CMM词法分析器

1. 背景

完成人:袁浩

学号:2013302580115

班级:国际软件学院软件工程3班

提交日期:2015-09-30

完成CMM词法分析器

1. 功能实现描述
2. 综述

主要实现词法分析的类是:

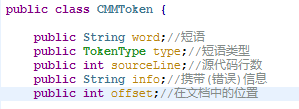
com.yuanhonglong.analysis.lexical\_analysis.LexicalAnalysis

主要的函数是startAnalysis,此函数模拟的是有穷自动机(不完全是有穷自动机),还有numbers(),symbols(),keywordAndIdentifier()等函数,分别用于识别数字,符号,关键字和标识符,至于注释和空白符则在startAnalysis()函数中进行分析.

1. 辅助函数

isKeyword()判断是否关键字,isSymbol()判断字符是否是特殊符号,isLetter()判断字符是否是字母,isNumber()判断字符是否是数字

1. 数据结构

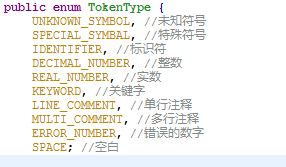


CMMToken存储的是分词结果,其中还携带了代码行数和短语在文档中的位置等信息,是给调试做准备的.

1. 特殊符号和关键字表

**public static final char** SKIP\_SPACE = ' ';  
**public static final char** SKIP\_TAB = '**\t**';  
**public static final char** SKIP\_RETURN = '**\r**';  
**public static final char** SKIP\_NEW\_LINE = '**\n**';  
**public static final char** UNDER\_LINE = '\_';  
//public static final String SKIP\_LINE\_COMMENT = "//";  
//public static final String SKIP\_COMMENT\_START = "/\*";  
**public static final String** SKIP\_COMMENT\_END = "\*/";  
//public static final String DECIMAL\_NUMBER =;  
//public static final String REAL\_NUMBER =;  
**public static final String** TRUE\_KEYWORD = "true";  
**public static final String** FALSE\_KEYWORD = "false";  
**public static final String** REAL\_KEYWORD = "real";  
**public static final String** INT\_KEYWORD = "int";  
**public static final String** BOOLEAN\_KEYWORD = "boolean";  
**public static final String** VOID\_KEYWORD = "void";  
**public static final String** IF\_KEYWORD = "if";  
**public static final String** ELSE\_KEYWORD = "else";  
**public static final String** READ\_KEYWORD = "read";  
**public static final String** WRITE\_KEYWORD = "write";  
**public static final String** WHILE\_KEYWORD = "while";  
**public static final String** FOR\_KEYWORD = "for";  
**public static final String** BREAK\_KEYWORD = "break";  
**public static final String** CONTINUE\_KEYWORD = "continue";  
**public static final String** RETURN\_KEYWORD = "return";  
//public static final String IDENTIFIER = ;  
**public static final char** LEFT\_SMALL\_BRACKET = '(';  
**public static final char** RIGHT\_SMALL\_BRACKET = ')';  
**public static final char** LEFT\_LARGE\_BRACKET = '{';  
**public static final char** RIGHT\_LARGE\_BRACKET = '}';  
**public static final char** LEFT\_MIDDLE\_BRACKET = '[';  
**public static final char** RIGHT\_MIDDLE\_BRACKET = ']';  
**public static final char** SEMICOLON = ';';  
**public static final char** COMMA = ',';  
**public static final char** DOT = '.';  
**public static final char** ASSIGN = '=';  
**public static final char** GREATER\_THAN = '>';  
**public static final char** LESS\_THAN = '<';  
**public static final char** BANG = '!';  
**public static final char** TILDE = '~';  
**public static final String** DOUBEL\_EQUAL = "==";  
**public static final String** LESS\_EQUAL = "<=";  
**public static final String** GREATER\_EQUAL = ">=";  
**public static final String** NOT\_EQUAL = "!=";  
**public static final String** DOUBLE\_OR = "||";  
**public static final String** DOUBLE\_AND = "&&";  
**public static final String** INCREMENT = "++";  
**public static final String** DECREMENT = "--";  
**public static final char** PLUS = '+';  
**public static final char** MINUS = '-';  
**public static final char** STAR = '\*';  
**public static final char** SLASH = '/';  
**public static final char** AND = '&';  
**public static final char** OR = '|';  
**public static final char** XOR = '^';  
**public static final char** PERCENT = '%';  
**public static final String** LEFT\_SHIFT = "<<";  
**public static final String** RIGHT\_SHIFT = ">>";  
**public static final String** PLUS\_ASSIGN = "+=";  
**public static final String** MINUS\_ASSIGN = "-=";  
**public static final String** STAR\_ASSIGN = "\*=";  
**public static final String** SLASH\_ASSIGN = "/=";  
**public static final String** AND\_ASSIGN = "&=";  
**public static final String** OR\_ASSIGN = "|=";  
**public static final String** XOR\_ASSIGN = "^=";  
**public static final String** PERCENT\_ASSIGN = "%=";  
**public static final String** LEFT\_SHIFT\_ASSIGN = "<<=";  
**public static final String** RIGHT\_SHIFT\_ASSIGN = ">>=";

1. 种类码表



定义在TokenType类中

1. 使用说明

本程序提供的是可执行的.jar包,需要使用JDK7及以上版本的Java虚拟机才能运行.

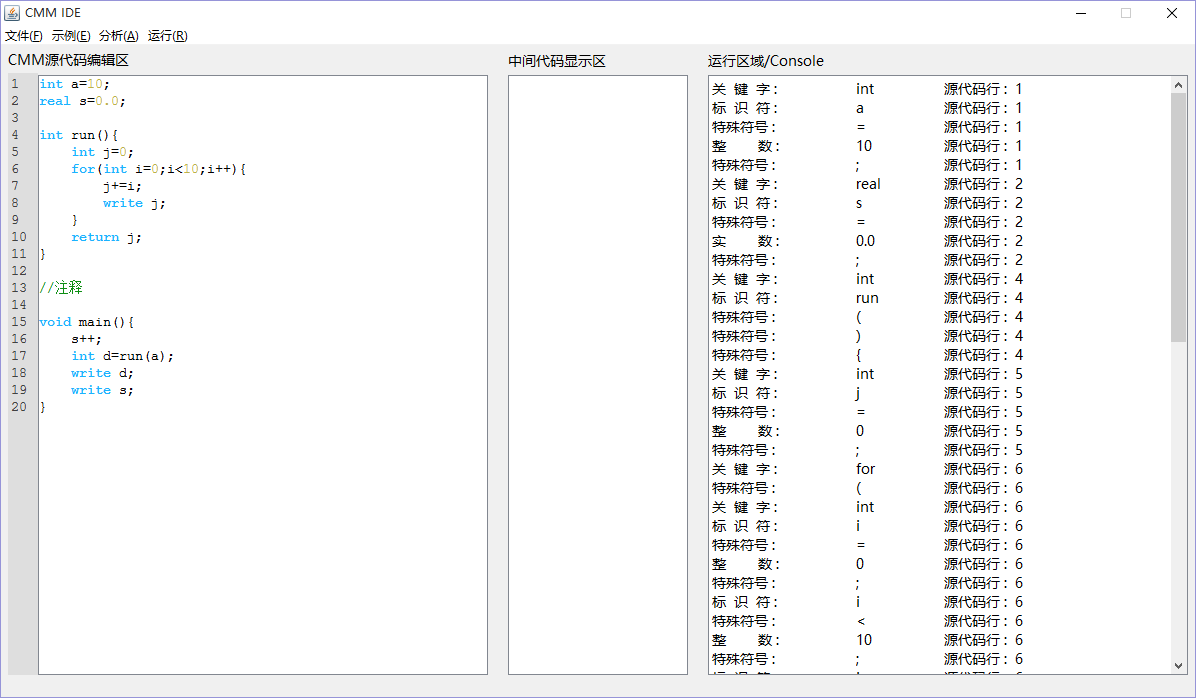
1. 参考资料

<<编译原理及实践>>

1. 运行说明

词法分析快捷键Ctrl+L,或者使用菜单(分析->词法分析)

本程序不但可以在界面中显示分词结果,还可以根据分词结果对源代码进行高亮处理.



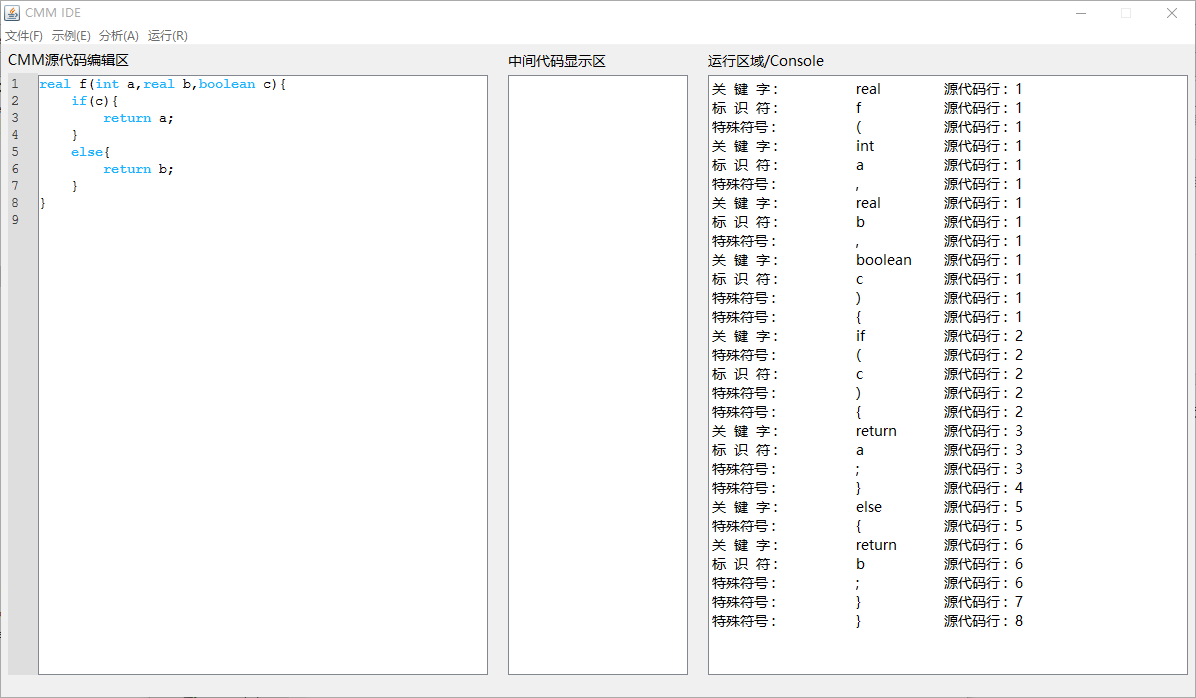
中间代码显示区是给语法分析和运行调试预留的,现在还没有内容

1. 测试数据

在tests文件夹下

正确的和错误的CMM文件都提供了四个,以下是部分截图:

正确:



错误:

