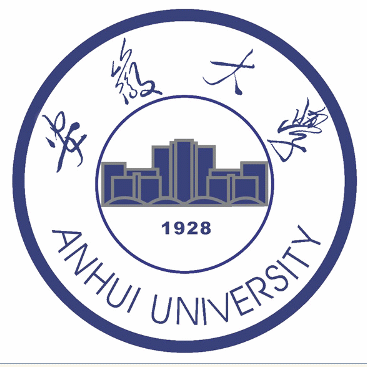
**《编译原理实验报告》**

****

**论文名称：**　　　词法分析程序

**年级专业：** 15级软件工程

**姓名学号：**何铭春 E21514033

## 一.实验内容

词法分析程序

待分析的简单语言的词法：

关键字：begin if then end (小写)

运算符：:= + - \* / > <

界符：; #

标识符：ID=letter (letter|digit)\*

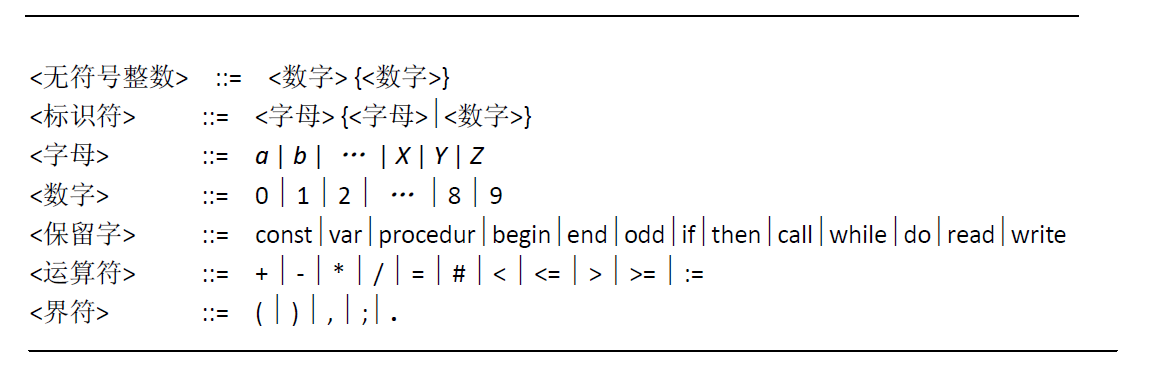
常数：NUM=digit digit\*

要求编写词法分析程序，

键盘输入源程序(以#结束)，输出单词符号的二元式。

（标识符1，常数2，关键字3，运算符4，界符5）

Ps/2:词法分析词法



输入：

begin

x:=9;

if x>0 then x:=2\*x+1/3

end

#

输出：

3 begin

1 x

4 :=

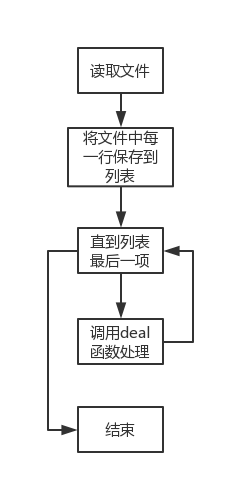
2 9

5 ;

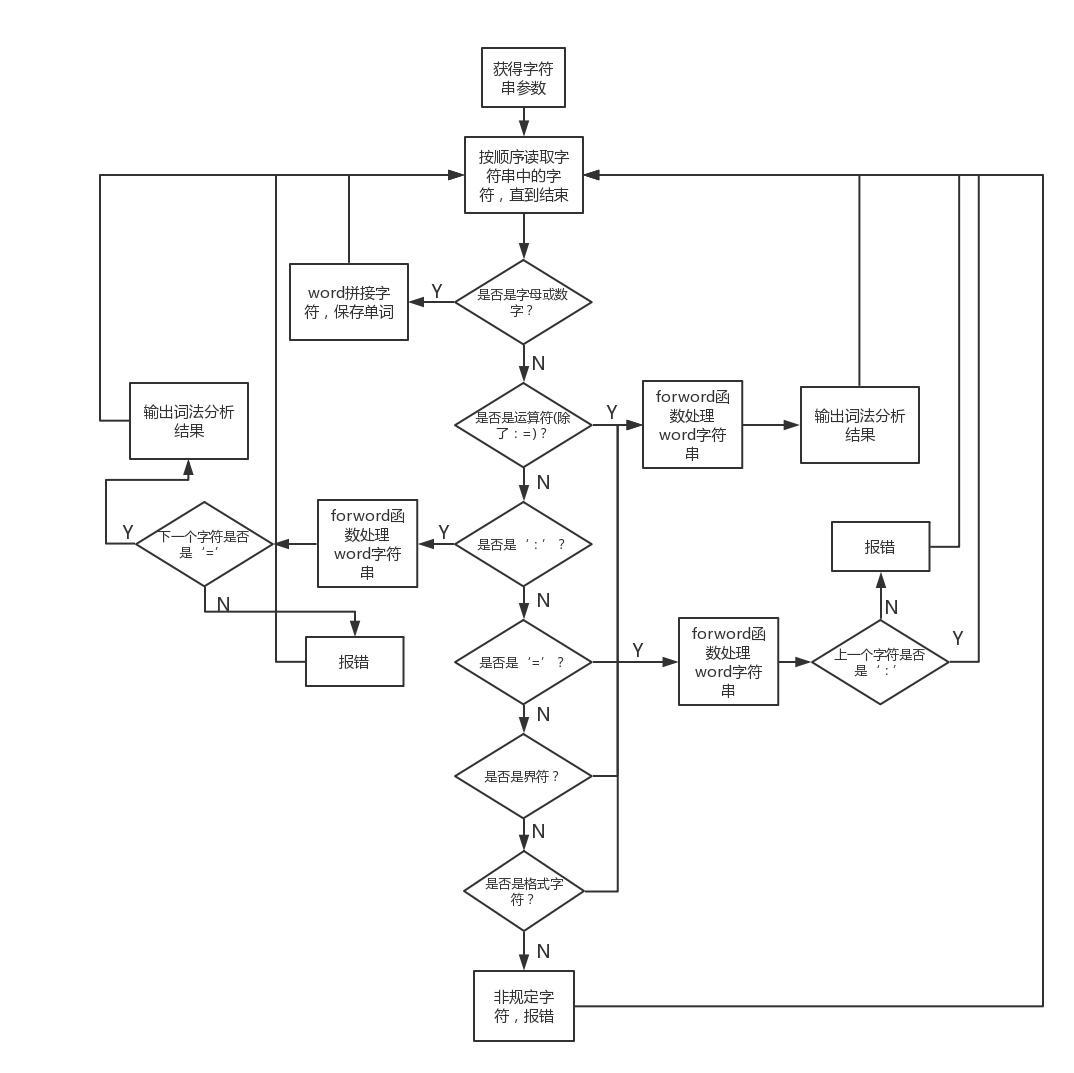
…….

## 二.基本思想（流程图）

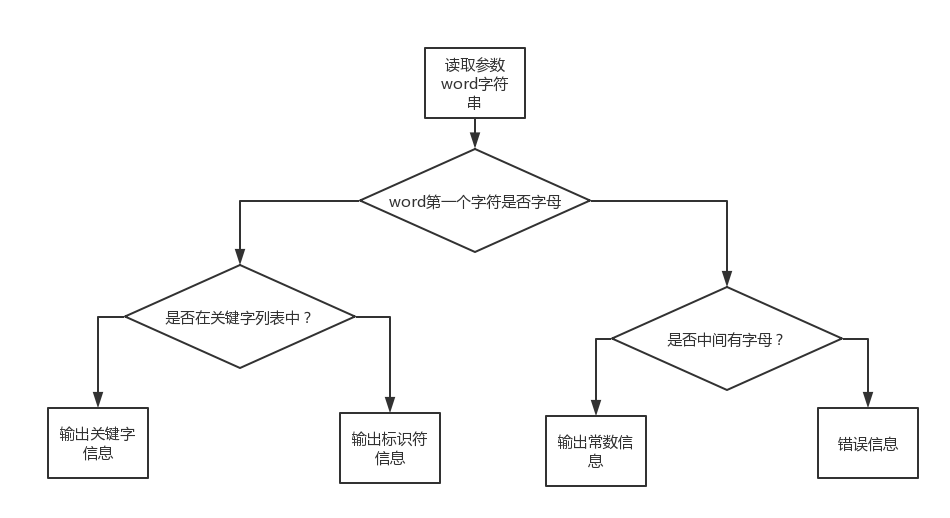
主函数：



deal函数流程图：



forword函数:



## 三.核心算法

1. forword函数

功能：对读取的字符串按照要求分类操作

输入：字符串word

输出：word分析结果

算法：根据word字符串第一个字符是数字还是字母判断，如果是字母分为关键词和标识符。如果是数字判断其中是否含有字母，有字母给出错误提示，没有字母为常数。

2.deal函数

功能：对字符串中字符一个个分析操作

输入：字符串work

输出：词法分析结果

算法：按照字符的不同分为7种情况：

①字母和数字：

字符串word拼接字符，待后续forword函数

②运算符（除了：=）

运行forword函数

输出运算符分析信息

③字符’：’

如果下一个字符是’=’，输出分析信息

否则输出错误提示

④字符’=’

如果上一个字符是’：’，什么都不做

否则输出错误提示

⑤界符（；，#）

运行forword函数

输出运算符分析信息

⑥格式字符（\n,\t,空字符）

运行forword函数

⑦其他非规定字符

输出错误提示

3.主函数

功能：读取文件并调用函数

输入：TXT文件

输出：词法分析结果

算法：

①读取文件

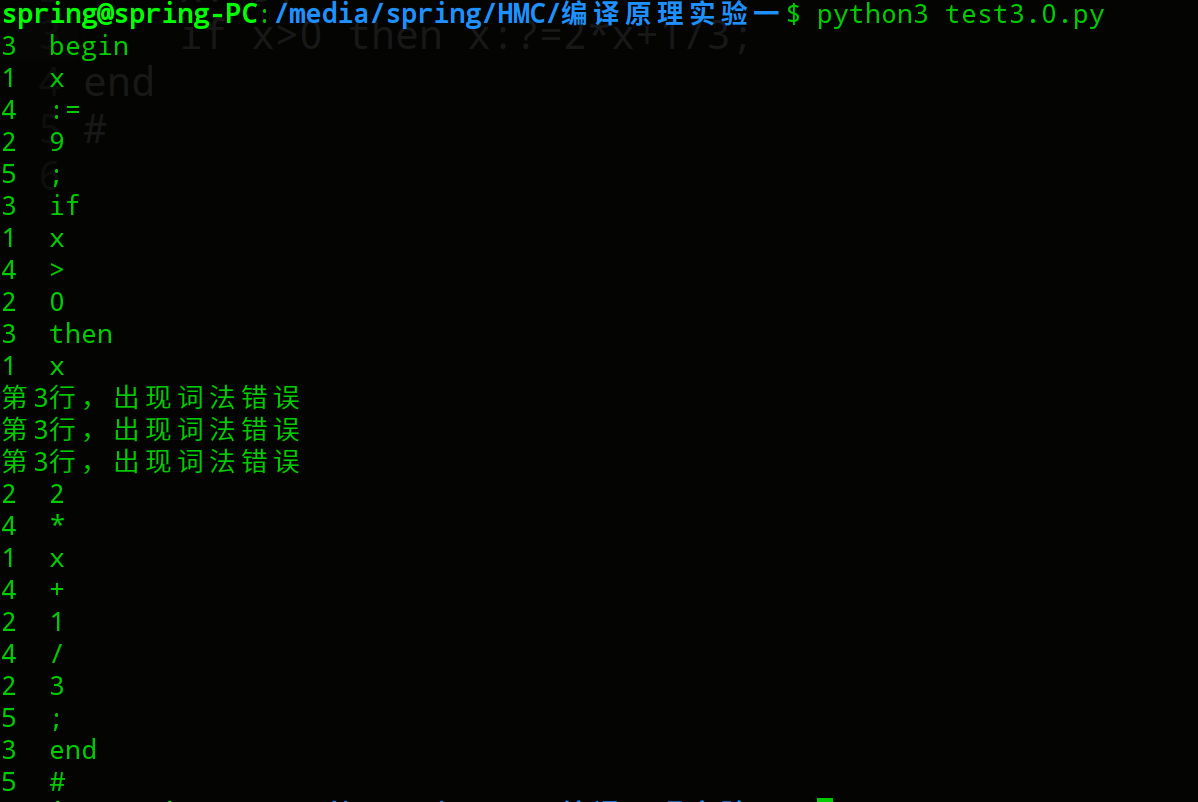
②对字符串调用函数

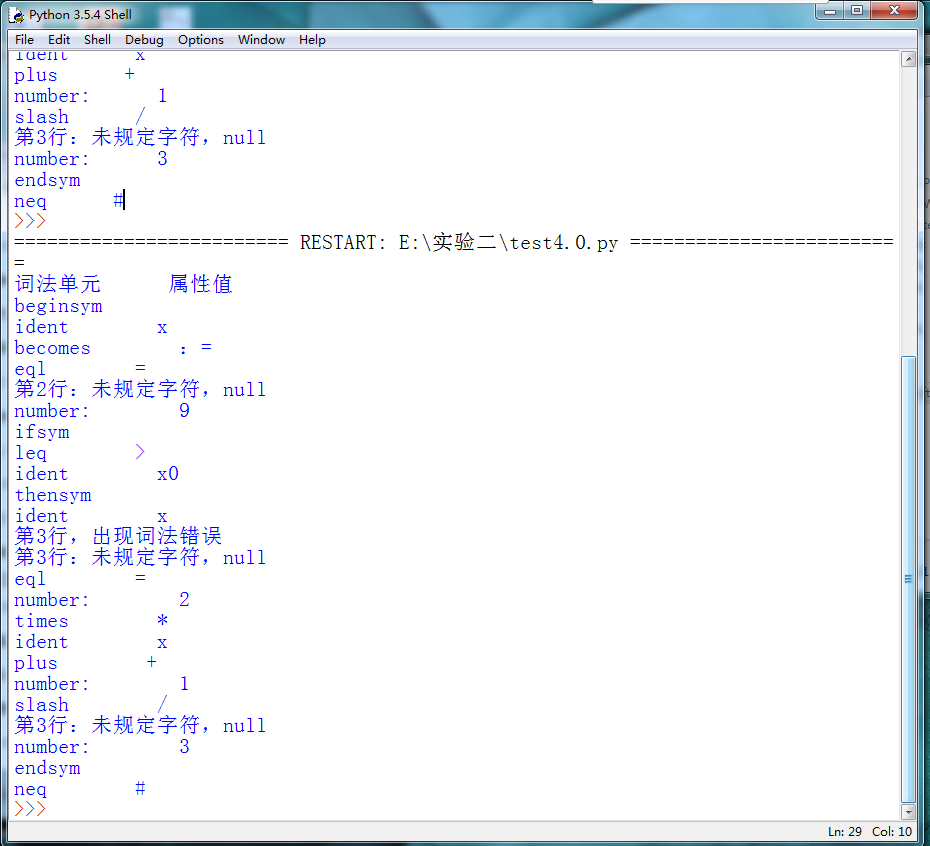
## 四.实验结果

PPT示例代码分析结果：



添加了if x>0 then x:?=2\*x+1/3错误





## 五.实验心得

1.使用一个多重嵌套的条件语句来作为程序主体。

2.对于数字和字母采用一个拼接的字符串，然后调用处理函数来判断是关键字还是标识符、常数、错误输入。

3.在调用条件上，在遇到非数字、字母时调用处理函数。

4.对于：=这个特殊符号，遇到：时向下继续读取一个字符，如果是=，则输出分析结果，否则报错。遇到=，则判断前一个字符。

5.对于不在规定范围的字符一律判定为错误，例如’?’。

6.使用字典保存分类结果

## 六．源代码

#关键词集合，用于判断是否是关键词

keywords={'const':'constsym','begin':'beginsym','if':'ifsym','then':'thensym','end':'endsym','var':'varsym','procedur':'procsym','odd':'oddsym','call':'callsym','while':'whilesym','do':'dosym','read':'readsym','write':'writesym'}

#运算符集合，用于判断

opration={'+':'plus','-':'minus','\*':'times','/':'slash','=':'eql','#':'neq'}

#界符

border={'lparen':'(','reparen':')','comma':',','semicolon':';','period':'.'}

#word字符串，用于将字符拼接成字符串

word=''

#特殊格式字符

format=[' ','\n','\t']

#对于拼接形成的字符串进行判断

def forword():

global word

#flag用于标记字符串是否为开头数字中间字母的错误

flag=0

#判断word是否为空

if(len(word)!=0):

#判断第一个字符是否字母

if word[0].isalpha():

#判断字符串是否为关键词

if word in keywords.keys():

print(keywords[word])

else:

print('ident'+' '+word)

else:

#对于数字开头的字符串，判断其中是否夹杂字母

for xyz in word:

if xyz.isalpha():

flag=1

break

#根据flag判断字符串是否错误

if flag==1:

print('第'+str(num+1)+'行，出现词法错误')

else:

print('number: '+word)

word=''

#接受一个字符串，循环判断字符

def deal(work):

global word

for i in range(0,len(work)):

#判断字符是否为字母，如果是拼接word

if work[i].isalpha() or work[i].isdigit():

word=word+work[i]

#判断字符是否为运算符，如果是就输出

elif work[i] in opration.keys():

forword()

print(opration[work[i]]+' '+work[i])

#判断字符是否为运算符:=，如果是就输出

elif work[i]=='<':

if work[i+1]=='=':

print('gtr <=')

else:

print('lss <')

elif work[i]=='>':

if work[i+1]=='=':

print('geq >=')

else:

print('leq >')

elif work[i]==':':

forword()

if work[i+1]=='=':

print('becomes ：=')

else:

print('第'+str(num+1)+'行，出现词法错误')

elif work[i]=='=':

if work[i-1]==':':

pass

elif work[i-1]=='<':

pass

elif work[i-1]=='>':

pass

else:

print('第'+str(num+1)+'行，出现词法错误')

#判断字符是否为界符，如果是就输出

elif work[i] in border.keys():

forword()

print('界符 '+border[work[i]])

#判断字符是否是格式字符

elif work[i] in format:

forword()

else:

print('第'+str(num+1)+'行：未规定字符，null')

temp=[]

work=[]

#打开文件

with open('test1.txt','r') as f:

temp.extend(f.readlines())

print("词法单元 属性值")

#对于每一行使用deal函数，num用于错误提示表示行数

for num in range(0,len(temp)):

deal(temp[num])