Oberon-0 的语言特点

1） 在定义变量方面，Oberon-0 需要在 PROCEDURE 过程前统一定义完成，不能在中间夹杂变量的定义。之前版本的 C 语言也是需要在方法前面统一定义好变量，现在当前版本的 C 或 java 都可以中途添加变量。

2） 在定义变量的时候，Oberon-0 会先用 Var 来声明后面的标识符代表变量，在标识符后面给出变量的类型，而 C 或 java 一般在定义时先给出类型然后后面接着变量名称。

3） Oberon-0 类型选择较少，只有 interger 和 boolean。

4） Oberon-0 可用的数据结构主要有数组，而 C 有结构体、java 有类。

5） Oberon-0 相应的语句比较少，仅有赋值、方法调用、if 语句、while 语句四种，而 C、java 除此之外还有 for 语句，do-while 语句等等。

Oberon-0 二义性讨论

我认为该文法没有二义性。在其他的高级程序语言中，比较常见的二义性问题是在类似类继承的时候若父类有多个且有相同名称的方法，子类调用就不知道调用哪个，Oberon-0 语言比较微型没有这类二义性问题，此外关于会出现多个语法树的问题以及类似 if-else 匹配问题也被在，文法的 EBNF 定义中给确实解决了。

比如 if-else 问题 C、java 都是采用规则限定，使用了就近匹配的原则来解决该问题，Oberon-0 对于该问题则是采用 end 标识，每个 if 语句都需要 end 来结束，这样就能解决 if-else 匹配问题。

同样，多个语法树这样的问题，主要是出现在运算的时候，Oberon-0 的 EBNF 通过将表达式的加减和乘除等分在不同的层次，体现出差别，来消除这样的二义性。

心得体会

第一个子实验的内容还是比较简单的，看完 Oberon-0 语言的文法之后，在对照着例子写出一个正确的 Oberon 程序就不算太难。而且 Oberon-0 也没有什么复杂的结构、语句，学习起来也方便。