过程部初步

———最基本的过程部语句

过程部是 COBOL 程序的核心部分，他决定计算机应做什么操作。因此，学习 COBOL 时，不应把四个部分并列的、孤立的进行学习。而应当一过程部为主体，找出几个部分之间的有机联系。

过程部的几个特点：

（1） 过程部是程序中的第四部分，它以部头 PROCEDURE DIVISION 开头。部头从 A 区开始书写。

例如：
```
PROCEDURE DIVISION.
A SECTION.
A1. MOVE 1 TO I.
    MOVE 2 TO J.
A2. ADD I TO J.
B SECTION.
……
```

（2） 过程部的语句都以一个动词开始，如 MOVE,OPEN,READ,WRITE 等。它表示计算机应执行的操作。

（3） 语句中的动词后面一般要跟以一个操作的对象。操作对象可以是数据名或文件名。

（4） 过程部的语句一律从 B 区开始书写。一个语句可以在一行或多行上。

一、输入输出语句

1. 接收语句(ACCEPT 语句)

ACCEPT 语句的一般形式

ACCEPT 标识符 [FROM 助忆名]

(1) 这里的标识符指的是能唯一的标识一个数据项的数据名。

(2) ACCEPT 后面只能跟一个标识符，不能出现两个或两个以上标识符，但可以用组合项。

2. 显示语句(DISPLAY 语句)

DISPLAY 的一般形式

DISPLAY { 标识符 1 / 常量 1 }, [标识符 2 / 常量 2] ……[UPON 助忆名]

(1) 如果没有 UPON 可选项，则在计算机隐含指定的输出设备上显示数据。

(2) 每执行一个 DISPLAY 语句，总是从一个新行开始显示的。

(3) 在运行正式程序时，一般不用 ACCEPT 和 DISPLAY，以提高计算机效率，减少程序员的干预。

3. 读语句(READ 语句)

READ 的一般形式

READ 文件名 RECORD [INTO 标识符] [; AT END 执行语句]

（1） 在 READ 语句中操作的对象是文件。每执行一次 READ 语句，就从指定文件中读入一

条记录。

（2） 在计算机内存区中专门开辟一片存储单元（输入记录区）来存放从文件读入的信息。

（3） 文件读完时的处理。这是由 READ 语句中的“ AT END”子句来实现的。

例： READ IN-FILE AT END STOP RUN.

此语句的作用是：读入 IN-FILE 文件，如果该文件已读完而遇到文件结束标志，则执行“ AT END ”子句中的语句 STOP RUN,程序结束运行。

（4） 可以用一个 READ 语句读入一条记录，并马上将记录区的内容转送到另一数据项中去。

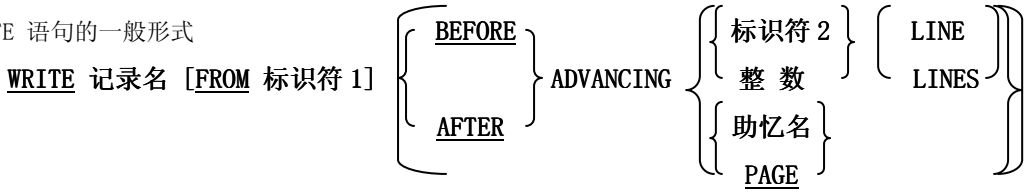
例： READ IN-FILE INTO A AT END STOP RUN.

这个语句包含两个作用：①读入 IN-FILE 文件的一个记录到输入记录区。

②将该记录区的内容传送到数据项 T 中去。

4. 写语句（WRITE 语句）

WRITE 语句的一般形式



（1） 将内存区中的内容输出到外部设备。

（2） 输出设备的选择。在环境部中确定所用的输出设备。

（3） 定义输出记录区。

（4） 在用 WRITE 语句输出一个输出记录之前应当向该记录区传送数据。

（5） 走纸的控制。

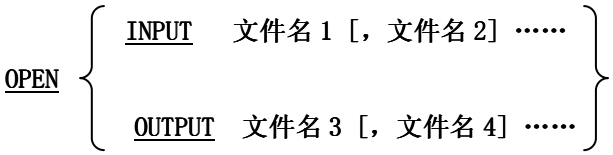
（6） 可以用一个 WRITE 语句先把内存中另一数据项的内容传给输出的记录区然后再输出。

例：WRITE OUTPUT-RECORD FROM T1 AFTER 3.

(7) 当不出现 BEFORE 或 AFTER 时，大多数系统按等价于 AFTER 1 处理。

5. 打开语句(OPEN 语句)

OPEN 语句的一般形式



（1） 程序如果需要读文件或写文件，则该文件必须先用 OPEN 语句打开。

（2） 一个 OPEN 语句可以打开一个或多个文件

6. 关闭语句(CLOSE 语句)

CLOSE 语句的一般形式

CLOSE 文件名 1 [，文件名 2]

（1） 当对一个文件的读或写操作已完成，就应关闭这个不再使用的文件，使它不再涉入以后的数据操作之中。

（2） CLOSE 与 OPEN 用法不同，只需指出文件名即可。

（3） 在程序中 CLOSE 与 OPEN 要相互对应。

（4） 文件关闭后就不能再对文件在进行读写操作，如有需要，可再打开。

二、算术运算语句

1. 加法语句(ADD 语句)

加法语句的几种不同形式：

(1) ADD A TO B. 表示 A + B => B

(2) ADD A , B TO C 表示 A + B + C => C

(3) ADD A , B GIVING C 表示 A + B => C

(4) ADD A , B TO C , D 表示 A + B + C => C , A + B + D => D

使用加法语句的几点注意：

(1) 在 TO 和 GIVING 后面只能跟数据名，而不能跟常量。

(2) TO 前后的数据名的次序不要随便改换。

(3) GIVING 的后面可以跟几个数据名。

(4) 参加运算的只能是数值量，它们的长度不应超过 18 位数字。

2. 减法语句(SUBTRACT 语句)

减法语句的几种不同形式：

(1) SUBTRACT B FROM A. 表示 A - B => A

(2) SUBTRACT B , C FROM A. 表示 A - B - C => A

(3) SUBTRACT B , C FROM A , T. 表示 A - B - C => A T - B - C => T

(4) SUBTRACT B , C FROM A GIVING T. 表示 A - B - C => T

使用减法语句的几点注意：

(1) GIVING 后面不能跟常量。

(2) 如不带 GIVING 部分，则 FROM 后面也不能跟常量。

3. 乘法语句(MULTIPLY 语句)

乘法语句的几种不同形式：

(1) MULTIPLY A BY B. 表示 A * B => B

(2) MULTIPLY A BY B GIVING C. 表示 A * B => C

(3) MULTIPLY A BY B , C. 表示 A * B => B A * C => C

使用乘法语句的几点注意：

(1) 当不带 GIVING 部分时，BY 后面不能跟常量。

(2) 带 GIVING 部分时，BY 后面可以是常数，而 GIVING 后面不能是常量。

(3) 总之，存放值的项只能是数据名，不能是常量。

4. 除法语句(DIVIDE 语句)

除法语句的几种不同形式：

(1) DIVIDE A INTO B. 表示 B / A => B

(2) DIVIDE A INTO B GIVING C.　　表示 B / A => C

(3) DIVIDE A BY B GIVING C.　　表示 A / B => C

使用除法语句的几点注意：

(1) 当用 GIVING 部分时，第一个运算量可以除第二个运算量，此时用 INTO。

(2) 允许 GIVING 后有几个标识符。

(3) 如除不尽则多余的位数被截去。

5．四种算术运算的小结

(1) 一个语句只能进行一种单一的运算，不能在一个语句中实现两种不同的运算。

(2) 加法和减法可以进行两个以上数值量的计算。

(3) 四种算术运算，都有两种形式，即带 GIVING 和不带 GIVING 的。

6．计算语句(COMPUTE 语句)

(1) 算术表达式：算术表达式是由算术初等量（数值常量、数值型数据项），算术运算符(+ 、-等)，括号，所组成的有意义的式子。

(2) 运算次序：①括号

②单目运算符（正负号）　　高

③乘方

④乘，除

⑤加，减　　低

同级运算符按自左向右次序。

计算语句的一般表达式：

COMPUTE 标识符 1 [， 标识符 2] …… = 算术表达式

书写表达式时注意：(1) 所有运算符两侧均应留一空格。

(2) 括号的外侧应留空格，内侧不要留空格。

三、传送语句(MOVE 语句)

MOVE 语句的一般形式

MOVE { **标识符 1** / **常量 1** } **TO 标识符 2 [， 标识符 3] ……**

1．传送语句的作用：

MOVE 语句用来实现数据的传送，将一个数据从一个内存域送到另一个内存域中。可以将常量（包括数值常量、非数值常量、表意常量）或一数据项的内容传送给另一数据项。

2．传送原则

MOVE A TO B　　其中 A 称为发送项，B 称为接收项。

(1) 如果接收项和发送项在数据部中描述的类型和长度相同，则按字节一一对应的传送。

(2) 如果接收项和发送项长度不相同，而两者都是数值数据项，则按“小数点对齐”原则处理。如

果是整数，则认为小数点在最后一位数字之后。如果接收项长度大于发送项，则多余位补零。如果接收项长度小于发送项，则产生截断。

(3) 对字母或字符数据（非数值型数据）的传送，按“左对齐”原则处理。如果接收项长度大于发送项长度，则多余位补空格。如果接收项长度小于发送项的长度，则从右端截断。

(4) MOVE 语句可以将一初等项内容传送给另一初等数据项，也可以将一组合项内容传送给一初等项，也可以将一初等项内容传送给一组合项。

四、转移语句(GOTO 语句)

有时需要使程序改变正常执行的顺序，这时可以使用 GOTO 语句。这是一个无条件转移语句

转移语句的一般格式

GO TO 过程名 1 [， 过程名 2] …… 过程名 n DEPENDING ON 标识符

五、条件语句(IF 语句)

关系运算符：

COBOL 关系运算符	意义	相当数学上的符号
IS GREATER THAN	大于	>
IS LESS THAN	小于	<
IS EQUAL TO	等于	=
NOT GREATER THAN	不大于	≦
NOT LESS THAN	不小于	≧
NOT EQUAL TO	不等于	≠

1．关系运算规则

(1) 数值量之间的比较，按他们的代数值进行比较（即按有效数和代数符号）。

(2) 字母型数据的比较按英文字典顺序，在英文字典中位置在后的字比位置在前的字大。

2．IF 语句的两种形式

(1) IF 条件 语句组

(2) IF 条件 语句组 1 ELSE 语句组 2

六、停止语句(STOP 语句)

STOP 语句的一般形式

STOP { RUN / 常量 }

说明：(1) 当用 STOP RUN 时，执行此语句将程序停止运行，停止后不能再接着向下运行。

(2) 用 STOP 常量 ， 表示程序暂时挂起不往下执行，显示出此常量（可以是数值常量或非数值常量或表意常量）