



# 语法分析作业简介

---

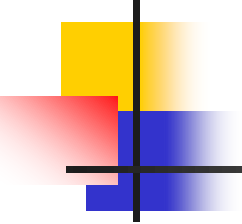
2019年10月15日

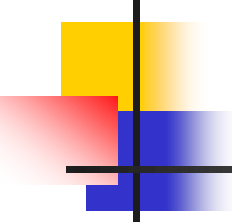


# 目标和要求

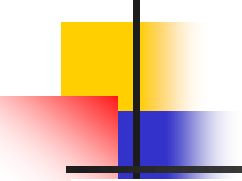
---

- 根据具体文法编写语法分析程序，考核学生对语法分析方法的掌握情况，培养学生编写符合文法的自顶向下语法分析程序的能力。
- 学生在理论课所学语法分析方法基础上，设计实现语法分析程序，并按规定格式输出单词信息（沿用词法分析器的输出）和语法成分名称。

- 
- 请根据给定的文法设计并实现语法分析程序，能基于上次作业的词法分析程序所识别出的单词，识别出各类语法成分。输入输出及处理要求如下：
    - （1）需按文法规则，用递归子程序法对文法中定义的所有种语法成分进行分析；
    - （2）为了方便进行自动评测，输入的被编译源文件统一命名为**testfile.txt**；输出的结果文件统一命名为**output.txt**；结果文件中包含如下两种信息：
      - 1）按词法分析识别单词的顺序，按行输出每个单词的信息（要求同词法分析作业，对于预读的情况不能输出）。
      - 2）在下列高亮显示的语法分析成分分析结束前，另起一行输出当前语法成分的名字，形如“<常量说明>”



<加法运算符> ::= + | -  
<乘法运算符> ::= \* | /  
<关系运算符> ::= < | <= | > | >= | != | ==  
<字母> ::= \_ | a | . . . | z | A | . . . | Z  
<数字> ::= 0 | <非零数字>  
<非零数字> ::= 1 | . . . | 9  
<字符> ::= '<加法运算符>' | '<乘法运算符>' | '<字母>' | '<数字>'  
<字符串> ::= " {十进制编码为32,33,35-126的ASCII字符} "  
<程序> ::= [<常量说明>] [<变量说明>] {<有返回值函数定义> | <无返回值函数定义>} <主函数>  
<常量说明> ::= const <常量定义>; { const <常量定义>;}  
<常量定义> ::= int <标识符> = <整数> { <标识符> = <整数> }  
                  | char <标识符> = <字符> { <标识符> = <字符> }  
<无符号整数> ::= <非零数字> { <数字> } | 0  
<整数> ::= [+ | -] <无符号整数>  
<标识符> ::= <字母> { <字母> | <数字> }  
<声明头部> ::= int <标识符> | char <标识符>  
<变量说明> ::= <变量定义>; { <变量定义>;}  
<变量定义> ::= <类型标识符> (<标识符> | <标识符> '[' <无符号整数> ']') { (<标识符> | <标识符> '[' <无符号整数> ']') }  
          //<无符号整数>表示数组元素的个数，其值需大于0  
<类型标识符> ::= int | char



```

<有返回值函数定义> ::= <声明头部>'(' <参数表> ')' '{' <复合语句> '}'
<无返回值函数定义> ::= void <标识符>'(' <参数表> ')' '{' <复合语句> '}'
<复合语句> ::= [ <常量说明> ] [ <变量说明> ] <语句列>
<参数表> ::= <类型标识符> <标识符> { <类型标识符> <标识符> } | <空>
<主函数> ::= void main '(' ')' '{' <复合语句> '}'
<表达式> ::= [ + | - ] <项> { <加法运算符> <项> } // [+|-]只作用于第一个<项>
<项> ::= <因子> { <乘法运算符> <因子> }
<因子> ::= <标识符> | <标识符>'(' <表达式> ')' | <标识符>'(' <表达式> ')' <整数> | <字符> | <有返回值函数调用语句>
<语句> ::= <条件语句> | <循环语句> | '{' <语句列> '}' | <有返回值函数调用语句>;
| <无返回值函数调用语句>; | <赋值语句>; | <读语句>; | <写语句>; | <空>; | <返回语句>;
<赋值语句> ::= <标识符> = <表达式> | <标识符>'(' <表达式> ')' = <表达式>
<条件语句> ::= if '(' <条件> ')' <语句> [ else <语句> ]
<条件> ::= <表达式> <关系运算符> <表达式> //整型表达式之间才能进行关系运算
| <表达式> //表达式为整型，其值为0时条件为假，值不为0条件为真
<循环语句> ::= while '(' <条件> ')' <语句> | do <语句> while '(' <条件> ')' | for '(' <标识符> = <表达式>; <条件>; <
标识符> = <标识符> (+|-) <步长> ')' <语句>
<步长> ::= <无符号整数>
<有返回值函数调用语句> ::= <标识符>'(' <值参数表> ')'
<无返回值函数调用语句> ::= <标识符>'(' <值参数表> ')'
<值参数表> ::= <表达式> { <表达式> } | <空>
<语句列> ::= { <语句> }
<读语句> ::= scanf '(' <标识符> { <标识符> } ')'
<写语句> ::= printf '(' <字符串> , <表达式> ')' | printf '(' <字符串> ')' | printf '(' <表达式> ')'
<返回语句> ::= return '(' <表达式> ')'

```

例如有如下程序段：

```
const int const1 = 1, const2  
= -100;  
const char const3 = '_';  
int change1;  
char change3;  
int gets1(int var1,int var2){  
    change1 = var1 + var2;  
    return (change1);  
}
```

则输出的结果文件具有的内容为：

```
CONSTTK const  
INTTK int  
IDENFR const1  
ASSIGN =  
INTCON 1  
<无符号整数>  
<整数>  
COMMA ,  
IDENFR const2  
ASSIGN =  
MINU -  
INTCON 100  
<无符号整数>  
<整数>  
<常量定义>  
SEMICN ;  
CONSTTK const  
CHARTK char  
IDENFR const3  
ASSIGN =  
CHARCON _  
<常量定义>  
SEMICN ;
```

```
<常量说明>  
INTTK int  
IDENFR change1  
<变量定义>  
SEMICN ;  
CHARTK char  
IDENFR change3  
<变量定义>  
SEMICN ;  
<变量说明>  
INTTK int  
IDENFR gets1  
<声明头部>  
LPARENT (  
INTTK int  
IDENFR var1  
COMMA ,  
INTTK int  
IDENFR var2  
<参数表>  
RPARENT )  
LBRACE {  
IDENFR change1
```

```
ASSIGN =  
IDENFR var1  
<因子>  
<项>  
PLUS +  
IDENFR var2  
<因子>  
<项>  
<表达式>  
<赋值语句>  
SEMICN ;  
<语句>  
RETURNTK return  
LPARENT (  
IDENFR change1  
<因子>  
<项>  
<表达式>  
RPARENT )  
<返回语句>  
SEMICN ;  
<语句>  
<语句列>
```

例如有如下程序段：

```
const int const1 = 1, const2
= -100;
const char const3 = '_';
int change1;
char change3;
int gets1(int var1,int var2){
    change1 = var1 + var2;
    return (change1);
}
```

则输出的结果文件具有的内容为：

```
<复合语句>
RBRACE }
<有返回值函数定义>
VOIDTK void
MAINTK main
LPARENT (
RPARENT )
LBRACE {
PRINTFtk printf
LPARENT (
STRCON Hello World
<字符串>
RPARENT )
<写语句>
SEMICN ;
<语句>
PRINTFtk printf
LPARENT (
IDENFR gets1
LPARENT (
INTCON 10
<无符号整数>
<整数>
```

```
<因子>
<项>
<表达式>
COMMA ,
INTCON 20
<无符号整数>
<整数>
<因子>
<项>
<表达式>
<值参数表>
RPARENT )
<有返回值函数调用语句>
<因子>
<项>
<表达式>
RPARENT )
<写语句>
SEMICN ;
<语句>
<语句列>
<复合语句>
RBRACE }
```

```
<主函数>
<程序>
```

注：未要求输出的语法成分仍需要进行分析

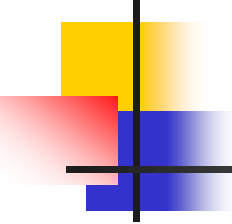


# 注意事项

---

- 输出有统一的要求仅为了便于进行自动评测，数据结构及与其他各逻辑组成部分的接口自行定义
- 未要求输出的语法成分仍需要进行分析
- 用C/C++开发均可，在提交作业时注意选择对应的开发语言选项（clang或clang++）
- 提交的作业源代码命名无统一要求，提交时选择所有源文件打包提交即可
- 自动评测在本课程首次推行，若发现问题请及时向老师或助教反映



- 
- 
- 课程设计有关的材料请从  
[judge.buaa.edu.cn](http://judge.buaa.edu.cn)教学平台“课件下载”中获取
  - 另外，需及时关注教学平台的公告、答疑论坛