对block、proc、statement、condition、expression、term、factor进行分析，画出语法分析图，在此基础上描述这些子程序的设计思想；

**[ ] 1次或0次**

**{} 0~n次**

**程序<prog> → program <id>；<block>**

**程序块<block> → [<condecl>][<vardecl>][<proc>]<body>**

**/常量说明<condecl> → const <const>{,<const>};**

**/常量<const> → <id>:=<integer>**

**/变量说明 <vardecl> → var <id>{,<id>};**

**分程序<proc> → procedure <id>（[<id>{,<id>]}）;<block>{;<proc>}**

**复合语句<body> → begin <statement>{;<statement>}end**

**语句<statement> → <id> := <exp>**

**|if <lexp> then <statement>[else <statement>]**

**|while <lexp> do <statement>**

**|call <id>（[<exp>{,<exp>}]） call 标识符( 参数)**

**|<body>**

**|read (<id>{，<id>}) ( )**

**|write (<exp>{,<exp>}) ( )**

**条件<lexp> → <exp> <lop> <exp>|odd <exp> 表达式 关系 表达式**

**/表达式<exp> → [+|-]<term>{<aop><term>} 表达式-》项+|-项**

**/项<term> → <factor>{<mop><factor>} 项 -》因子 \*|/因子**

**/因子<factor>→<id> |<integer>|(<exp>) 变量|常数|表达式**

**关系运算符<lop> → =|<>|<|<=|>|>=**

**加法运算符<aop> → +|-**

**乘法运算符<mop> → \*|/**

**变量标识符<id> → *l*{*l*|d} （注：*l*表示字母）**

**数字<integer> → d{d}**

1. 标识符未声明 查表：遇到标识符
2. 标识符号重复声明或者声明冲突 ：var或者const进入 查表
3. 标识符使用关键字 声明时查：关键字表

1. 缺失关键字
2. 缺失；
3. 缺失（ ）
4. 缺失， 声明变量 var or const
5. 左右括号未匹配
6. := 为 =
7. Read 的
8. Write
9. Begin 和end匹配
10. While 和 do 匹配
11. If 和 then 匹配