

# Oberon-0 词汇表

类型	实例
Identifier	letter(letter digit)*
number	digit+   0(digit)+
保留字	if else elsif then end array var of while do record begin module procedure const mod div or boolean
关键字	write writeln read integer
操作符	+ - * ; , ( ) [ ] . : = # < > >= :=

# Oberon 词法规则

Alphanum	-> [:letter:]   [:digit:]
Identifier	-> [:letter:]{Alphanum}*
Integer	-> [:digit:]+
OctalInteger	-> 0[:digit:]+
WhiteSpace	-> [ \t\n\r]*
Comment	-> "(*" ~"*)"
Operator	-> +   -   *   ;   ,   (   )   [   ]   .   :   =   #   <   <=   >   >=   :=

# Oberon-0与其他语言词法比较

- Oberon-0是大小写无关的，而C/C++等语言是大小写有关
- Oberon-0的数字类型只有整型，而C等语言还有浮点类型
- Oberon-0的支持的整型数字进制只有两种，分别是十进制和八进制。C等语言有三种分别是十进制，八进制和十六进制
- C等语言的操作符都是特殊符号，而Oberon-0部分操作符是单词（符合标志符规则）。例如C语言中取模运算符是 `%`，在Oberon-0中是 `MOD`。

# 三个词法生成器关于词法规则定义的比较

- 三个生成器都有 `start condition`，配合上用户自定义的变量可以实现更加复杂的词法识别（比普通的正则定义式更强大）。例如在词法分析阶段可以支持嵌套注释。
- 当一串字符同时满足多个词法规则时，三个词法生成器的行为相同。都是采用最长匹配原则，如果最长匹配不足以解决冲突，那么就在候选规则中，匹配为第一个定义出来的规则。