子实验一：熟悉Oberon-0语言定义

* 1. **讨论Oberon-0语言的特点**

1. 根据Oberon-0语言的BNF定义，Oberon-0程序中的表达式语法规则与Java、C/C++等常见语言的表达式有何不同之处？试简要写出他们的差别。
2. 在Oberon-0中一个过程采用BEGIN和END来界定一个过程的开始和结束，在Java、C/C++中采用的是花括号来界定一个过程的范围；
3. 在Oberon-0中可以在一个方法中定义一个方法，即可以进行方法的嵌套定义，而在Java、C/C++中不可以进行方法的嵌套定义；
4. 在Oberon-0中变量类型是写在变量名之后，而在Java、C/C++中变量类型写在变量名之前；
5. 在Oberon-0中WHILE语句和IF语句也是采用END来进行范围的界定，表示WHILE语句和IF语句的结束，而在Java、C/C++中采用花括号来进行范围界定。
6. 在Oberon-0中的数据类型比较少，只有INTEGER和BOOLEAN并且布尔值不能适用true和false，Oberon-0中声明数组的形式也和Java、C/C++不同。
7. 在Oberon-0中不支持FOR循环和DO-WHILE循环；
   1. **讨论Oberon-0文法定义的二义性**

根据Oberon-0语言的BNF定义，讨论 Oberon-0 程序的二义性问题，即讨论根据上述BNF定义的上下文无关文法是否存在二义性。

答：Oberon-0语言中不存在二义性。在其他高级语言程序设计语言中常见的二义性主要有：

1. 算术表达式中运算符号的优先级问题；

在Oberon-0中算数表达式运算符优先级的问题，采用了在文法中使用层次分层的方法进行二义性的消除，具体方法如下：

expression = simple\_expression

[(“=”|“#”|“<”|“<=”|“>”|“>=”)

simple\_expression];

simple\_expression = [“+”|“-”]term{(“+”|“-”|“OR”)term};

term = factor{(“\*”|“DIV”|“MOD”|“&”)factor};

1. IF语句中的多个if和else的比配问题。

在Oberon中的IF语句，采用END标识符表示一个IF语句的结束，故不存在多个if语句和else语句的匹配上由二义性的问题。采用的方法如下所示：

“IF” expression “THEN”

statement\_sequence

{“ELSIF” expression “THEN”

statement\_sequence }

[“ELSE”

statement\_sequence ]

“END”;