

过程部之三

——执行语句(PERFORM 语句)

一、执行语句的作用

在一个 COBOL 程序中过程部中往往有一部分语句是需要多次执行的。这时需要使用执行语句。

二、执行语句最基本的形式 (SAMPLE6-1)

PERFORM 语句的最基本的形式：

PERFORM 过程名

过程部中的节名、段名称为过程名。用 PERFORM 语句只能转到指定的节或段的开头，执行完该节（段）的全部语句后返回。

有时，需要多次执行的不是一个段（节），而是若干个端（节），则应指明从哪一段（节）起到哪一段（节）止。则 PERFORM 语句的一般格式可扩充为

$$\underline{\text{PERFORM}} \text{ 过程名 1 } \left[\left\{ \begin{array}{c} \underline{\text{THROUGH}} \\ \underline{\text{THRU}} \end{array} \right\} \text{ 过程名 2 } \right]$$

三、执行语句的使用规则

1. PERFORM 语句的嵌套

嵌套不能交叉，后一个被“调用”的语句序列或者全部套在前一个被“调用”的语句序列之中，或者全部在它之外是正确的。不允许二者交叉或者有共同的终点

2. 在 PERFORM 语句中所执行的语句序列中，可以含有转移语句，可以使流程转到语句序列之外，但一般应转回到此语句序列，以便最后能执行此语句序列的最后一个句子。

例： T1. PERFORM T3 THRU T5

T2. ...

T3. ...

T4. GO TO T6

T5. ...

T6. ...

T7. GO TO T5

执行顺序：T1 => T3 => T4 => T6 => T7 => T5 => T2 => T3 => T4 => T6 => T7
=> T5 => T6 => T7 => T5

3. 使用 PERFORM 语句实现循环

这种形式的 PERFORM 语句的一般形式为

$$\underline{\text{PERFORM}} \text{ 过程名 1 } \left[\left\{ \begin{array}{c} \underline{\text{THROUGH}} \\ \underline{\text{THRU}} \end{array} \right\} \text{ 过程名 2 } \right] \left\{ \begin{array}{c} \text{整数} \\ \text{标识符} \end{array} \right\} \underline{\text{TIMES}}$$

说明：（1）标识符应为整数数据类型。

（2）如果此标识符的值在执行语句序列中有变化，不会影响执行的次数。即以它开始时的值决定执行的次数

例：PERFORM T1 THRU T5 N TIMES.

四、执行语句的较复杂的形式 (SAMPLE6-2)

PERFORM 语句还有以下两种比较复杂的形式

1. 反复执行指定的语句序列，直到给定的条件满足为止。

PERFORM 过程名 1 $\left(\begin{array}{c} \text{THROUGH} \\ \text{THRU} \end{array} \right) \text{过程名 2}$ **UNTIL 条件**

注意：在执行 PERFORM 语句时，先判断指定的“条件”是否满足，若满足则执行，如果一开始“条件”不满足，则一次也不执行。

2. 类似于 FOR 循环的 PERFORM 语句 (SAMPLE6-3)

PERFORM 过程名 1 $\left(\begin{array}{c} \text{THROUGH} \\ \text{THRU} \end{array} \right) \text{过程名 2}$ **VARYING 标识符 1**
FROM $\left\{ \begin{array}{c} \text{常数 1} \\ \text{标识符 2} \end{array} \right\}$ **BY** $\left\{ \begin{array}{c} \text{常数 2} \\ \text{标识符 3} \end{array} \right\}$ **UNTIL 条件**

注意：循环变量初至可以为正值、负值或零，步长也可以是正值、负值但不能为零。循环变量的值在每次循环中是自动按步长增长的。在最后的出口条件中，不一定要和循环变量有关。

例：PERFORM T1 THRU T2 VARYING X FROM A BY B UNTIL X > 5

六、执行语句的多重循环形式 (SAMPLE6-4)

COBOL 允许用到三重循环，其一般形式为

PERFORM 过程名 1 $\left(\begin{array}{c} \text{THROUGH} \\ \text{THRU} \end{array} \right) \text{过程名 2}$
[VARYING 参数 1 FROM 初值 1 BY 步长 1 UNTIL 条件 1]
[AFTER 参数 2 FROM 初值 2 BY 步长 2 UNTIL 条件 2]
[AFTER 参数 3 FROM 初值 3 BY 步长 3 UNTIL 条件 3]

例：PERFORM A VARYING I FROM 1 BY 1 UNTIL I > 10

AFTER J FROM 1 BY 1 UNTIL J > 10

AFTER K FROM 1 BY 1 UNTIL K > 10

七、几种形式的执行语句的小结

前面已经学过了五种形式的执行语句，现各举一个例子，进行小结

1. PERFORM T1 THRU T2.

最简单的执行语句。

2. PERFORM T1 THRU T2 5 TIMES.

已知执行次数的执行语句。

3. PERFORM T1 THRU T2 UNTIL N > 10.

带判断条件的执行语句。

4. PERFORM T1 THRU T2 VARYING X FROM 1 BY 1 UNTIL X > 5.

循环变量自动变化，直到满足出口条件。

5. PERFORM T1 THRU T2

VARYING I FROM 1 BY 1 UNTIL I > 5

AFTER J FROM 1 BY 1 UNTIL J > 5

使用执行语句实现二重循环。

PERFORM 的作用：（1）“调用” 本程序中某一语句序列

（2）实现循环

八、出口语句(EXIT 语句)

EXIT 语句提供了一组过程的公共出口，或者说它指出了被调用过程的逻辑终点。一般用来作 PERFORM 语句调用的语句序列的出口。

注意：EXIT 必须是本段中唯一的语句，即该段中不允许有其他语句。EXIT 语句前必须有一段名。

九、修改语句(ALTER 语句)

修改语句的一般格式

ALTER 过程名 1 **TO** [**PROCEED TO**] 过程名 2

[, 过程名 3 **TO** [**PROCEED TO**] 过程名 4]

ALTER 语句用来改变 GO TO 的转向点。该语句使以过程名 1，过程名 3，……命名的各段中的 GO TO 语句的转向点分别被修改为过程名 2，过程名 4，……各段只能由一条 GO TO 语句单独组成。

例： ALTER PROC1 TO PROC5.