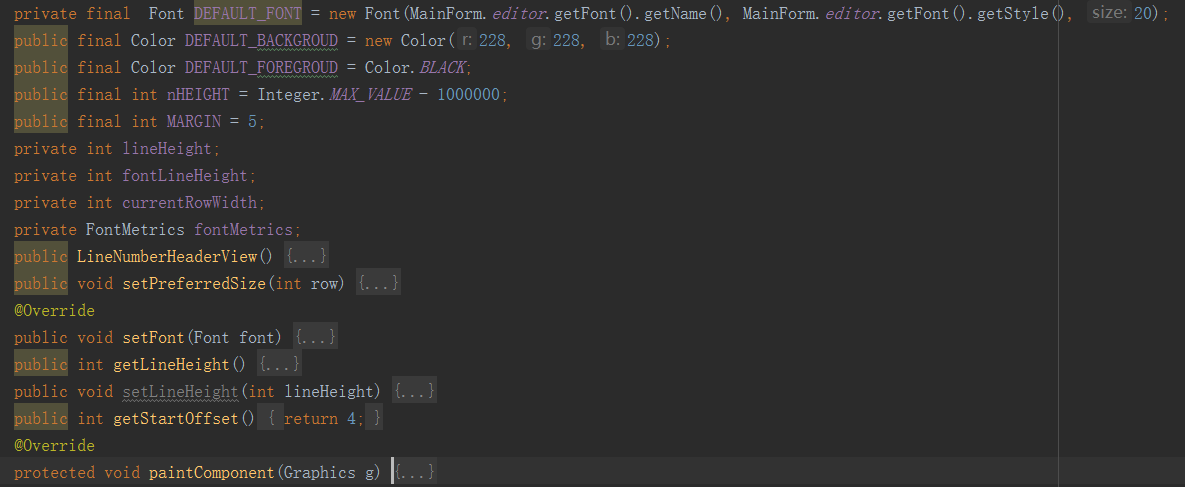
新建了一个IMainForm的接口类，提供以下方法：



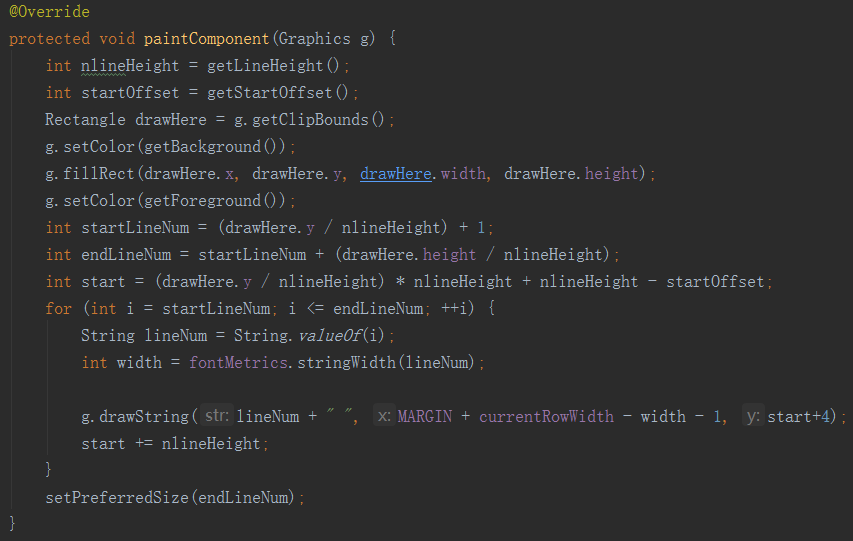
让之前创建的MainForm类实现这个接口，可以提高代码的封装性和安全性。

1. **界面显示行号**

新建一个LineNumberHeader类，继承自Jcomponent，属性与方法如下：

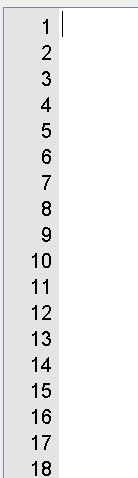


主要是重写了其中的paintComponent，对组件进行绘制，具体代码如下：



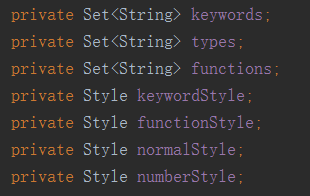
按照设定的行高、字体、颜色来绘制组件。

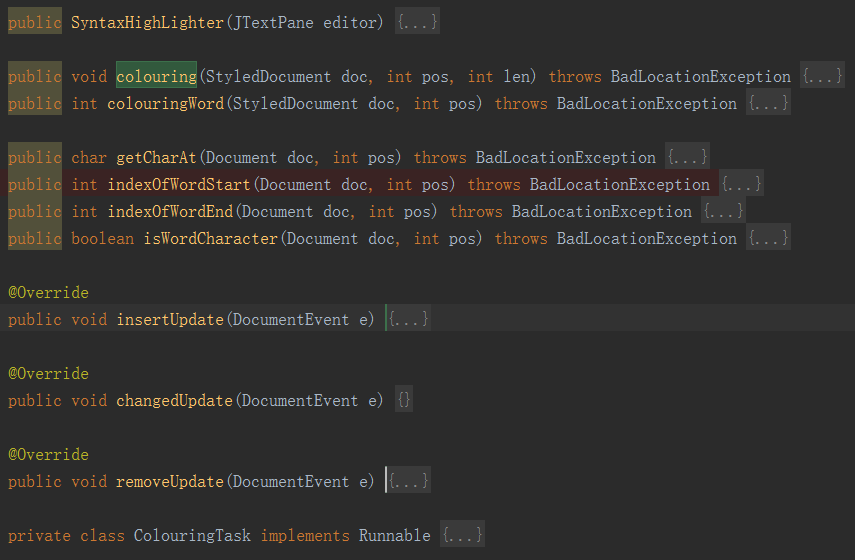
效果如下：



1. **关键词高亮**

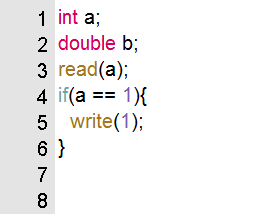
新建SyntaxHighLighter类，来实现DocumentListener接口，属性与方法如下：





重写了DocumentListener接口中的insertUpdate，changedUpdate，removeUpdate方法，当编辑面板增加或删除字符时调用Listener中对应的监听方法，在监听方法中再调用colouring函数，判断已经输入的组合在一起的单词、数字等是否为关键字（if，else，while，for，read，write，for，数字等），若是则新建一个线程进行内容替换（把普通的颜色替换为关键字的颜色），若不是则不替换颜色。

效果如下：

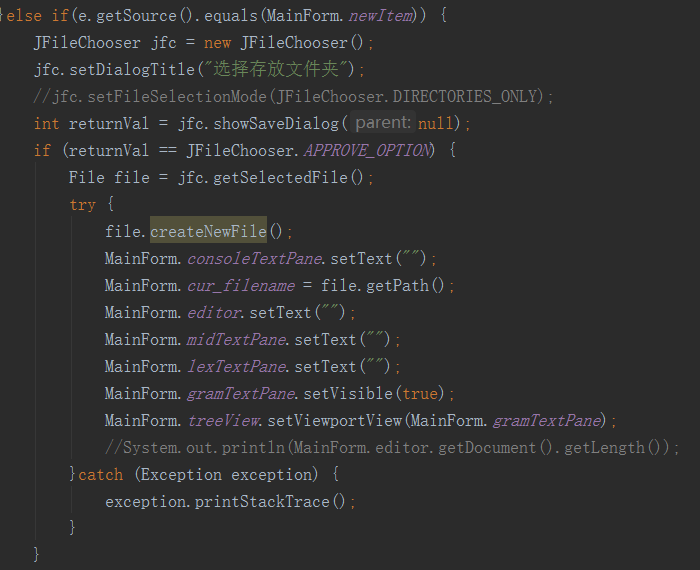


不仅仅是输入，打开已存在的文件时，高亮效果同样存在。

1. **新建、打开、保存文本文件功能，撤销和恢复文本功能，解决编码问题，统一使用GBK编码**

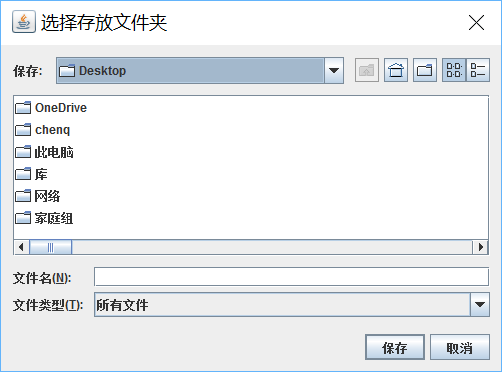
新建了一个MenuItemListener类，实现ActionListener接口，用来监控菜单项的点击：

监控“新建”菜单项点击的代码：

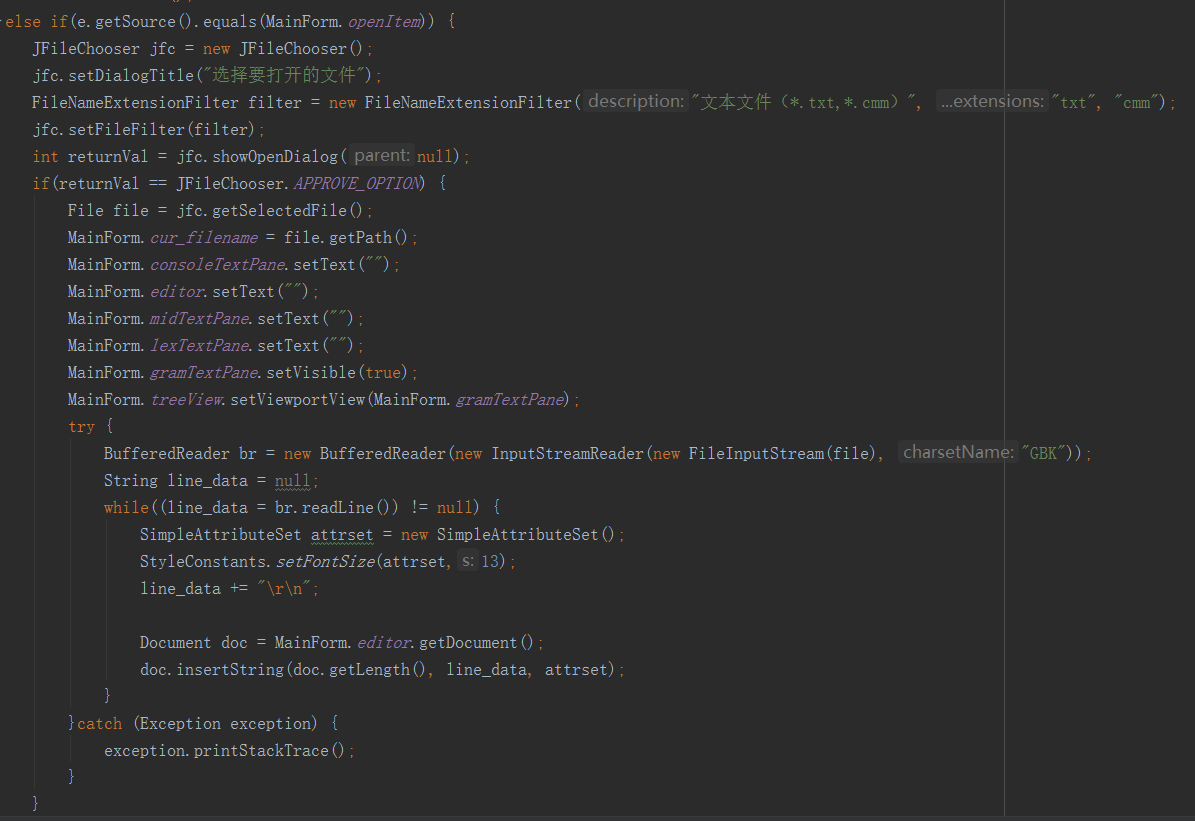


使用JfileChooser控件来实现此功能，当新建成功后初始化MainForm里的内容。

点击“新建”效果如下：

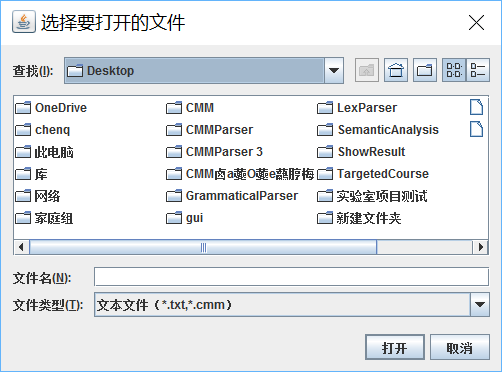


监控“打开”菜单项点击的代码：

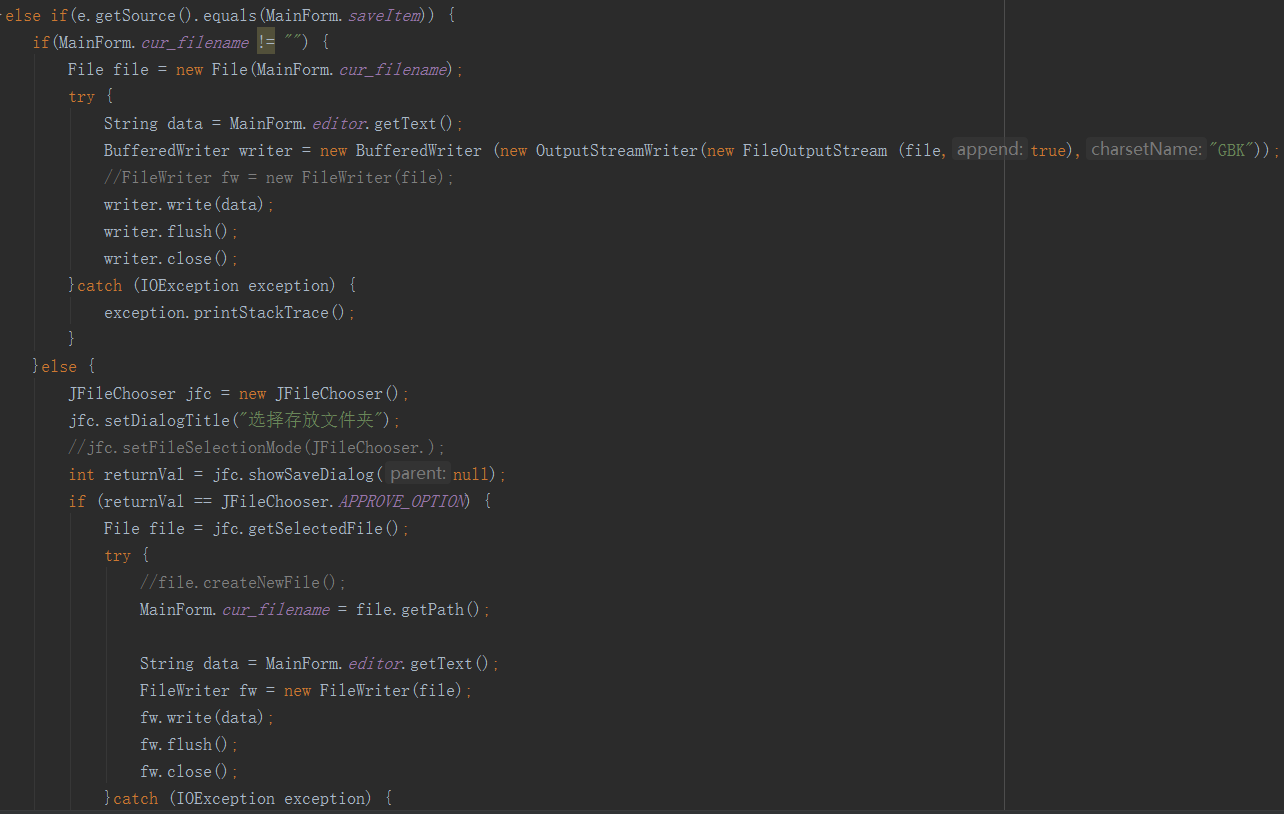


使用JfileChooser控件来实现此功能，当打开成功后初始化MainForm里的内容，设置被打开的文本的编码为GBK，防止中文乱码。

点击“打开”效果如下：

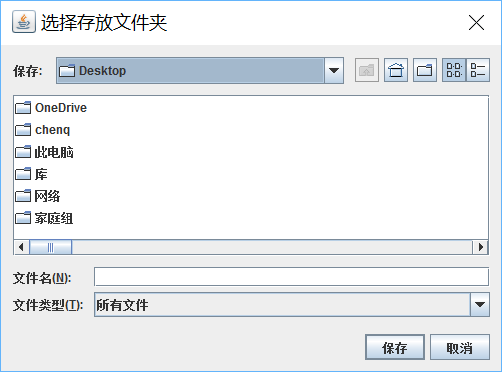


监控“保存”菜单项点击的代码：



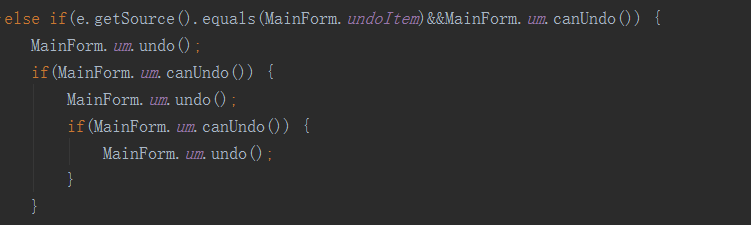
使用JfileChooser控件来实现此功能，当前已有打开的文件时，将编辑面板中全部内容写到原文件中，当前没有打开的文件时，新建一个文件，并存储编辑面板里的内容。保存时设置将要存储的内容编码为GBK，防止中文乱码。

当前无打开的文件时，点击“保存”效果如下：



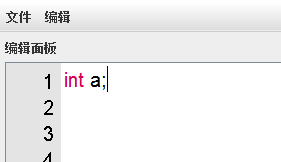
当前有打开的文件，点击“保存”直接保存。

监控“撤销”菜单项点击的代码：

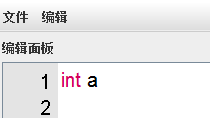


由于实现了着色功能，仅使用一次undo无法退回到输入字符之前的状态，只能退回到颜色、字体替换之前状态，所以需要三次undo。

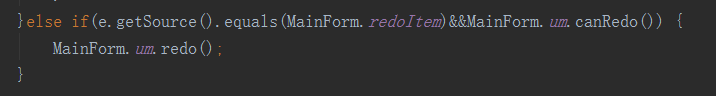
实现效果如下：



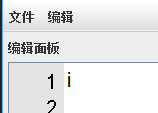
点击撤销后：



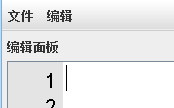
监控“恢复”菜单项点击的代码：



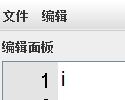
效果如下：



点击撤销：



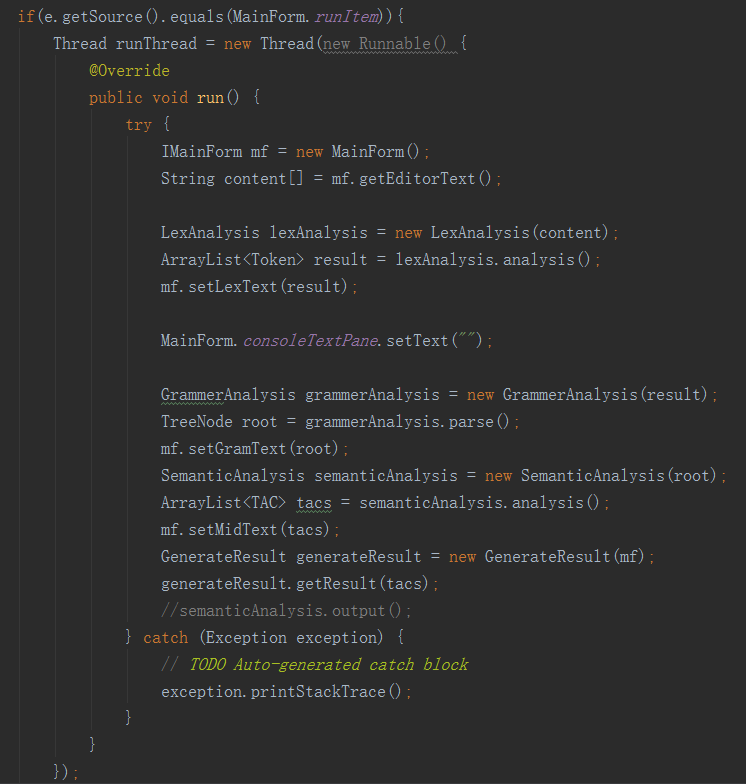
点击恢复：



1. **运行功能**

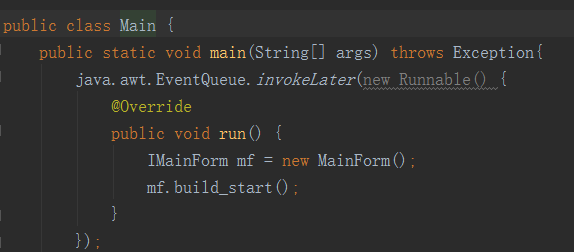
在MenuItemListener类中添加监控“运行”菜单项的功能：

点击“运行”菜单项时代码如下：

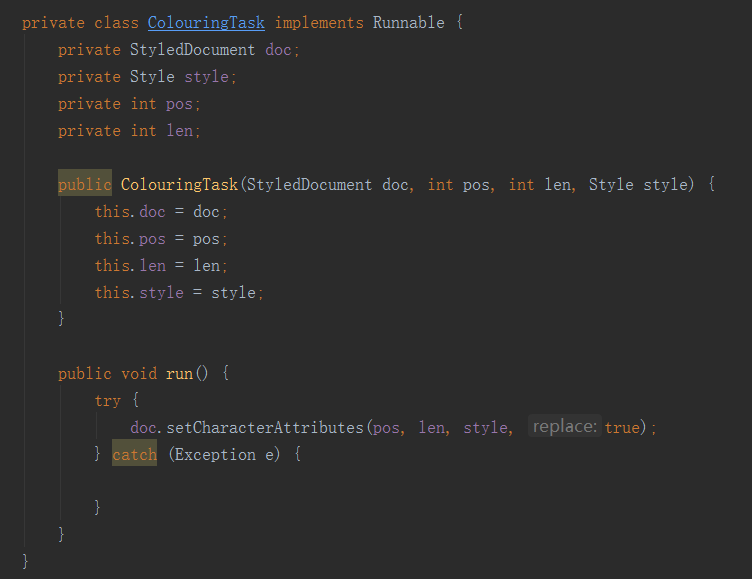


如图所示，新建了一个线程用来调用后台进行代码分析。

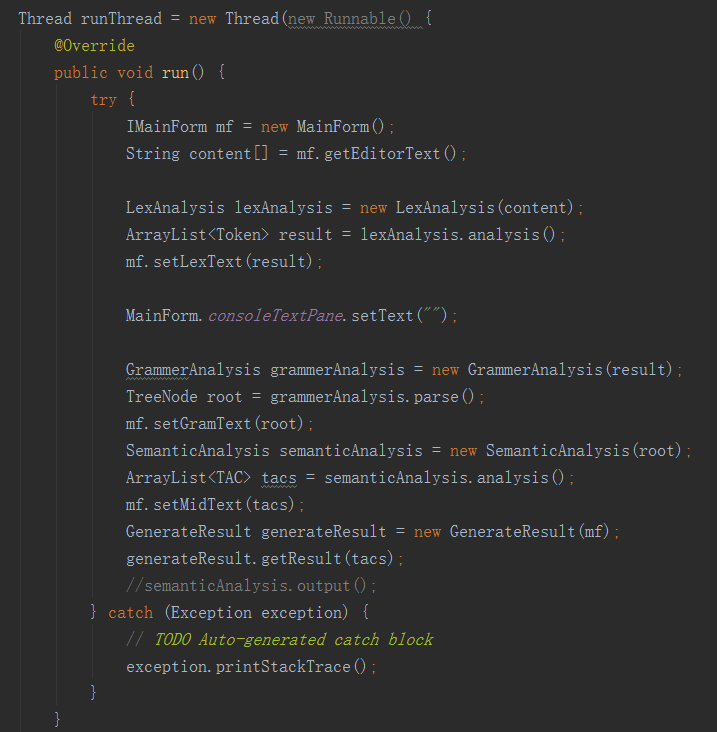
1. **使用多线程分离复杂任务**
2. 在整个项目的Main类中，初始新建一个线程用来启动前端GUI界面，代码如下：



1. 关键词着色时，判断需要着色后，新建一个线程执行关键的着色功能，代码如下：

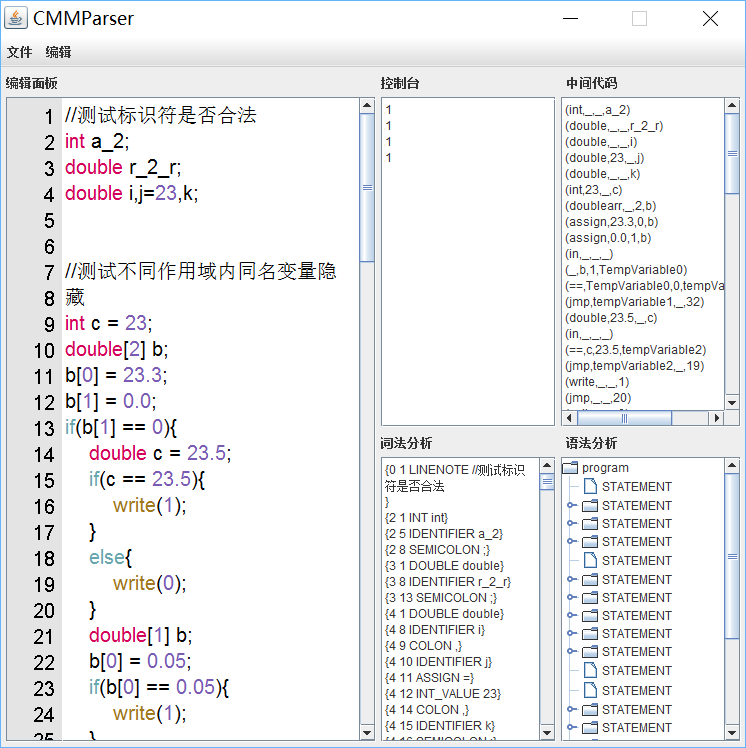


1. 点击“运行”菜单项后，新建一个线程进行代码分析，代码如下：

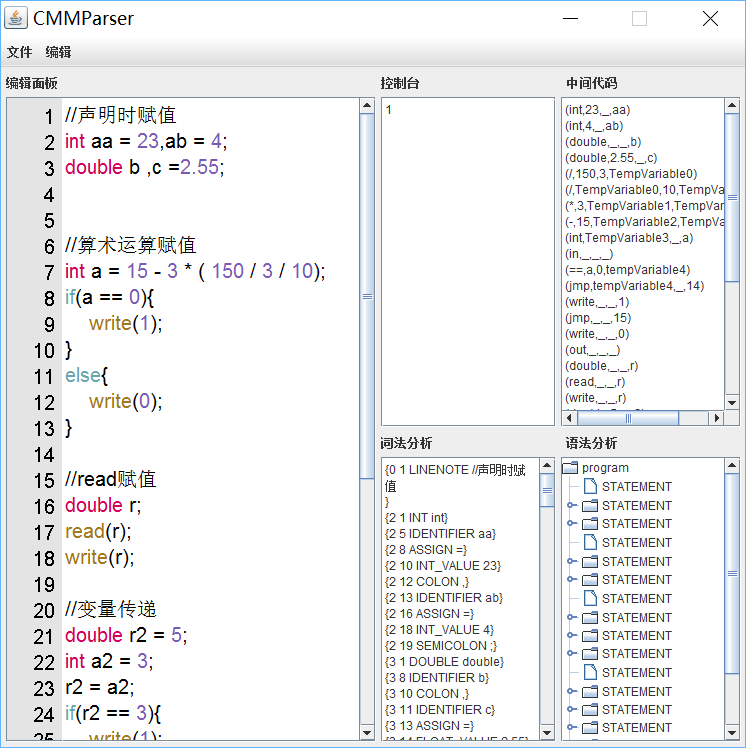


**三、测试结果：**

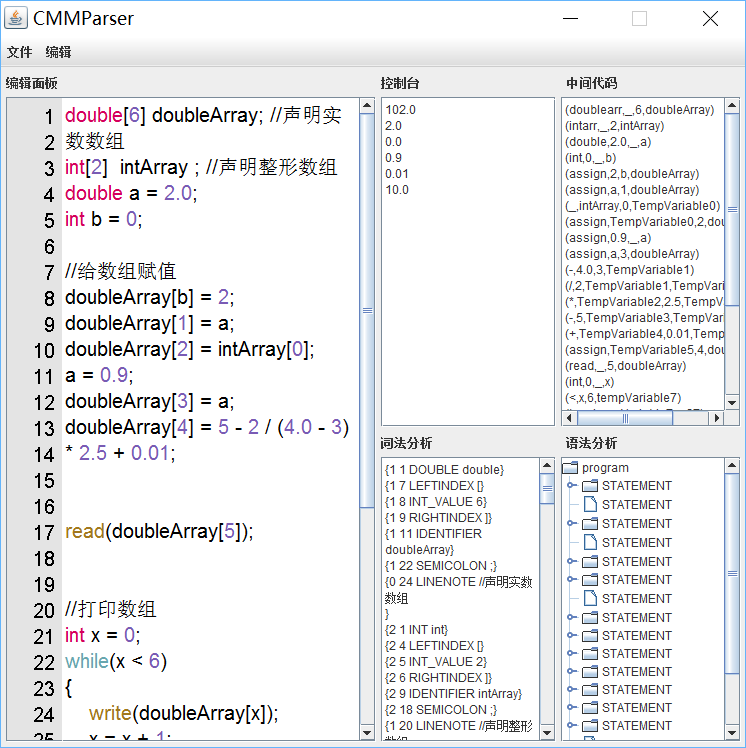
test\_case1：



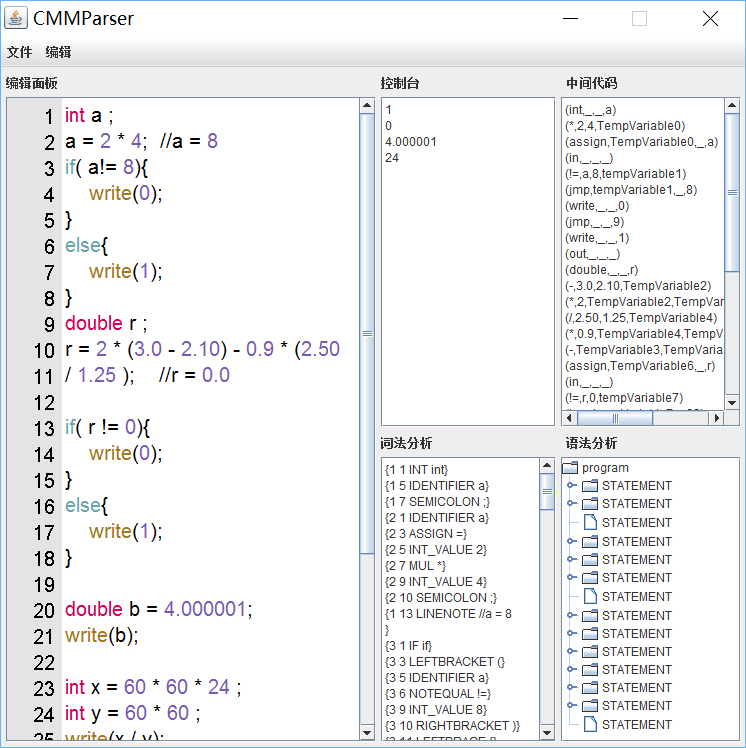
test\_case2：



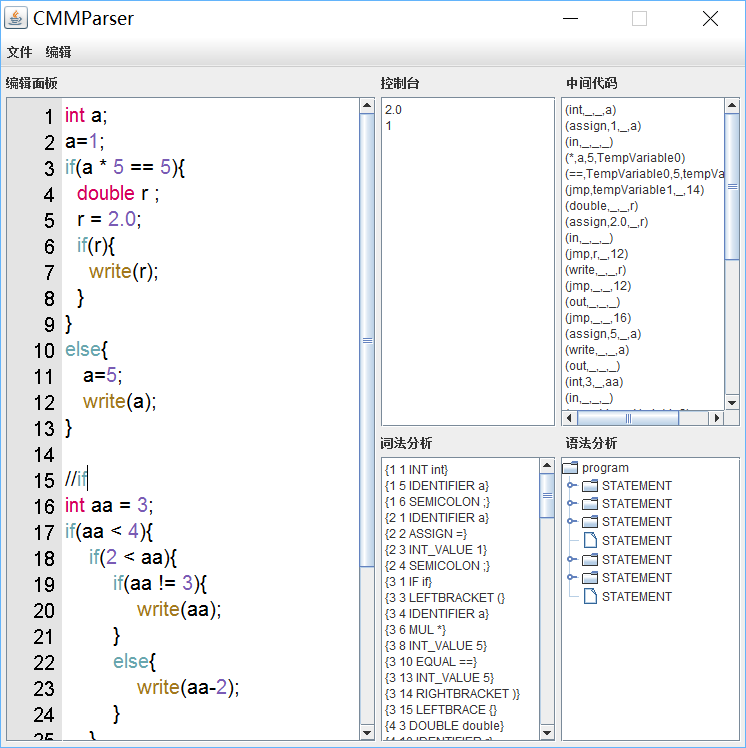
test\_case3：



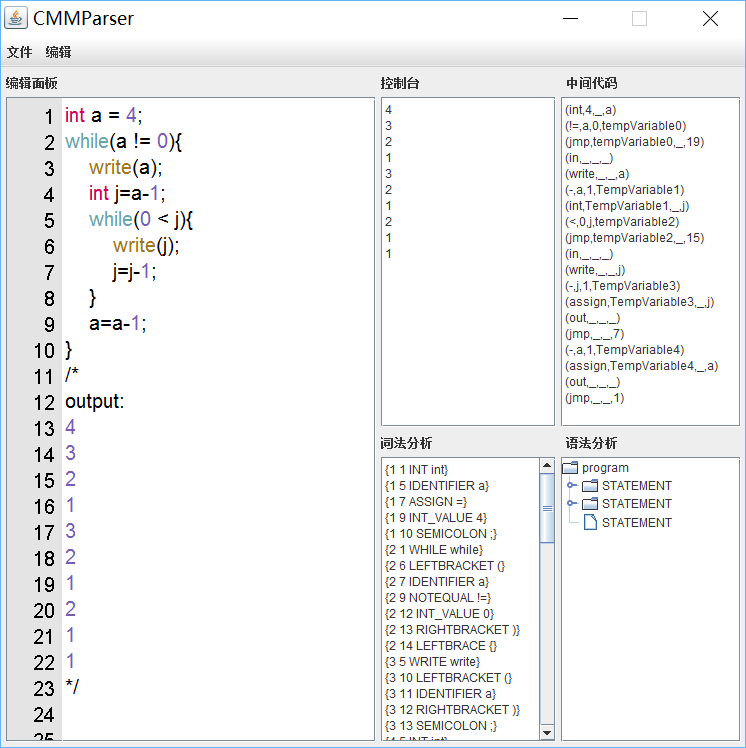
test\_case4：



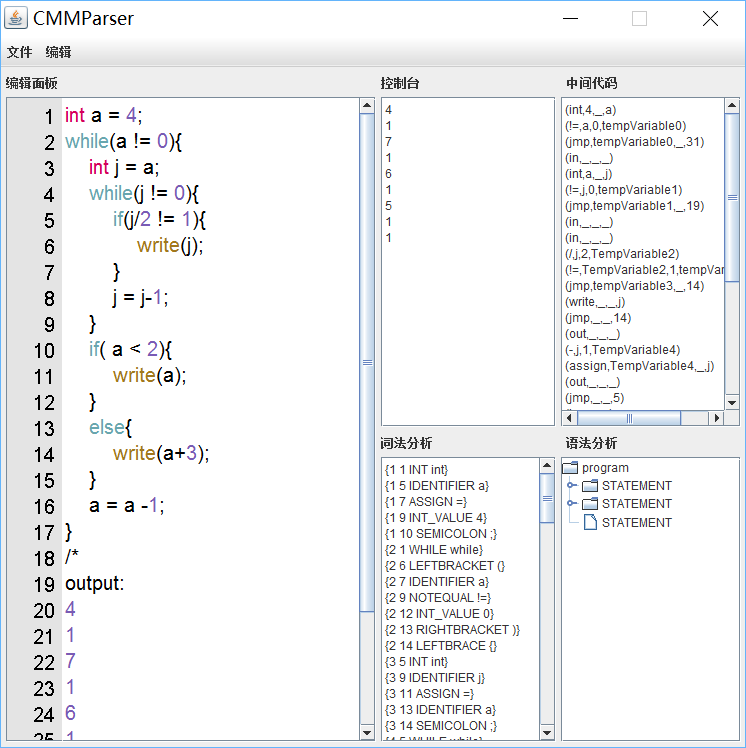
test\_case5：



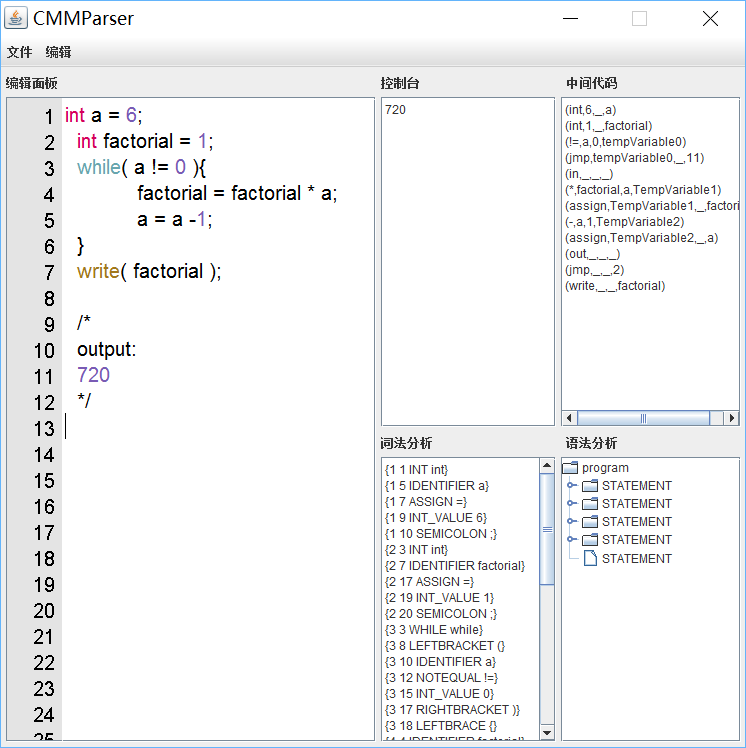
test\_case6：



test\_case7：



test\_case8：



test\_case9：

