

## 1. Background:

### 1. Group member:

Name	Student Number	Work Content
郑啸宇	2013302580130	Lexical analysis, Syntactic analysis, Generate the syntax tree
赵一鸣	2013302580010	Semantic analysis Output the result
黄慧娟	2013302580007	Eliminate the comments, Design and implement UI, Test

### 2. Tools and Environment:

Operating system: windows8 /Mac OS

Development Environment: JAVA JDK 1.8

Development Tool: IntelliJ IDEA Community, JAVACC

## 2. CMM Grammar:

### 1. Lexical grammar:

SKIP: " "

| "\r"  
| "\t"  
| "\n"  
| "//" (~["\n", "\r"])\* ("\"|\"r\"|\"r\n")  
| "/\*" (~["\*"])\* "\*" (~["/"] (~["\*"])\* "\*" )\* "/" ;

ASSIGN: "=";

GT: ">" ;

LT: "<" ;

EQ: "==" ;

LE: "<=" ;

GE: ">=" ;

NE: "!="

| "<>" ;

AND: "&&" ;

OR: "||" ;

COMMA: "," ;

PLUS: "+" ;

MINUS: "-" ;

MULTIPLY: "\*" ;

DIVIDE: "/";  
 MOD: "%";  
 LC: "(";  
 RC: ")";  
 LM: "[";  
 RM: "]"  
 LB: "{";  
 RB: "}";  
 BOOLEAN: "boolean";  
 CHAR: "char";  
 INT: "int";  
 REAL: "real";  
 DIGIT : [ "0"-"9" ];  
 INTEGER:[ "1"-"9" ](DIGIT)\*  
           |(DIGIT)+;  
 STRING: [ "\"" ]( ~[ "\"" ])\*[ "\"" ];  
 DOUBLE:(DIGIT)+ "."  
           |(DIGIT)+ "."(DIGIT)+  
           | "."(DIGIT)+;  
 BREAK: "break";  
 ELSE: "else";  
 WHILE: "while";  
 IF: "if";  
 TRUE: "true";  
 FALSE: "false";  
 WRITE: "write";  
 READ: "read";  
 END: "end";  
 LETTER:[ "a"-"z", "A"-"Z", "\$" ];  
 ID: LETTER( LETTER| DIGIT| "\_" )\*;

## 2. Syntactical grammar

program: statement statements;  
 statements: program?;  
 statement: assignment ";"  
           | declaration ";"  
           | ifstatement  
           | whilestatement  
           | readstatement ";"  
           | writestatement ";"  
 ifstatement: IF LC condition RC  
               LB statements RB  
               ( ELSE ( LB statements RB |statement ))\*;  
 condition: expression ( EQ expression

```

        | NE expression
        | GT expression
        | LT expression
        | LE expression
        | GE expression
        | AND expression
        | OR expression)?;
expression: term ( PLUS term
        | MINUS term )*;
term: (factor)? ( MULTIPLY factor
        | DIVIDE factor
        | MOD factor )*;

factor: id
        | LC expression RC
        | INTEGER
        | REAL
        | STRING
        | bool;
id: ID ( LM expression RM )?;
bool: TRUE
        | FALSE;
whilestatement: WHILE LC condition RC ( LB statements RB );
declaration: intDec
        | realDec
        | boolDec
        | charDec;
intDec: INT ids;
realDec: REAL ids;
boolDec: BOOLEAN ids;
charDec: CHAR ids;
ids: id ( COMMA id )*;
assignment: ids ASSIGN expression;
readstatement: READ LC id RC;
writestatement: WRITE LC expression RC;

```

### 3. Design and Implementation

Our CMM interpreter firstly we write a grammar and use javacc and jjtree to analyse and create the syntax tree according to the grammar given above.

All the file give by Javacc is in the directory "\ast"

Since the grammar is not very complex, after we get the syntax tree, we don't change it into three address code but try to run it directly,

when we try to implement the program, symbol table is used to store some information about

variables, as for the array, we use another symbol table to store them, the symbol table class is in the directory "\run", they are packaged as a package run. The symbol table has level attribute such that local variable is allowed in the program.

The class "Run" is the main class that will traverse the tree to run the program and find the error. If the semantic analysis find no error, the method `getErrorMessage()` in class Run will be empty, otherwise some error message such as "try to divide 0", "array subscribe out of range" will be included. If the program can pass the semantic analysis, it will give the output result by use `getResult()` method. When traversing the tree, different type of node will do some proper behaviors, the `arithCalc(node)` method is used for calculate values for the expression subtree with root node, other type of node such as "read", "write", "if", "while", etc are implemented separately.

The variable types we support are int, real, boolean, char(which is string)

So there will be some details about the calculation:

Here is some settings that should be noticed about cast between different data types:

- when try to cast boolean to int, true will be considered as 1, false is 0, same cast to real

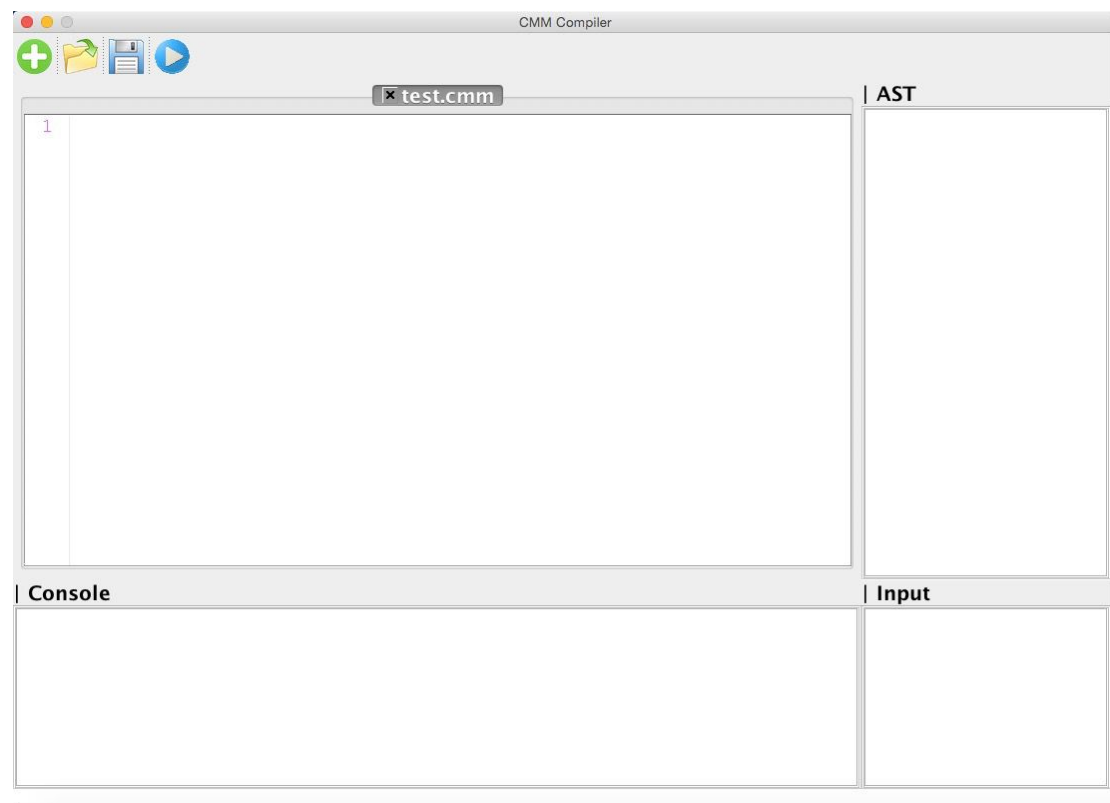
- when try to cast int to boolean, 0 is false, else is true

- when try to cast int to real, value will be same

- the other way to do casting will receive error message.

In the interpreter if you write `while(true){}`, that is, loop condition is const and true, infinite loop error will be reported.

The class UI is the main class and the UI implementation file. We realize the open file, save file, create file operations. And the interface is user-friendly, shown as below.

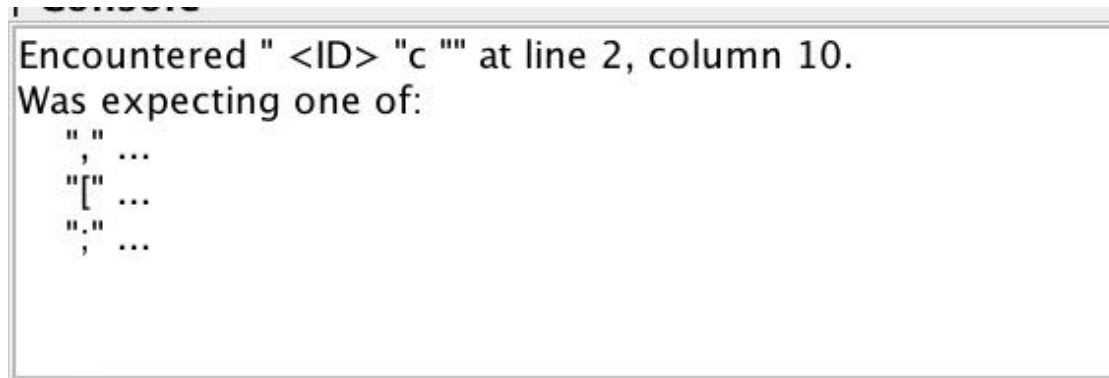


As we can see, besides the file operations, we support showing AST, input box for read

operation, and console. In addition, we support multi-file:



If we encounter the grammar or lexical error, the output will tell you which line the error lies.



If we don't input values for the read operation, it will tell you how many values you need to input.



## 4. Type Rules:

There are several rule different from the CMM examples:

1. Declaration and Assignment should be in different sentences.
2. After “if” and “while”, there must be “{”
3. If real number is less than  $10^{(-12)}$ , it will be considered as 0
4.  $\neq$  and  $\lt$  both mean not equal
5. If's condition should be same type like “if (1.2>1.0)” instead of “if(1.2>1)”
6. “#n#” is used to create another line
7. True and false can not be in the “while” and “if” condition directly.

## 5. Testing

For testing, we create several files to test, here are the screen shot and source code of them.

阶乘测试:

```
int a ;
int factorial;
a =6;
factorial =1;
while( a <> 0 )
{
    factorial = factorial * a;
    a = a -1;
}
write( factorial );
/*output:
factorial :
720
*///:~
```

output:

## Console

No Grammer Error  
720

数组测试:

```
real realArray[6];
int  intArray [2];
real a ;
a = 2.0;
int b;
realArray[b] = 2;
realArray[1] = a;
realArray[2] = intArray[0];
a = 0.9;
realArray[3] = a;
realArray[4] = 5 - 2 / (4.0 - 3) * 2.5 + 0.01;
```

```
read(realArray[5]);
int x ;
while(x < 6)
{
    write(realArray[x]);
    x = x + 1;
}
```

输入 9，得到结果如图

## Console

```
No Grammer Error
2 2.0 0 0.9 0.01 9.0
```

```
//字符加法测试
char a,b,c ;
a ="I ";
c ="Like ";
b ="You";
write(a+c+b);
```

## Console

```
No Grammer Error
I Like You
```

if 语句

```
/**
 * IF-ELSE 测试
 * IF-ELSE 测试 主要是:
 * 1.条件判断
 * 2.语句执行
 * 3.IF-ELSE 嵌套
 */
```

```
int a;
a=1;
if(a * 5 == 5)
{
    real r ;
    r = 2.0;
    if(r>1.0)
    {
        write(r);
    }
}
else
{
    a=5;
    write(a);
}

//
int aa ;
aa =3;
if(aa < 4)
{
    if(2 < aa)
    {if(aa <> 3)
        {write(aa);}
        else
            write(aa-2);
    }
}
```

```
/*output:
2.0
1
*///:~
```



## | Console

No Grammer Error  
2.0 1

注释嵌套:

```
/**
 * 注释报错 测试
 * 2010-11-13
 * 邱忠磊
 *
 * 注释报错 测试 主要是:
 * 1.注释嵌套
 * 2.多行注释无结尾
 */
```

```
int l[6];
int i;
i =0;
```

```
while(i<6)
{
    l[i] = i;
    i=i+1;
}
```

```
/*output:
没什么报错: 报错信息
*///:~
```

```
//注释嵌套
/* ffff /* mmmm */ ffff */
//多行注释无结尾
/* 不报错.../*/
```

## Console

No Grammer Error

ifwhile 结合:

```
/**
 * IF-ELSE 与 WHILE 混合嵌套 测试 主要是:
 * 1.条件判断
 * 2.语句执行
 * 3.IF-ELSE 与 WHILE 混合 嵌套
 */
int a ;
a =4;
while(a <> 0)
{
    int j ;
    j =a;
    while(j <> 0)
    {
        if(j/2 <> 1)
            {write(j);}
        j = j-1;
    }
    if( a < 2)
    {
        write(a);
    }
    else
        write(a+3);
    a = a -1;
}
/*output:
4
1
7
1
6
1
5
1
1
```

\*///:~

## | Console

No Grammer Error  
4 1 7 1 6 1 5 1 1

运算测试

```
int a ;
```

```
a = 2 * 4; //a = 8
```

```
if( a<> 8)
```

```
    {write(0);}
```

```
else
```

```
    write(1);
```

```
real r ;
```

```
r = 2 * (3.0 - 2.10) - 0.9 * (2.50 / 1.25 );    //r = 0.0
```

```
if( r <> 0.0)
```

```
{
```

```
    write(0);
```

```
}
```

```
else
```

```
    write(1);
```

```
real b;
```

```
b =4.0001;
```

```
write(b);
```

```
int x ;
```

```
x =69*96;
```

```
int y  ;
```

```
y =60*60;
```

```
write(x / y);
```

```
/*output:
```

```
1
```

```
1
```

```
4.000001
```

```
1
```

\*///:~

## | Console

No Grammer Error  
1 1 4.0001 1

冒泡排序:

/\*冒泡排序\*/

/\*

啦啦啦

\*/

int p[10];

int a;

a = 0;

while(a < 10)

{

    read(p[a]);

    a = a + 1;

}

int i, n, pos;

n = 10;

i = 0;

while(i < n)

{

    pos = i + 1;

    while(pos < n)

    {

        if(p[pos] > p[i])

        {

            a = p[pos];

            p[pos] = p[i];

            p[i] = a;

        }

        pos = pos + 1;

    }

    i = i + 1;

}

```

write("The array after sorting is:#n#");//#n#为换行符
i = 0;
while(i <= 9)
{
    write(p[i]);
    write(" ");
    i = i + 1;
}
input 0 1 2 3 4 5 6 7 8 9

```

```

No Grammer Error
The array after sorting is:
9 8 7 6 5 4 3 2 1 0

```

```

/*语义测试
包括:
类型检查
声明检查
赋值检查
分母为零
数组越界
IF 语句条件为布尔值或者 1 0
real 值极小的情况
强制转换
*/
int a ;
real b[5];
char d;

d = 1;
b[1] = "a";
a = 0.1;//类型检查

int b[3];//声明检查
int m[3];
m[3] = 0;//数组越界 参数为非整形
m[2.5]=0;

```

## | Console

Error

Error: int is not allowed to assign to string or error type

Error: String is not not allow to assigned to other type variable

Error: real type is not allowed to be assigned to other type

Error: array variable b has been declared before

Error: the subscribe of array m is larger then array length, please check

Error: the subscribe of array m must be integer

```
int k ;
```

```
k =1;
```

```
write(k/0);//分母为零
```

```
read(k);//输入"a" 1.5 true 0
```

Error

Error: divide by zero error

Error: Unable to output the error type value in the write operation

Error: the input is not a integer!

```
real m ;
```

```
m =1.0;
```

```
real i;
```

```
i =2.0;
```

```
int n;
```

```
n = 1;
```

```
boolean c ;
```

```
c =true;
```

```
char d ;
```

```
d = "true";
```

```
if(m){write(1);}//错误
```

```
if(n){write(1);}//正确
```

```
if(c){write(1);}//正确
```

```
if(i>m){write(1);}//正确
```

```
if(d){write(1);}//错误
```

```
real test ;
```

```
//默认 real 小于 0.00000000000001 时为 0
```

```
test = 0.0000000000000001;
```

```
if(test>0.0){
```

```
write(test);}
```

```
test = 0.1;
```

```
if(test>0.0){  
write(test);
```

## Console

Error

Error: if condition can't be char(string) type or real type

Error: if condition can't be char(string) type or real type

```
}
```

注释掉错误后

## Console

No Grammer Error

1 1 1 0.000000000000000001 0.1

```
boolean c;//强制转换  
c = true;  
int a;  
real b;  
a = 3/4;  
b = 3.0/4.0;  
write(a);  
write(b);  
b = a;  
write(b);  
a = b;//错误 注释掉可以显示其他结果  
a =c;  
write(a);
```

Error

Error: real type is not allowed to be assigned to other type

注释掉错误以后

**| Console**

No Grammer Error  
0 0.75 0 1

死循环:

```
boolean a;  
a =true;  
while(a){  
write(a);  
}
```

**| Console**

Error  
Error: while dead loop

## 6.How to Use

Use java -jar

we can open the jar file, and the JDK version is 1.8.



A screenshot of a macOS terminal window. The title bar at the top reads 'zhengxiaoyu — java — 80x24'. The terminal content shows the last login time as 'Sun Jan 10 21:29:55 on ttys000'. The user 'zhengxiaoyu' is at the 'zhengxiaoyudeMacBook-Air' prompt. They have entered the command 'java -jar /Users/zhengxiaoyu/Desktop/compiler/CMMCompiler.jar'. The command has been executed, and a cursor is visible on the line following it.

```
zhengxiaoyu — java — 80x24
Last login: Sun Jan 10 21:29:55 on ttys000
zhengxiaoyudeMacBook-Air:~ zhengxiaoyu$ java -jar /Users/zhengxiaoyu/Desktop/com
plier/CMMCompiler.jar
```

## 7.Conclusion:

To complete the whole work, we tried javacc, antlr, flex, bison, flex++, bison++, and realizing by pure java, trying something we don't know, in our opinions, always means something. And in conclusion, we will introduce something that we may be different from others:

- 1.UI supporting create file, open file, save file, and multi-file, showing the AST
- 2.Highlight the key word and comments, show the line number
- 3.Support string add operation
- 4.Report for dead loop
- 5.Real number's absolute value less than  $10^{(-12)}$  will be 0, considering the deviation
- 6.Support Boolean value

## 8.Reference:

- [1] Java - How to program 9<sup>th</sup> edition
- [2] Java 从入门到精通
- [3] Compiler Construction Principles and Practice
- [4] Compilers: principles, technique and Tools Second Edition