编译原理实验

> ECNU 后端及编译器具体实现 – 第六小组

实现语言

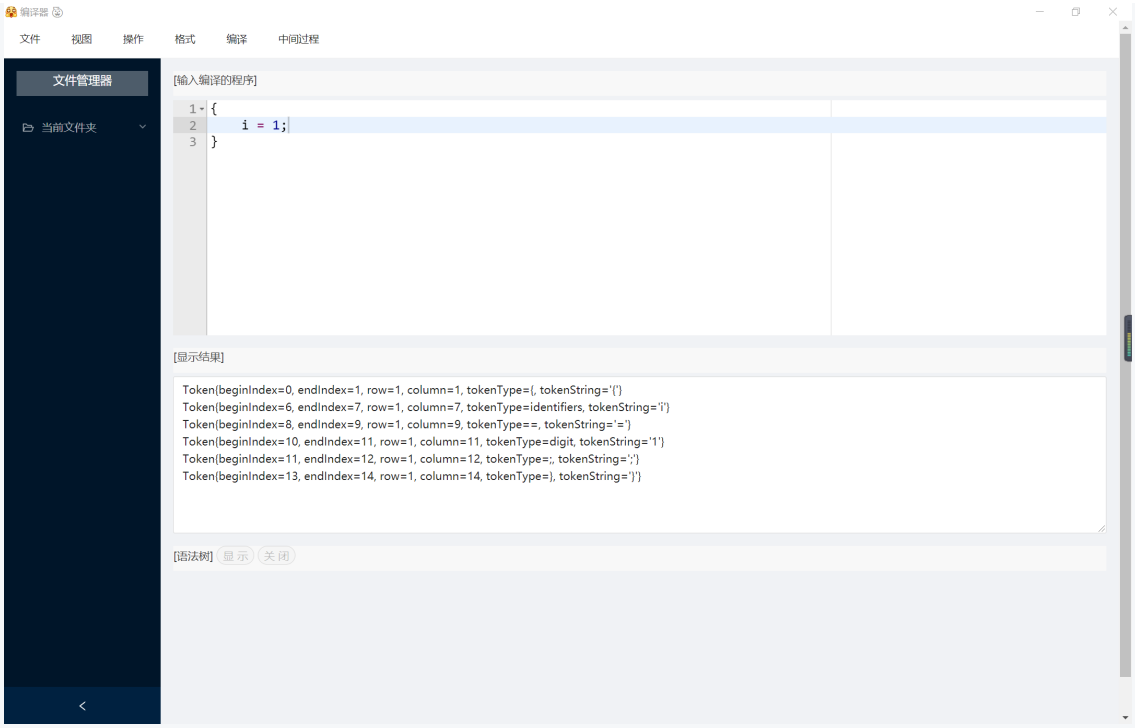
> JAVA

已完成

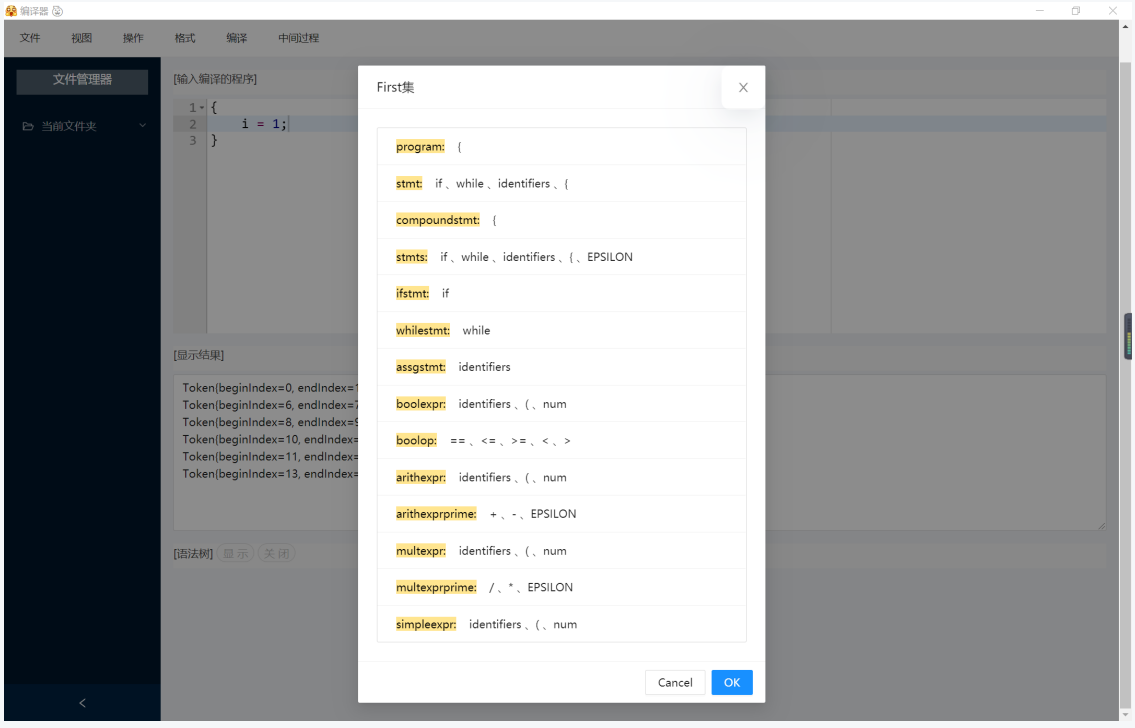
> - 词法分析 + 分词符号表
> - LL(1) 语法分析 + 恐慌模式错误恢复
> - 四元式中间代码 + 符号表生成
> - 语义分析解释器

效果图

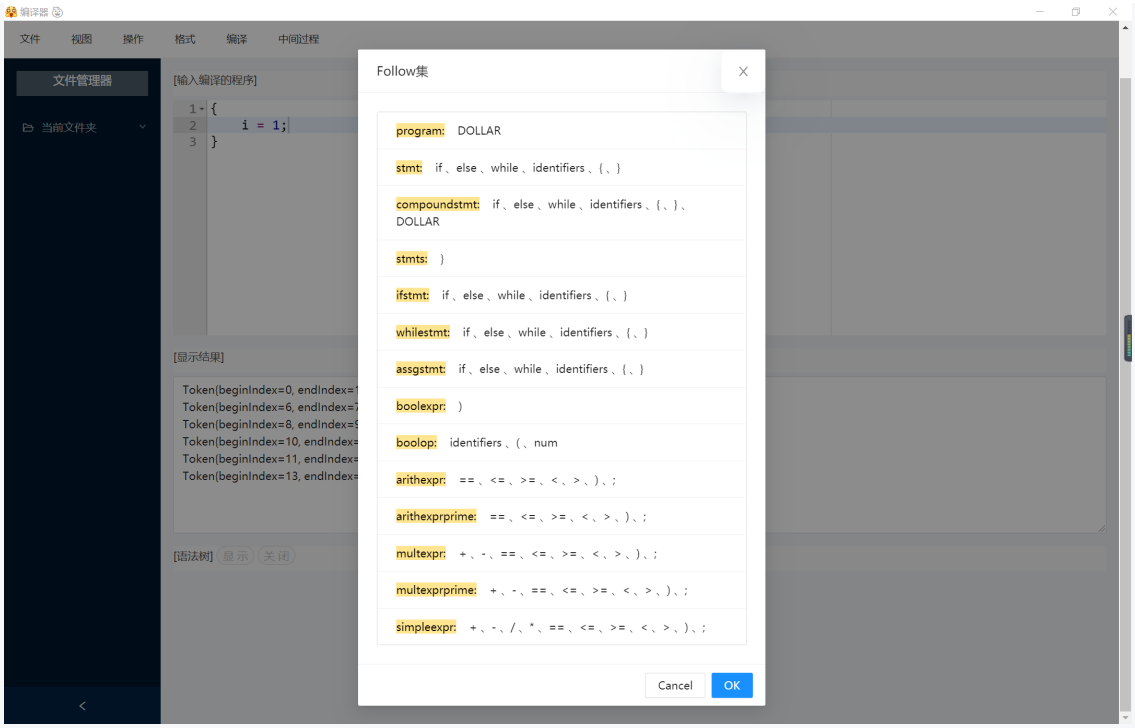
> **a) 基本的输入、输出界面；**
>
> **1. 输入**
>
> [输入编译的程序]
>
> ```
> 1 {
> 2     i = 1 ;
> 3 }
> ```
>
> **2. 词法分析结果**



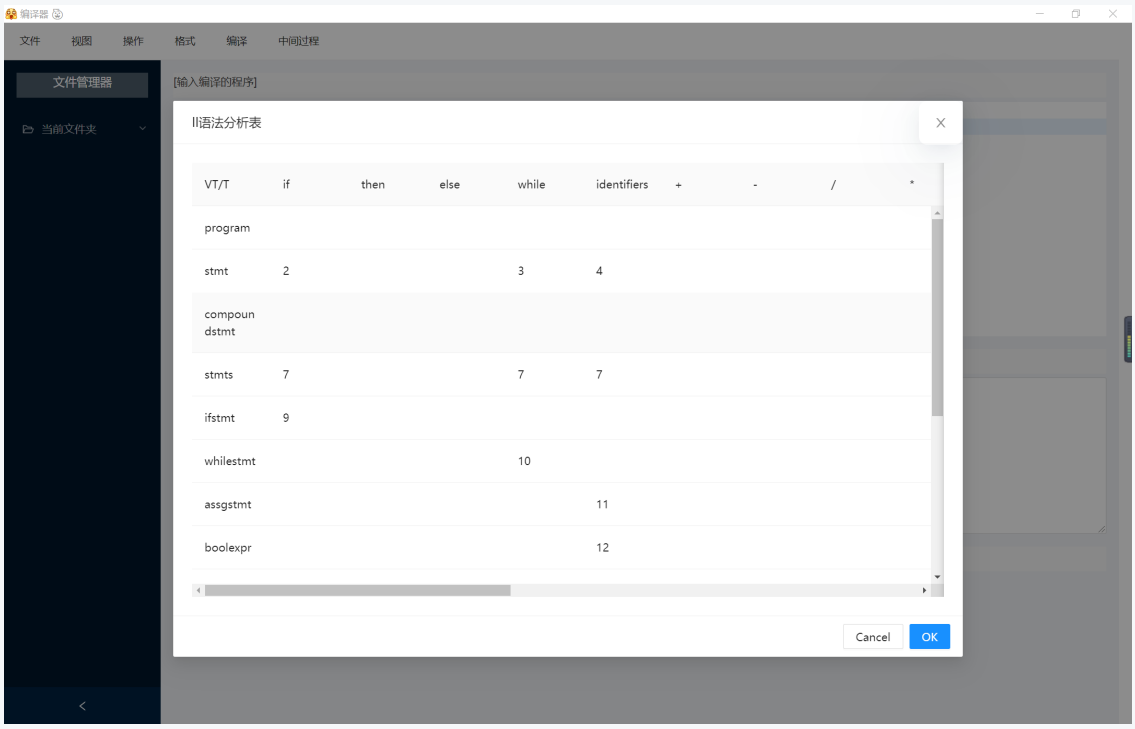
3. LL(1) FIRST 集



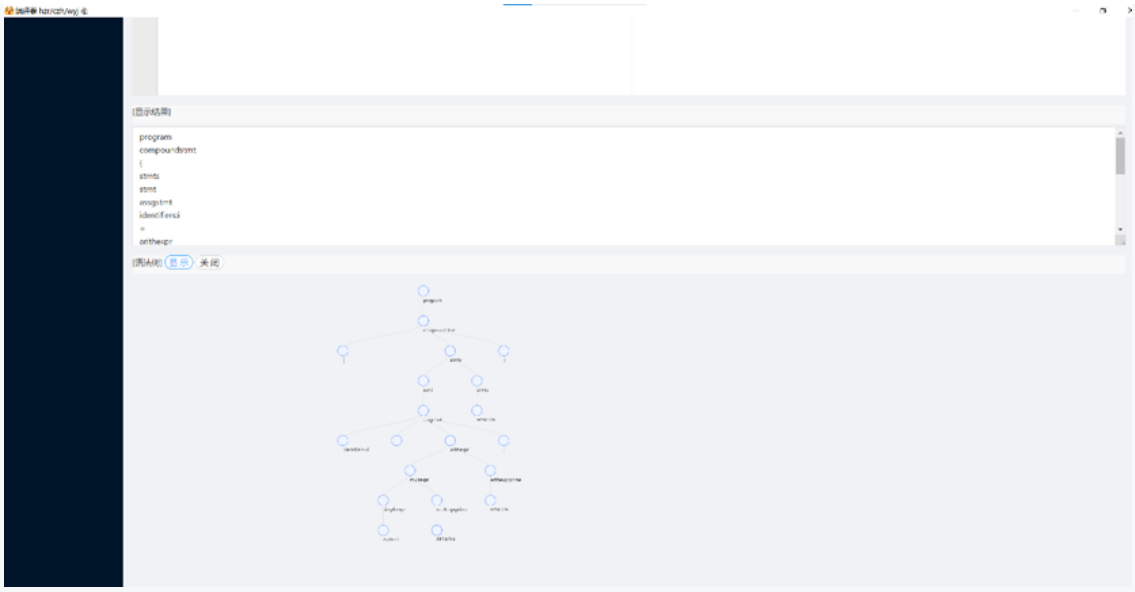
4. LL(1)Follow 集

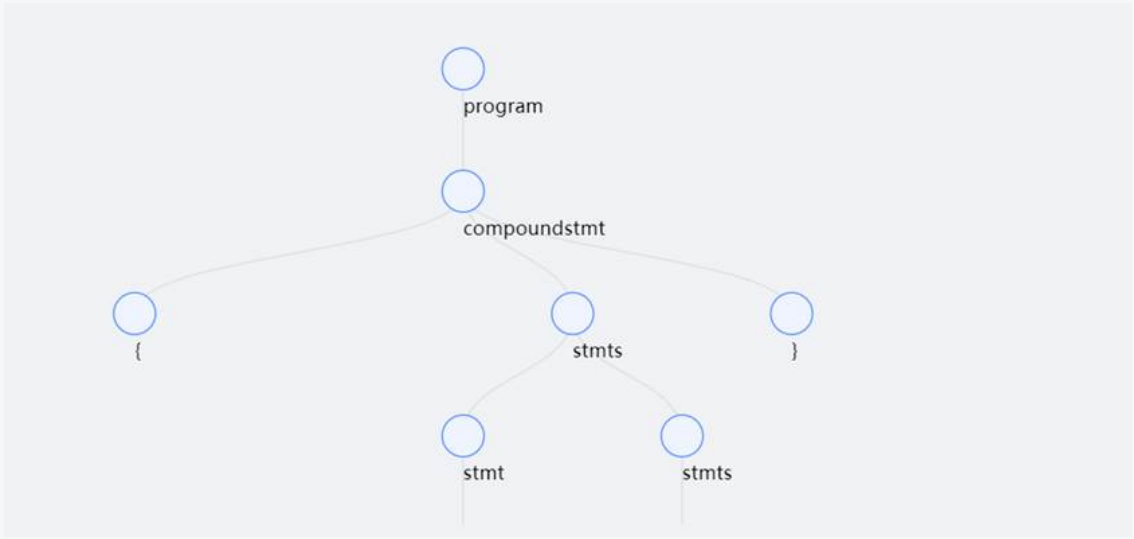


5. LL (1)解析表

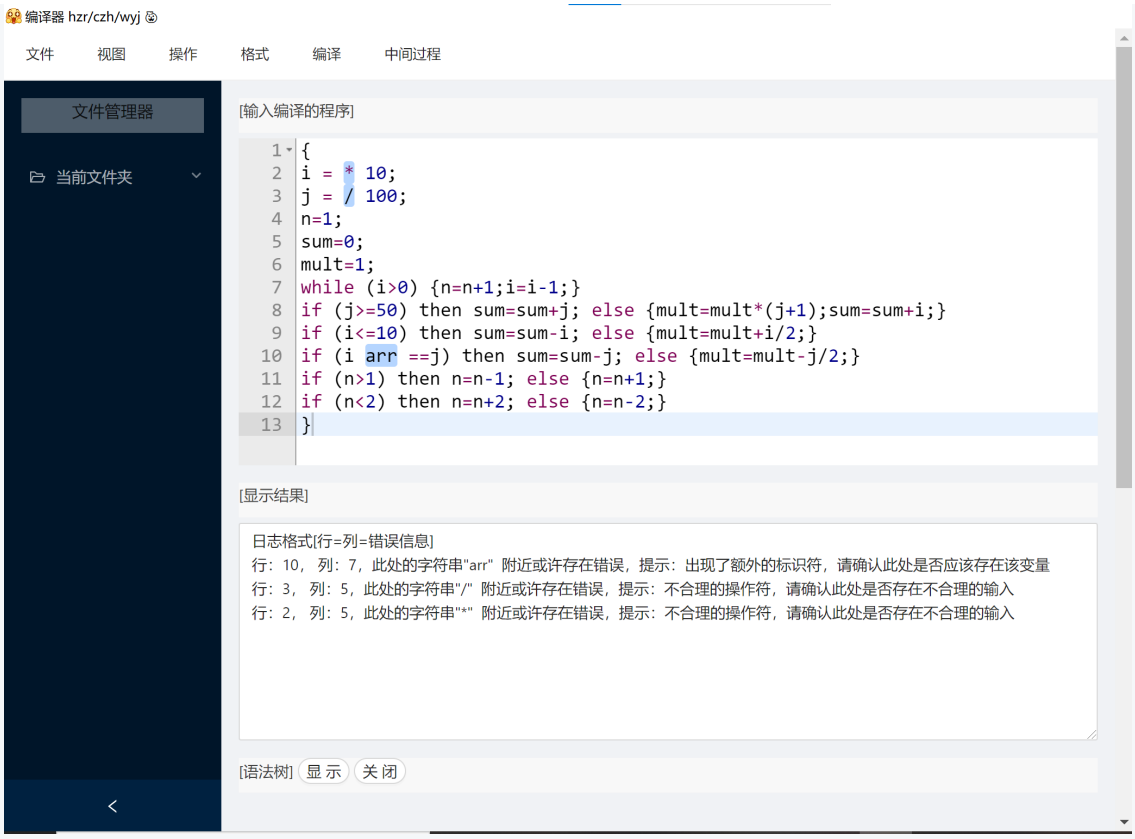


6. LL(1) 生成的语法树

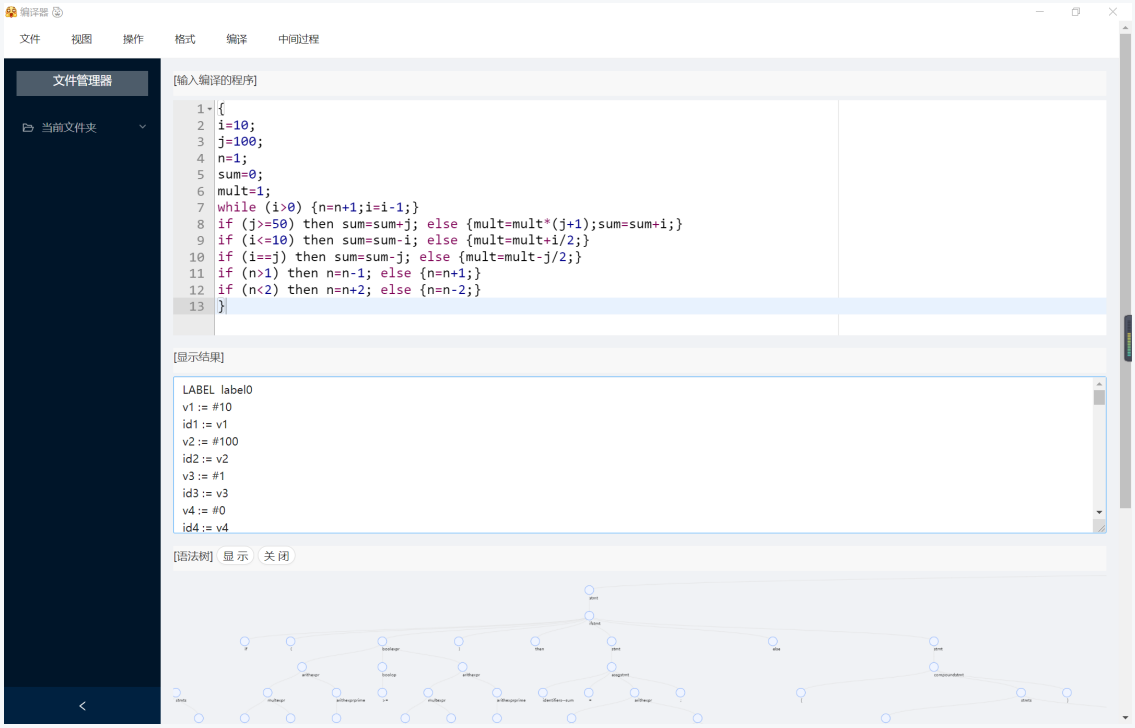




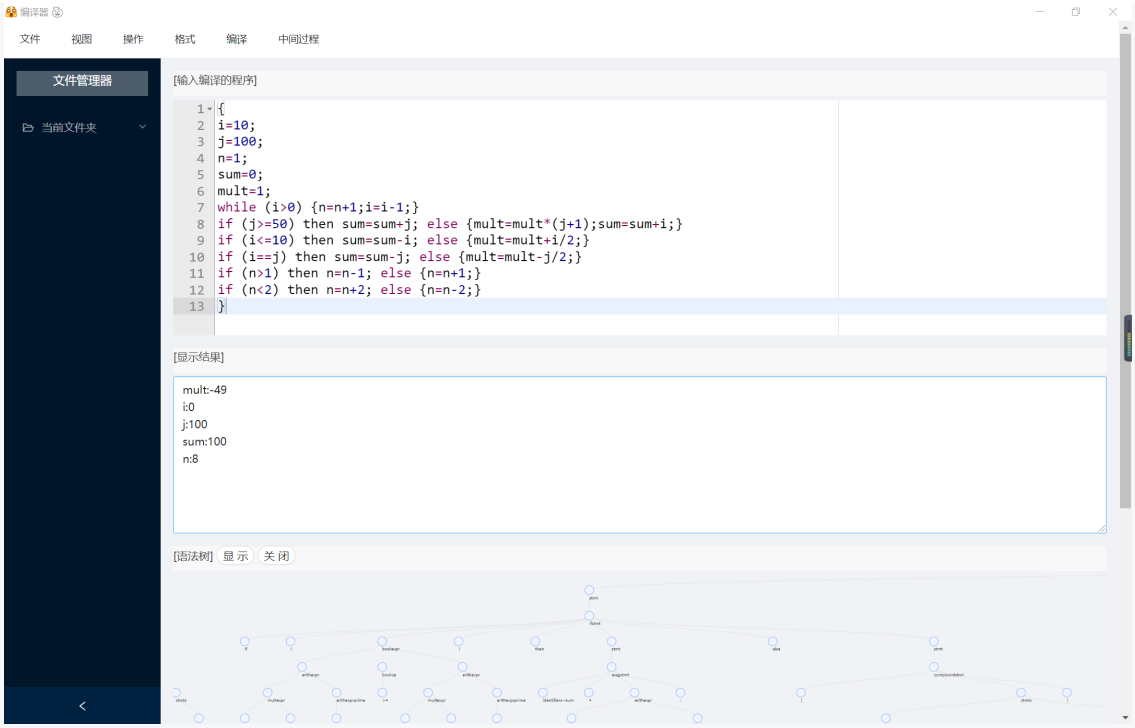
7. 错误提示（错误高亮 和 错误日志显示）



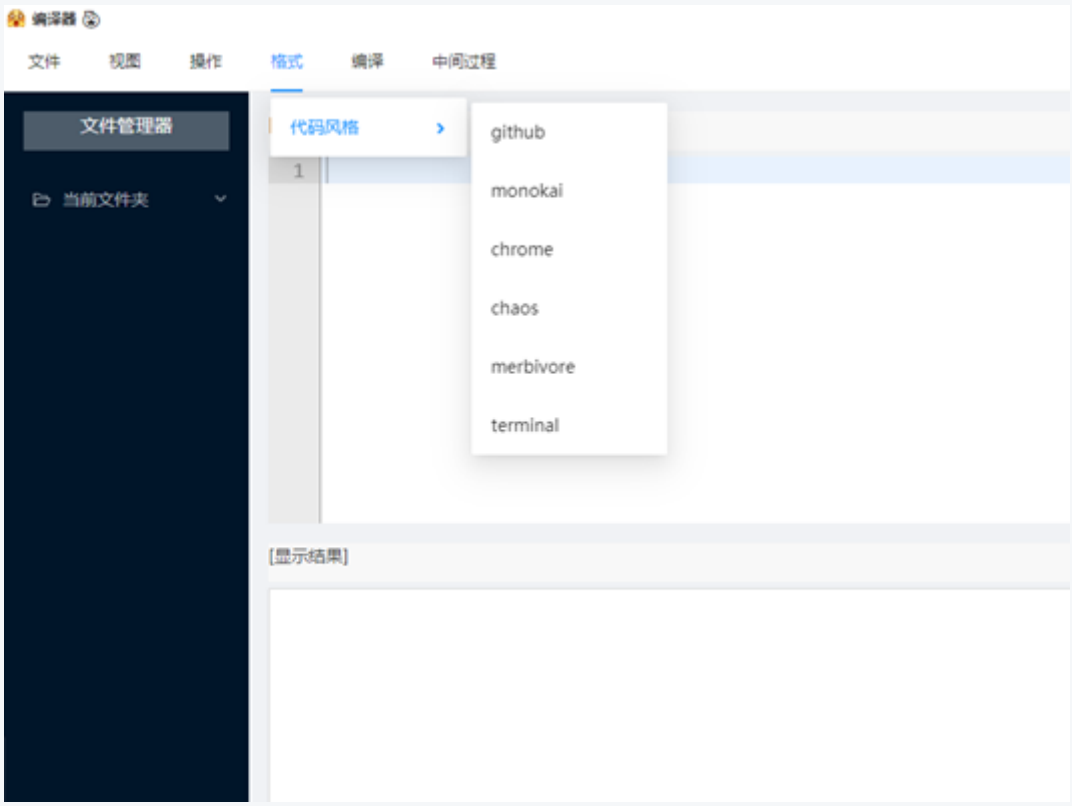
8. 中间代码生成



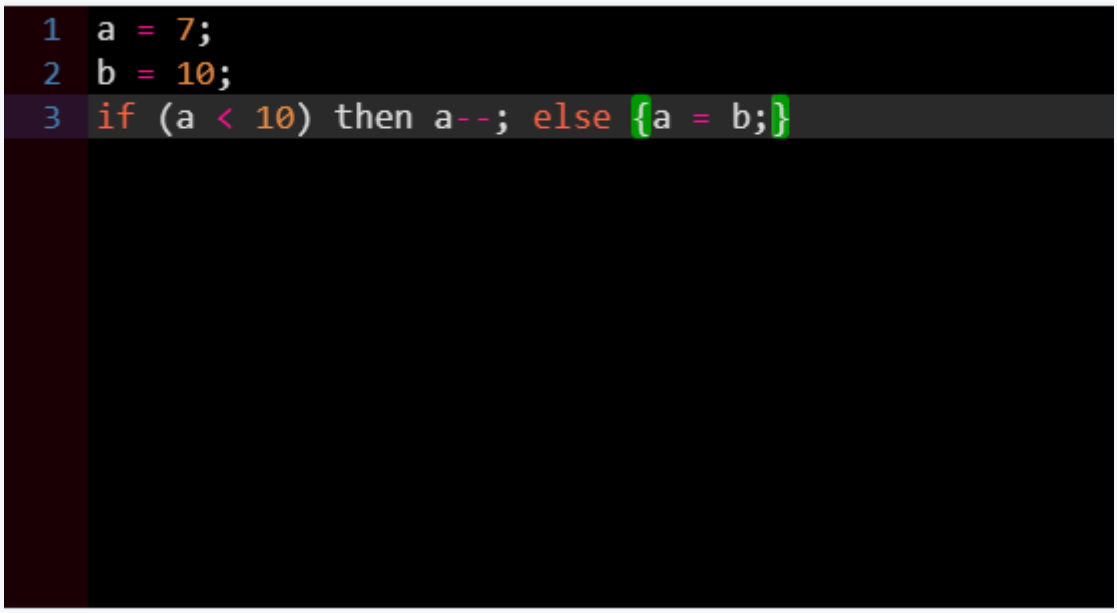
9. 语义分析解释结果

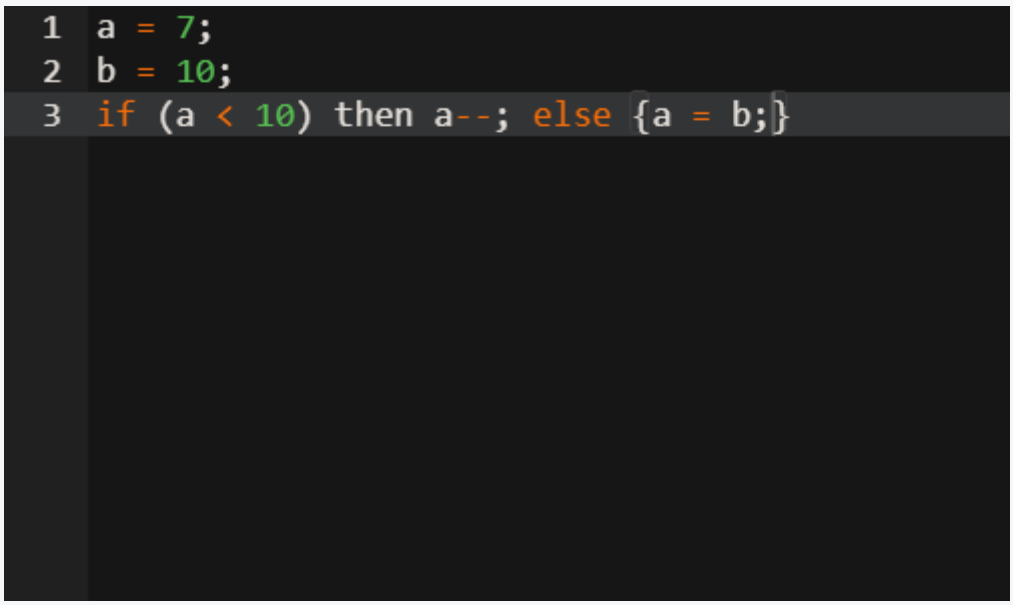


10. 自定义代码风格切换



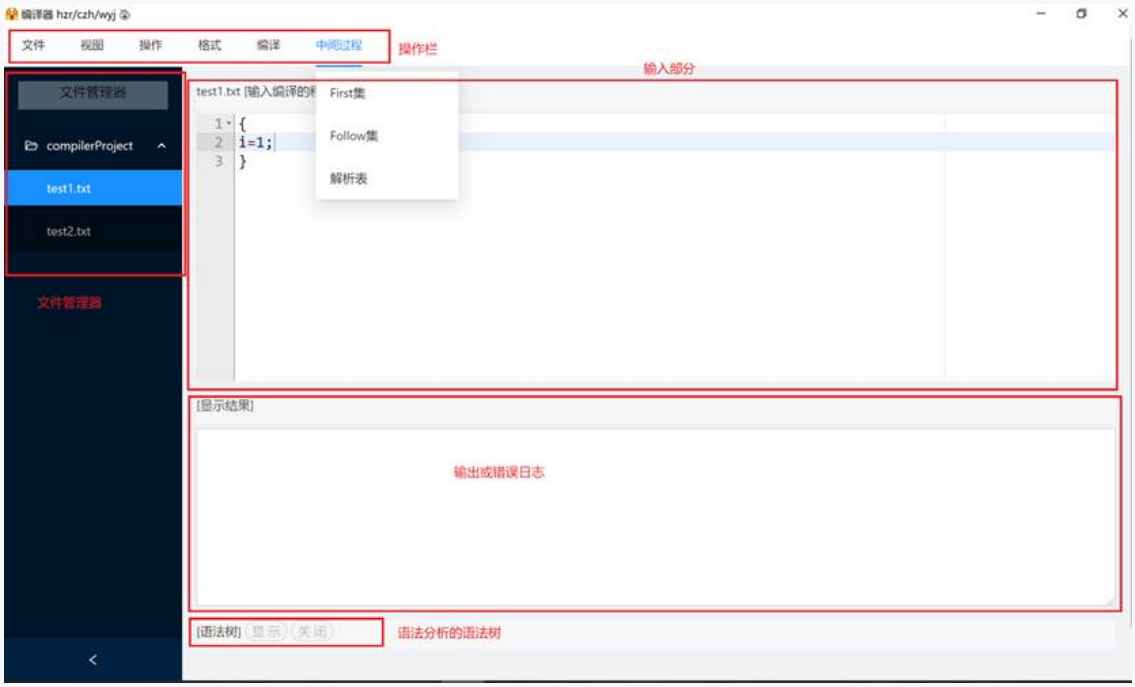
```
a = 7;
b = 10;
if (a < 10) then a--; else {a = b;}
```





```
a = 7;
b = 10;
if (a < 10) then a--; else {a = b;}
```

b) 界面布局与设计



运行说明

1. git clone https://github.com/Wind-Gone/Toy-compiler-GUI 克隆本项目的前端库
2. yarn 安装相关依赖包
3. yarn start 启动 Electron
4. git clone https://github.com/Wind-Gone/Toy-compiler 克隆项目后端编译器
5. 启动 Springboot 即可运行

项目目录

```
├─src
│  ├─main
│  │  ├─java
│  │  │  └─com
│  │  │      └─example
│  │  │          └─compiler
│  │  │              │  CompilerApplication.java          // Springboot启动入口
│  │  │              │
│  │  │              ├─controller                        // 用于处理前后端请求的Controller层目录
│  │  │              │      controller.java               // 具体的执行文件Controller.java
│  │  │              │
│  │  │              ├─entity                            // 实体目录
│  │  │              │  ├─gui                            // 用于与前端交互的特殊类目录
│  │  │              │  │      GuiNode.java              // 与树形组件配套的语法树类
│  │  │              │  │      Text.java                 // 前端传递的数据的实体类
│  │  │              │  │
│  │  │              │  ├─token                          // Token相关目录
│  │  │              │  │      Token.java                // 词法分析解析的Token类
│  │  │              │  │      TokenTable.java           // 词法分析表的主体类
│  │  │              │  │      TokenType.java            // Token类型的枚举类
│  │  │              │  │
│  │  │              │  ├─tree                           // 语法树相关目录
│  │  │              │  │      SyntaxTree.java           // 语法树的实体类
│  │  │              │  │      TreeNode.java             // 语法树节点的实体类
│  │  │              │  │
│  │  │              │  └─wrong                          // 错误信息相关目录
│  │  │              │          ErrorCode.java           // 错误代码和代码指示的信息的枚举类
│  │  │              │          WrongMessage.java        // 定义详细错误信息的实体类
│  │  │              │
│  │  │              ├─intermediateCodeGeneration        // 中间代码生成
│  │  │              │      Dag.java
│  │  │              │      Leaf.java
│  │  │              │      Node.java
│  │  │              │
│  │  │              ├─lexer                               // 词法分析器目录
│  │  │              │      Lexer.java                     // 词法分析器(ex1)
│  │  │              │      Main.java                      // 词法分析器主体，实现了对token判别类型
│  │  │              │
│  │  │              ├─llParser                            // 语法分析目录
│  │  │              │      Grammer.java                   // 存储所有语法规则的产生式类
│  │  │              │      LLParser.java                  // 语法分析器(ex2)
│  │  │              │      LLUtil.java                    // 语法分析器处理first/follow集合等函数的工具类
│  │  │              │      NonTerminalType.java           // 非终结符枚举类
│  │  │              │      ParsingTable.java              // 语法解析表实体类
│  │  │              │      Production.java                // 语法产生式实体类
│  │  │              │
│  │  │              ├─semantic                            // 语义分析目录
│  │  │              │      Main.java                      // 语义分析器(ex4)，调用中间代码生成解释器
│  │  │              │      SemanticAnalyzer.java          // 语义分析其具体执行类
│  │  │              │      SymbolTable.java               // 语义分析符号表类
│  │  │              │      ThreeCodeTable.java            // 三地址码实体类
│  │  │              │
│  │  │              └─utils                               // 项目所需要使用的工具类目录
│  │  │                      FileUtils.java                // 文件工具类，负责文件IO
│  │  │
│  │  └─resources
│  │          application.properties
│  │
│  └─test
│      └─java
│          └─com
│              └─example
│                  └─compiler
│                          CompilerApplicationTests.java     // 常规测试
```

```
TestForGrammer.java          // 语法分析测试
TestForLexer.java            // 词法分析测试
TestForSemantic.java         // 语义分析测试
```

特别鸣谢

@caizhenghai

@Ling-WYJ

@Wind-Gone