子实验一：熟悉 Oberon-0 语言定义

1.1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。

① 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；
② 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；
③ 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；
④ 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。
⑤ 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。
⑥ 在 Oberon-0 中不支持 FOR 循环和 DO-WHILE 循环；

1.2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};
```

（2） 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”;
```