解释器第二次实验报告

1.背景

姓名：余璞轩
学号：2014302580341
班级：卓越二班
提交日期：10月26日

任务：CMM 语言总体架构的设计和搭建；CMM 语言词法分析

2.方法

(1).写好antlr文件

```
grammar cmm;

program :
    (stmt)+
    ;

stmt :
    var_decl                #vardecl
    | if_stmt               #ifstmt
    | while_stmt            #whilestmt
    | break_stmt            #breakstmt
    | assign_stmt           #assignstmt
    | read_stmt             #readstmt
    | write_stmt            #writestmt
    | stmt_block            #stmtblock
    | LParen stmt RParen    #LParenStmtRParen
    ;

stmt_block :
    lcurly (stmt)* rcurly
    ;

var_decl :
    type var_list lineend
    ;

type :
    int_ |
    double_ |
    type lbracket intconstant rbracket
    ;

var_list :
    ident (comma ident)*
    ;

decl_assign :
    ident assign expr
    ;

if_stmt :
    if_ expr stmt |
    if_ expr stmt else_ stmt
    ;

while_stmt :
    while_ expr stmt
    ;

break_stmt :
    break_ lineend
    ;

read_stmt :
    read_ lparen ident rparen lineend |
    read_ lparen ident lbracket intconstant rbracket lineend
    ;

write_stmt :
    write_ lparen expr rparen lineend
    ;

assign_stmt :
    value assign expr lineend
    ;
```

```
value :
    ident |
    ident lbracket intconstant rbracket
    ;

constant :
    intconstant |
    realconstant |
    booleanconstant
    ;

expr :
    expr muldivmod expr |
    expr addmin expr |
    expr compop expr |
    lparen expr rparen |
    ident |
    constant |
    addmin expr
    ;

If : 'if' ;
Else : 'else' ;
While : 'while' ;
Read : 'read' ;
Write : 'write' ;
Int : 'int' ;
Double : 'double' ;
Break : 'break' ;

Ident :
    [a-zA-Z]([a-zA-Z] | '_' | [0-9])*
    ;

IntConstant : '0' | [1-9][0-9]*;
RealConstant : IntConstant('.'([0-9]+))? ;
BooleanConstant : 'true' | 'false' ;

CompOp : '<=' | '>=' | '>' | '<' | '!=' | '==' | '<>' |;
MulDivMod : '*' | '/' | '%';
AddMin : '+' | '-' ;

LParen : '(' ;
RParen : ')' ;
LCurly : '{' ;
RCurly : '}' ;
LBracket : '[' ;
RBracket : ']' ;

Assign : '=' ;

LineEnd : ';' ;

Comma : ',' ;

WS : [' '\t\r\n]+ -> skip ;

SL_COMMENT :   '//' ~[\r\n]* -> skip;
MUL_COMMENT :   '/*' .*? '*/' -> skip;
```

(2).在antlr文件中添加label

事实上只用给词法加label，但是antlr不支持给词法加label，所以考虑把词法当成语法来考虑，然后再加label。

```
if_ :
    If # getIf
    ;
else_ :
    Else # getElse
    ;
while_ :
    While # getWhile
    ;
read_ :
    Read # getRead
    ;
write_ :
    Write # getWrite
    ;
int_ :
    Int # getInt
    ;
double_ :
    Double # getDouble
    ;
break_ :
    Break # getBreak
    ;
ident :
    Ident # getIdent
    ;
intconstant :
    IntConstant # getIntConstant
    ;
realconstant :
    RealConstant # getRealConstant
    ;
booleanconstant :
    BooleanConstant # getBooleanConstant
    ;
compop :
    CompOp # getCompop
    ;
muldivmod :
    MulDivMod # getMulDivMod
    ;
addmin :
    AddMin # getAddMin
    ;
lparen :
    LParen # getLParen
    ;
rparen :
    RParen # getRParen
    ;
lcurly :
    LCurly # getLCurly
    ;
rcurly :
    RCurly # getRCurly
    ;
lbracket :
    LBracket # getLBracket
    ;
rbracket :
    RBracket # getRBracket
    ;
assign :
    Assign # getAssign
    ;
lineend:
    LineEnd # getLineEnd
    ;
comma:
    Comma # getComma
    ;
```

(3).在 basevisitor.java 的方法内加打印的函数

比如

```
/**
     * {@inheritDoc}
     *
     * <p>The default implementation returns the result of calling
     * {@link #visitChildren} on {@code ctx}.</p>
     */
    @Override public T visitGetRCurly(cmmParser.GetRCurlyContext ctx) {
    System.out.println("It is }");
    return visitChildren(ctx);
    }
```

这里读到右括号就会输出 `It is }` 。

以及：

```
/**
     * {@inheritDoc}
     *
     * <p>The default implementation returns the result of calling
     * {@link #visitChildren} on {@code ctx}.</p>
     */
    @Override public T visitGetBreak(cmmParser.GetBreakContext ctx) {
    System.out.println("BREAK statement");
    return visitChildren(ctx);
    }
```

这里读到break关键字就会输出 `BREAK statement` 。

(4).编写main.java测试文件

```
/**
```

```
  * Created by puxuan on 2016/10/25.
  */

import java.io.FileInputStream;
import java.io.InputStream;
import org.antlr.v4.runtime.*;
import org.antlr.v4.runtime.tree.*;

public class Main {
    public static void main(String[] args) throws Exception {
        String inputFile = args[0];
        InputStream is = System.in;

        if ( inputFile!=null ) is = new
FileInputStream(inputFile);
        ANTLRInputStream input = new ANTLRInputStream(is);

        cmmlexer = new cmmLexer(input);
        CommonTokenStream tokens = new
CommonTokenStream(lexer);
        cmmParser parser = new cmmParser(tokens);
        ParseTree tree = parser.program();
        cmmBaseVisitor vist = new cmmBaseVisitor();
        vist.visit(tree);
    }
}
```

(5).打包成jar文件

3.使用环境

使用方法：在命令行环境跳到jar包的目录，然后

```
$ java -jar cmm.jar test.txt
```

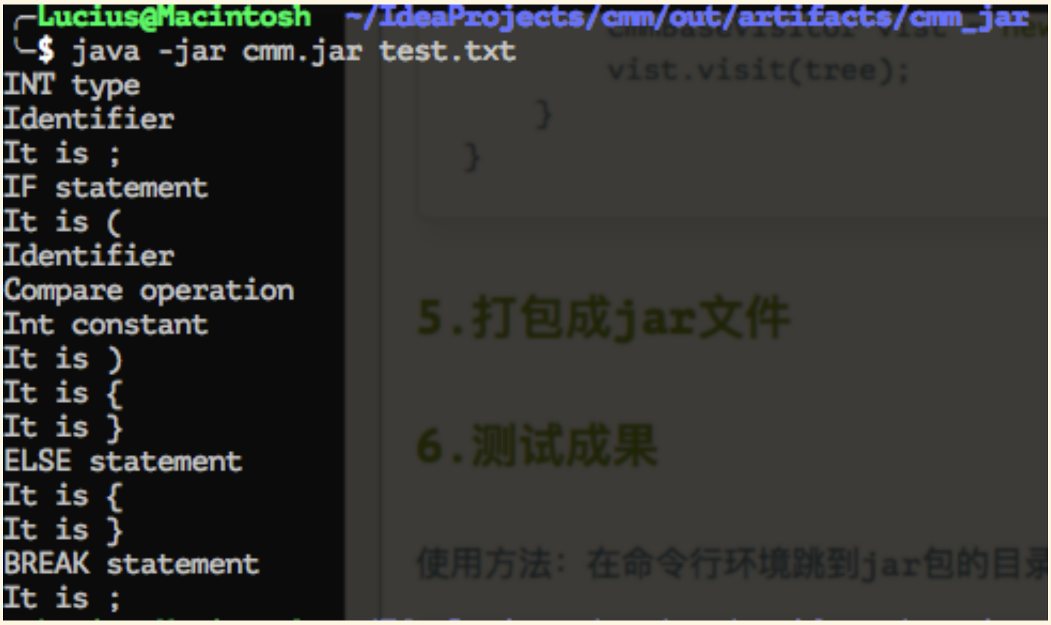
第一个参数是输入的文件，可以修改。

我使用的非常简单的test case：

```
int i;
```

```
  if (i > 1) {} else {}
  break;
```

输出结果为：



```
┌─Lucius@Macintosh  ~/IdeaProjects/cmm/out/artifacts/cmm_jar
└─$ java -jar cmm.jar test.txt
INT type
Identifier
It is ;
IF statement
It is (
Identifier
Compare operation
Int constant
It is )
It is {
It is }
ELSE statement
It is {
It is }
BREAK statement
It is ;
```