# 文法解读

|  |  |  |
| --- | --- | --- |
| 规则 | 分析 | 范例 |
| 1.<程序>::=<分程序>. | 程序主体是分程序，“.”相当于结束符号. | begin  end. |
| 2.<分程序>::=[<常量说明部分>][<变量说明部分>][<过程说明部分>]<复合语句> | 分程序严格按照顺序由常量说明部分，变量说明部分，过程说明部分，以及复合语句组成。  常量说明，变量说明，过程说明部分是【可有可无的】，复合语句是【必须存在的】 | const a=1,b=2;  var x,y;  begin  x:=a;  y:=b  end |
| 3.<常量说明部分> ::= const<常量定义>{,<常量定义>}; | const a=1,b=2,c=3;这种格式是正确的  const a=1; const b=2;这种是错误的 | const a=1,b=2; |
| 4.<常量定义>::=<标识符>=<无符号整数> | 常量定义的时候必须带<无符号整数>,并且用的是“=”而不是“:=” | a=1 |
| 5.<无符号整数>::=<非零数字>{<数字>} | 这里的无符号整数不允许0开头，并且是不包含0的。  我觉得这里的文法改动略微有些不妥，虽然改动文法有助于锻炼同学们的能力，但是把{1,2,3,4......}作为无符号整数的定义与常识相悖，一些无助于知识技能掌握的逻辑问题可能会阻碍同学们的学习进程，我建议以后学弟学妹的教学这里可以改成：  <无符号整数>::=<非零数字>{<数字>}|0 | 正确：1,2,10,876  错误：0,000003,0010 |
| 6.<标识符>::=<字母>{<字母>|<数字>} | 标识符必须是是字母打头，后面可以跟数字或字母，按照附加说明，标识符不区分大小写字母 | 正确：a,b,a1,p5p  错误：3c,25,5b2 |
| 7.<变量说明部分>::=var<标识符>{, <标识符>}; | 与常量说明类似  var x,y,z;这种格式是正确的  vart x; var y;这种是错误的  并且只能跟标识符，不能给标识符赋值。 | var x,y,z; |
| 8.<过程说明部分>::=<过程首部><分程序>{;<过程说明部分>}; | 过程说明部分右递归 | procedure test;  begin  end; |
| 9.<过程首部>::=procedure<标识符>; | 与常量、变量说明类似，procedure后跟一个标识符和“;” | procedure test; |
| 10.<语句>::=<赋值语句>|<条件语句>|<当循环语句>|<过程调用语句>|<复合语句>|<读语句>|<写语句>|<空> | 语句包括赋值语句、条件语句、当循环语句、过程调用语句、复合语句、读语句、写语句、空语句 | x:=1;  if x>2 then  x:=3  else  x:=4;  while x>1 do  call dosth;  begin  read(x);  write(2)  end;  ; |
| 11.<赋值语句>::=<标识符> := <表达式> | 注意赋值语句本身是没有“;”的，另一点由于无符号整数规则的缘故x:=0也是不符合规则的 | x:=1  x:=1-1 |
| 12.<表达式>::= [+|-]<项>{<加法运算符><项>} | 项的前面只能带一个加法运算符，形如++x,+-y是不符合规则的 | -2+3\*5 |
| 13.<项>::=<因子>{<乘法运算符><因子>} | 项由因子和乘法运算符组成 | 2\*3/4 |
| 14.<因子>::=<标识符>|<无符号整数>| '('<表达式>')' | 因子可以是标识符，也可以是无符号整数，也可是括号括起来的表达式。 | x  2  (1+3) |
| 15.<加法运算符>::=+|- | 加法运算符包括+、- |  |
| 16.<乘法运算符>::=\*|/ | 乘法运算符包括\*、/ |  |
| 17.<条件>::=<表达式><关系运算符><表达式>|odd<表达式> | 条件分为两种，一种是二元的<表达式><关系运算符><表达式>，一种是一元的odd<表达式>，odd是判奇数 | x>3  odd 7 |
| 18.<关系运算符>::==|<>|<|<=|>|>= | 关系运算符包括等于、不等于、小于、小于等于、大于、大于等于 |  |
| 19.<条件语句>::=if<条件>then<语句>[else<语句>] | 条件语句 if 条件 后面必须跟 then 语句，else 语句可有可无，并且then 语句后面不跟“;” | if x>2 then  x:=3  else  x:=4 |
| 20.<当循环语句>::=while<条件>do<语句> | 当循环语句 while 条件后面必须跟 do 语句 | while(x>1) do  write(1) |
| 21.<过程调用语句>::=call<标识符> | 过程调用语句以 call打头 后面跟标识符 | call dosth |
| 22.<复合语句>::=begin<语句>{;<语句>}end | 复合语句以begin为开头，end为结尾，中间语句以“;”作分割，但是最后一条语句和end之间没有“;” | begin  read(x);  write(2)  end |
| 23.<读语句>::=read '('<标识符>{, <标识符>}')' | 读语句以read开头，括号里面跟标识符，以“,”分割。 | read(x,y,z) |
| 24.<写语句>::=write '('<表达式>{, <表达式>}')' | 与读语句类似，以write开头，括号里面跟表达式，以“,”分割 | write(x,x+y+z) |
| 25.<字母>::=a|b|c|d…x|y|z|A|B...|Z | 字母包括a~z和A~Z，根据附加说明，标识符不区分大小写字母。 |  |
| 26.<数字>::=0|<非零数字> | 数字包括0、1、2、3、4、5、6、7、8、9 |  |
| 27.<非零数字>::=1|2|3…8|9 | 非零数字包括1、2、3、4、5、6、7、8、9 |  |

附加说明：标识符不区分大小写字母