**《编译原理》计算器编译器参考文档**

王泽鹏 2017301510036 网安二班

2019.10.12

**文法（输入文件流的每行代码格式）：**

stmt -> declaration | assign || print ;

declaration -> type var;

type -> int | float;

assign -> var=expression;

var -> [a-zA-Z]([a-zA-Z]|[0-9])\*

expression -> expression+term | expression-term | term | var

term -> term\*factor | term/factor | factor

factor -> (expression) | number

number -> ([1-9][0-9]\*) | [1-9][0-9]\*.[0-9]+ | 0 | 0.[0-9]+

print -> write(var);

**代码结构：**

Main.cpp：入口文件

MyStack.h：存放有关栈操作的函数

Analyse.h：主要函数头文件，包括对四则运算的求值、对文件流句子的读取与解析等关键函数。

**计算表达式使用说明：**

计算表达式只能作为赋值语句的右值而使用，其中表达式可以包含已声明且赋值的变量，进行符合规则的四则运算，但对于负数的读取因时间关系没有实现，即类似：a=(-3)\*5; 的表达式无法运行，在测试时需注意。对于未初始化或未定义变量、除数为零等情况都会有相应的错误提示。

其他使用细节包括：除声明语句外，其他语句中避免使用多余空格；注释使用‘#’实现；代码末尾使用’.’作为文件读取的结束符。

此外，该编译器实现了syntax error、重复声明、句末分号检查等其他常见错误，详细请看以下测试用例。

**测试用例：**

#Calculator test

int c#short of ";"

float c=1.2;

double b;#not allowing type:"double"

int aa;

int aa;#repeatedly statement

float d;

float cc;

int u;

int v;

aa=17;

cc=abc+1;#undefined identifier

write(cc);

cc=aa+(3+aa)\*4.5;

d=aa/0;#divided by zero

u=u+1;

syntax\_error\_test1#syntax error

syntax\_error\_test2;#syntax error

write(aa);

write(cc);

write(abc);#undefined identifier

write(u);#uninitialized identifier

write(v);#uninitialized identifier

write(d).#end of code

**测试输出结果如下：**

