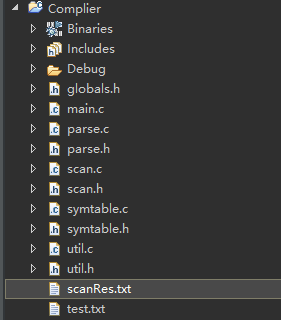
**许东泽 2015302462 2018/7/5**

**一 项目结构**

* **文件结构**



* **注释**

|  |  |
| --- | --- |
| 文件名 | 备注 |
| globals.h | 公共头文件, 包括符号表、token、树节点等的定义 |
| scan.h | 相应c文件的接口函数声明 |
| parse.h |
| systable,h |
| util.h |
| scan.c | 词法分析主体getToken()函数实现 |
| parse.c | 语法分析主体parse()函数实现 |
| systable.c | 符号表相关操作 |
| util.c | 包括字符串复制等的实用函数库 |
| test.txt | 测试代码 |
| scanRes.txt | 测试结果 |

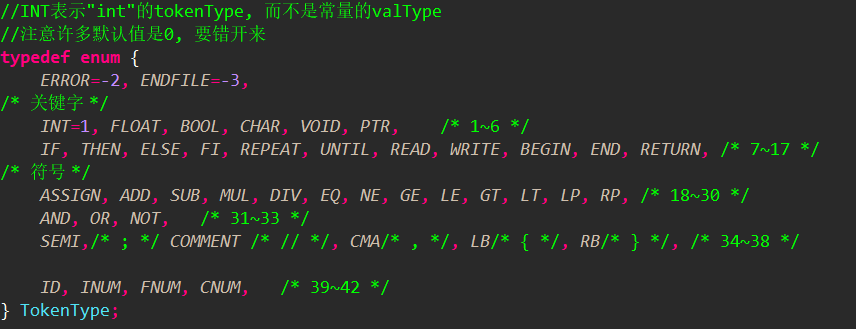
**二 词法分析**

|  |  |  |  |
| --- | --- | --- | --- |
| 序列号 | 集合 | 字符串表示 | 类型 |
| 1 | 关键字 | int | INT |
| 2 | float | FLOAT |
| 3 | void | VOID |
| 4 | char | CHAR |
| 5 | If | IF |
| 6 | Then | THEN |
| 7 | Else | ELSE |
| 8 | Fi | FI |
| 9 | repeat | REPEAT |
| 10 | until | UNTIL |
| 11 | begin | BEGIN |
| 12 | End | END |
| 13 | Read | READ |
| 14 | Write | WRITE |
| 15 | return | RETURN |
| 16 | 标识符 | [a-zA-Z]+ | ID |
| 17 | 数字 | [0-9]+ | INUM |
| 18 | [0-9]\*/.([0-9]+([+-]?[eE][0-9]+)?)? | FNUM |
| 19 | 运算符 | && | AND |
| 20 | || | OR |
| 21 | < | LT |
| 22 | > | GT |
| 23 | == | EQ |
| 24 | >= | GE |
| 25 | <= | LE |
| 26 | != | NE |
| 27 | + | ADD |
| 28 | - | SUB |
| 29 | \* | MUL |
| 30 | / | DIV |
| 31 | 括号 | ( | LP |
| 32 | ) | RP |
| 33 | { | LB |
| 34 | } | RB |
| 35 | 界符 | ; | SEMI |
| 36 | , | CMA |
| 37 | 注释 | %.\*% | COMMENT |
| 38 | 文件结束符 | EOF | ENDFILE |

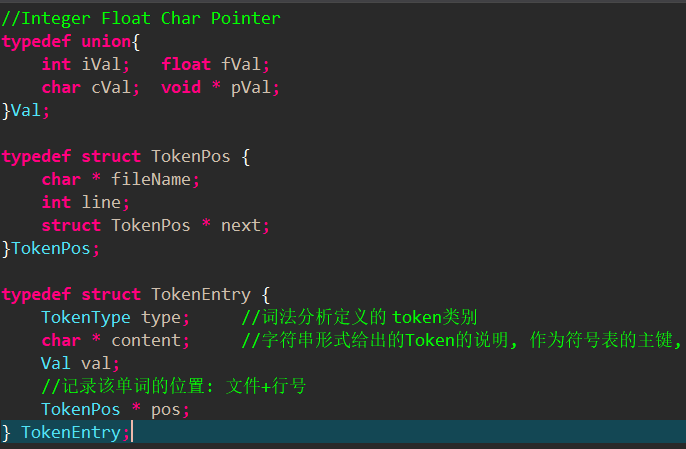
**scan状态转换图：略。**

**数据结构:**

* Globals.h : token类别。



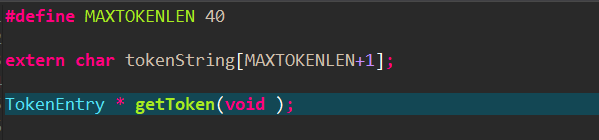
* Globals.h: Token 实体



**函数接口：**

* Scan.h:

tokenString : 当前分析的token的字符串表示，超出最大长度后超出部分忽略。



**三．语法分析**

**语法产生式设计**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 描述节点 | 产生式左端 | 产生式右端 | 节点类型 | 备注 |
| 1 | 函数 | Program | ProgramHead LB ProgramBody RB | ProgramK | funcdefsize++ |
| 2 | 函数头 | ProgramHead | TYPE ID LP ParaList RP | ProgramHeadK | 填充函数定义表 |
| 3 | 形参列表 | ParaList | TYPE ID ParaListSuf | ParaListK | 可选（可为NULL） |
| ParaLIstSuf | (CMA TYPE ID)\* | ParaListK |
| 4 | 函数体 | ProgramBody | TypeDeclar StmtList | ProgramBodyK |  |
| 5 | 变量声明 | TypeDeclar | DeclarList\* | TypeDeclarK | 可选（可为NULL） |
| 6 | 声明语句 | DeclarStmt | Type IDList SEMI | DeclatStmtK |  |
| 7 | 标识符列表 | IDList | ID IDListSuf | IDListK | 至少包含一个ID |
| IDListSuf | (CMA ID)\* | IDListK | 可选（可为NULL） |
| 8 | 语句列表 | StmtList | Stmt\* | StmtListK | 可选（可为NULL） |
| 9 | 语句 | Stmt | AssignStmt|IfStmt|RepeatStmt | 虚拟节点 | 虚拟节点 |
| 10 | 赋值语句 | AssignStmt | ID ASSIGN Exp SEMI | AssignStmtK | 将表达式赋给变量 |
| 11 | 选择语句 | IfStmt | IF LP Exp RP LB StmtList RB ELSE LB StmtList RB | IfStmtK | else部分可选 |
|
|
| 12 | 循环语句 | RepeatStmt | REPEAT LB StmtList RB UNTIL LP Exp RP | RepeatStmtK |  |
| 13 | 布尔表达式 | Exp | Exp01 ((AND|OR) Exp01)\* | 二元布尔运算链 | E\_BOOLK |
| 14 | 关系表达式 | Exp01 | Exp02 (ShipOp Exp02)\* | 二元关系运算链 | E\_SHIPK |
| 15 | 加减表达式 | Exp02 | Exp03 ((ADD|SUB)Exp03)\* | 加减运算链 | E\_ADDSUBK |
| 16 | 乘除表达式 | Exp03 | Factor ((MUL|DIV)Factor)\* | 乘除运算链 | E\_MULDIVK |
| 17 |  | Factor | ID | INUM | CNUM | FNUM | LP Exp RP | FuncCallExp | NULL |  | 终结符, 可为空 |
| 18 | 函数调用 | FuncCallExp | ID LP RParaList RP | E\_FUNCCALLK |  |
| 19 | 实参列表 | RParaList | Exp RParaListSuf | RParaListK | 可选（可为NULL） |
|  | RParaListSuf | (, Exp)\* | PRaraListK | 可选（可为NULL） |

采用递归下降语法分析。注意到该文法并非严格的LL(1) 语法：

在产生式17中 由于同时存在：

Factor -> ID

Factor -> FuncCallExp

两者first集合交集：ID。所以具有冲突。再次做了回溯处理：

记录当前token到临时变量，更新当前token为下一个token。即往前继续看一步，若为LP，选用 函数调用 产生式，否则选用 ID 产生式。

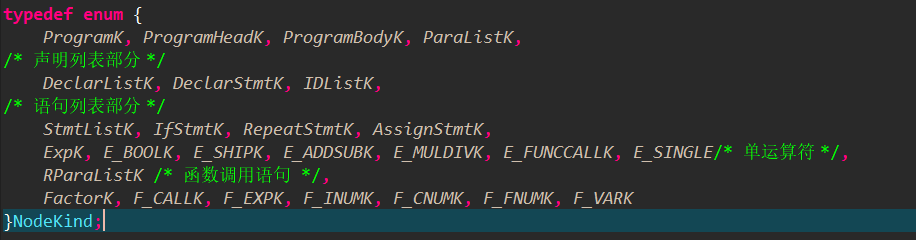
除此之外，没有其他的冲突，其余均可以正常使用递归下降分析。

**数据结构：**

* Parse.c:



* Globals.h: 分析树节点类型



* Globals.h: 分析树节点结构。 Size: 20Byte。详见代码。

Struct TreeNode {

类型：NodeKind

Union{

对每一个产生式，定义一个节点结构体。

}

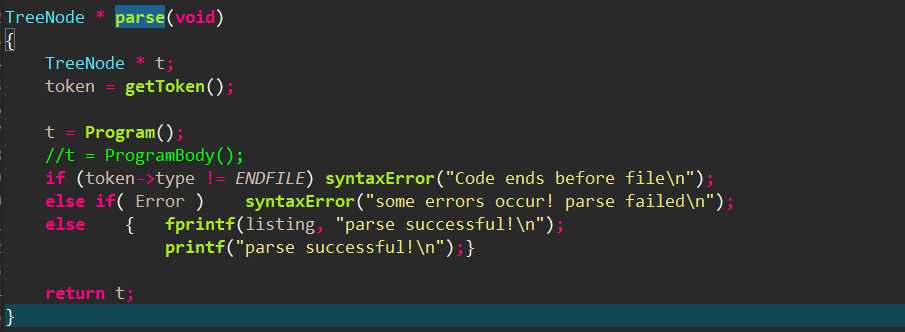
}

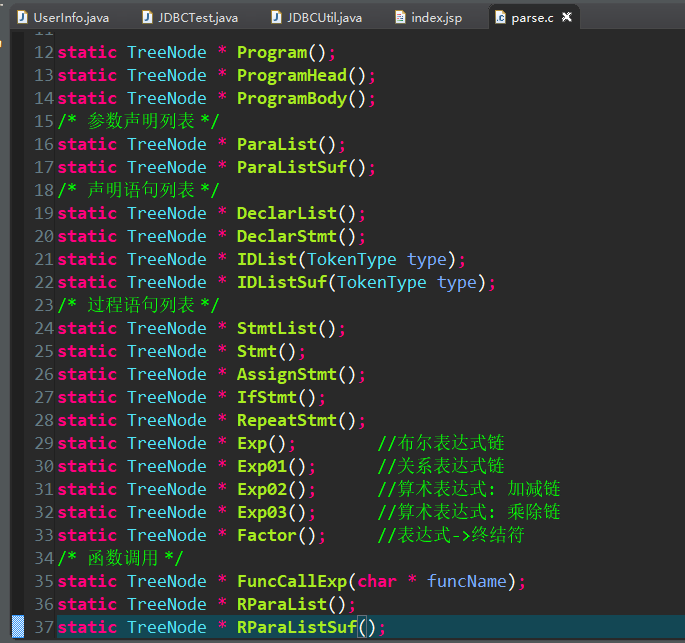
**函数接口**

* Parse.h:



* Parse.c:



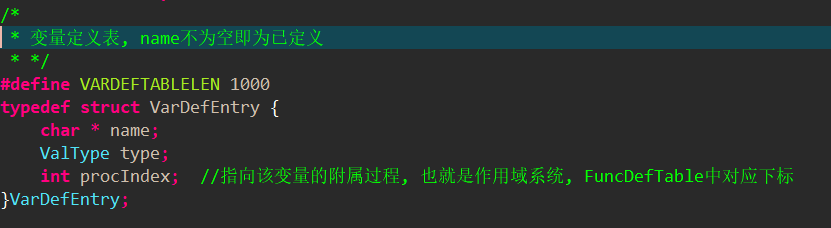


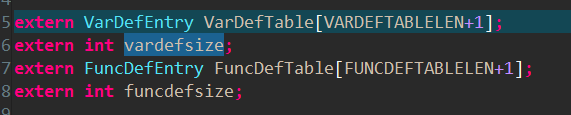
**四、符号表**

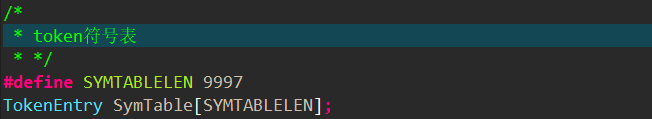
**简介**

包含一个变量定义表、一个函数定义表、一个符号统计表。

**数据结构 定长数组实现**

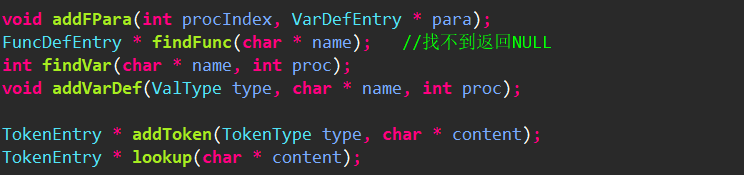




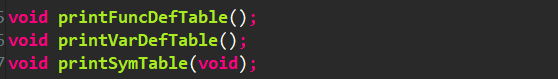


**符号表接口**

* 添加定义



* 符号表打印



* 语法树打印



**五．测试结果：**

**测试内容**

* 词法token打印
* 语法分析树打印
* 符号表打印

**测试实例**

* Test.txt

**测试结果**

* ScanRes.txt