

# 总结 Oberon-0 语言的词汇表

操作符	+   -   *   div   mod   :=   >   <   #   =   >=   <=   &   or   ~   .   [   ]   (   )   :
关键字	If   then   elesif   while   do   begin   end   of   ineger   write   read   writeln   record   array   module   const   type   var   procedure
注释	“( * {除了*} ) * ”
分割	;   ,
标识符	letter(letter   digit)*
数值	[1-9](0-9)*   0(0-7)*

## 单词分类的理由

分类最主要的依据就是单词的功能，根据单词不同属性、实现功能的相似区分为这六类，比如标示符，常量数值，预定义单词、注释符和分割符等明显可以根据本身的属性来区分。

# Oberon-0 语言的词法规则

## 正则定义式

- Operator

->

“+” | “-” | “\*” | “div” | “mod” |  
“:=” | “>” | “<” | “#” | “=” | “>=” | “<=” |  
“&” | “or” | “~” | “.” | “[” | “]” | “(” | “)” | “:”
- Keyword

->

If | then | elesif | while | do | begin | end |  
of | ineger | write | read | writeln | record |  
array | module | const | type | var | procedure
- Comment

->

“( \* ( ) \* ” ” ) ”
- Separation

->

“;” | “,”
- Identifier

->

[a-zA-Z]([a-zA-Z|0-9])\*

Number       ->   Decimal | Octal

Decimal       ->   [1-9](0-9)\*

Octal          ->   0(0-7)\*

## **Oberon-0 与其他高级语言的词法规则的异同比较**

### 1. 标识符和常量

Oberon 与其他语言基本相同，但 Oberon 功能较少，只有没有十六进制

### 2. 注释

C、java 采用的是//与/\*\*/来注释，而 Oberon 则用(\*\*)

### 3. 语句

Oberon 的 If 和 WHILE 是通过 BEGIN、THEN、END 等标识来约束范围，而 java、C 则是用花括号{}，Oberon 声明变量是用到了 Var，其他语言没有。

### 4. 操作符

Oberon 的许多操作符和 C、java 没有差别，例如+ - \*，但有一些也有不同包括除法“DIV”，取模“MOD”等等