

原创文章，转载请注明出处。

更多实用资料请登录方正智芯官网：www.founderchip.com

作者：北岛李工

前面的文章我们介绍了西门子SCL编程的变量和表达式，今天这篇文章，我们来了解下SCL的语句与结构。



语句在计算机科学中被称为"Statement"。它是一条能被执行的代码，其作用是向计算机/PLC系统发出操作指令，要求执行相应的操作。语句经过编译后会产生若干条机器指令。

在很多高级语言中，代码必须提供某种符号来表示一条语句，以便编译器能识别并编译。比如，C语言的每一条语句末尾都要加英文分号 (;)，而VB则以回车换行符来表示一条语句；西门子SCL的语法源自PASCAL，其每条语句的末尾也要加英文分号 (;)。

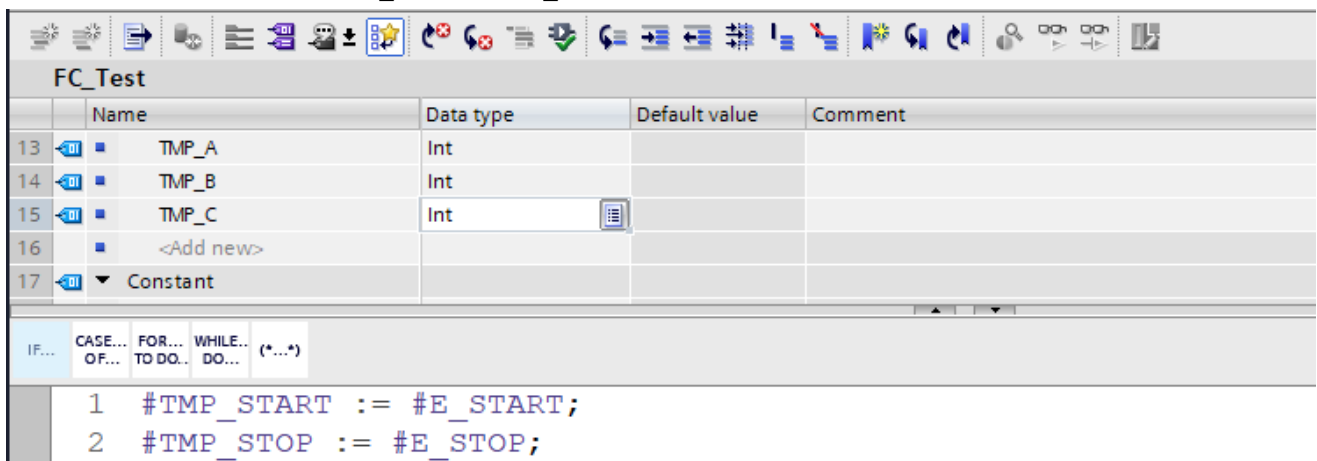
语句有简单与复杂之分。

1-简单语句

简单语句用来组成顺序结构，即自上而下的顺序代码。

赋值语句属于简单语句。

例如下面的代码，将形参E_START和E_STOP赋值给临时变量。



图片中的": ="符号表示赋值;

赋值语句的右边也可以是表达式，例如下面的代码：

```
4 #TMP_B := #TMP_A * 3;
5 #TMP_C := #TMP_A + #TMP_B;
6
```

第一条语句将TMP_A乘以3，然后赋值给TMP_B；

第二条语句将TMP_A与TMP_B相加，然后赋值给TMP_C；

2-复杂语句

复杂语句可以构成选择结构或循环结构。

1、选择结构

选择结构用来根据某些条件来选择性的执行代码。

选择结构包括IF语句和CASE语句。

1.1 IF语句

IF语句用来判断某种条件是否满足。如果满足的话，则执行其内部的代码。

比如下面的代码：

```
7 IF #E_CODE = 1 THEN
8     // Statement section IF
9     #TMP_A := 1;
10    #TMP_B := 2;
11 END_IF;
12
```

如果E_CODE的值为1，则将TMP_A赋值为1；TMP_B赋值为2；

上述代码中，如果E_CODE的值不等于1，则不执行其内部的语句；程序会跳转到END_IF之后的语句继续执行；

如果希望在E_CODE的值不等于1的情况下，执行某些语句，可以使用ELSE关键词，如下图：

```
7 IF #E_CODE = 1 THEN
8     // Statement section IF
9     #TMP_A := 1;
10    #TMP_B := 2;
11 ELSE
12    #TMP_A := 100;
13    #TMP_B := 200;
14 END_IF;
15
```

上图中，如果E_CODE的值等于1，则将TMP_A赋值为1，TMP_B赋值为2；否则的话，将TMP_A赋值为100，TMP_B赋值为200；

IF语句的内部还可以嵌套IF语句，比如下面的代码：

```

7 IF #E_CODE = 1 THEN
8     // Statement section IF
9     #TMP_A := 1;
10    #TMP_B := 2;
11    //another IF statement
12 IF #E_CONDITION=100 THEN
13     // Statement section IF
14     #TMP_C:=100;
15 END_IF;
16
17 ELSE
18     #TMP_A := 100;
19     #TMP_B := 200;
20 END_IF;

```

如果E_CODE的值等于1，则将TMP_A赋值为1，TMP_B赋值为2；在这个条件下，如果E_CONDITION的值等于100，则将TMP_C赋值为100；

如果要判断的分枝比较多，比如，E_CODE的值等于1、2、3、4等要执行不同的代码，则可以使用CASE语句；

1.2 CASE语句

CASE语句用来完成多分枝的选择判断，比如下面的代码：

```

22 CASE #E_CODE OF
23     1: // Statement section case 1
24         #TMP_A := 1;
25         #TMP_B := 2;
26     2..4: // Statement section case 2 to 4
27         #TMP_A:=10;
28         #TMP_B := 20;
29     5:
30         #TMP_A := 100;
31         #TMP_B := 200;
32     ELSE // Statement section ELSE
33         #TMP_A:=30;
34         #TMP_B := 40;
35 END_CASE;
36

```

当E_CODE的值等于1时，将TMP_A赋值为1，TMP_B赋值为2；

当E_CODE的值等于2或者3或者4时，将TMP_A赋值为10，TMP_B赋值为20；

当E_CODE的值等于5时，将TMP_A赋值为100，TMP_B赋值为200；

其它情况下，将TMP_A赋值为30，TMP_B赋值为40；

2、循环结构

循环结构可以在某种条件下反复执行某段代码，包括FOR语句、WHILE语句和REPEAT语句。

2.1 FOR语句

FOR语句用于以次数确定的方式来执行某段代码。比如下面的图片：

```
37 □ FOR #TMP_COUNTER := 1 TO 1000 DO
38     // www.founderchip.com
39     // increase TMP_C 1000 times
40     #TMP_C:=#TMP_C+1;
41 END_FOR;
42
```

该代码使用TMP_COUNTER作为计数变量，其起始值被赋值为1，每执行一次语句TMP_C:=TMP_C+1，TMP_COUNTER的值自动加1，直到其值大于1000跳出循环；在FOR语句中，可以通过关键词BY修改计数器的步值，即计数器每次自动增加多少。例如下面的代码：

```
43 □ FOR #TMP_COUNTER := 1 TO 1000 BY 2 DO
44     // Statement section FOR
45     // www.founderchip.com
46     // increase TMP_C 1000 times
47     #TMP_C := #TMP_C + 1;
48 END_FOR;
```

该代码将TMP_COUNTER的起始值赋值为1，每执行一次语句TMP_C:=TMP_C+1，TMP_COUNTER的值自动加2，直到其值大于1000跳出循环；步值可以为负数，表示每执行一次循环，循环计数器减去相应的数值。

例如下面的代码：

```
50 □ FOR #TMP_COUNTER := 1000 TO 1 BY -2 DO
51     // Statement section FOR
52     // www.founderchip.com
53     #TMP_C := #TMP_C + 1;
54 END_FOR;
55
```

循环计数器TMP_COUNTER的值从1000开始，每执行一次循环，其值减2；直到小于1跳出循环。

2.2 WHILE语句

WHILE语句适用于次数不确定的循环。

比如下面的代码：

```
56 □ WHILE #E_CONDITION=1 DO
57     // Statement section WHILE
58     #TMP_C := #TMP_C + 1;
59 END_WHILE;
60
```

只要E_CONDITION的值等于1，将无限次循环执行句：TMP_C:=TMP_C+1；如果我们真的将这段代码下载到CPU中执行，可能会产生很糟糕的结果。因为一旦E_CONDITION=1的条件始终都被满足，CPU将陷入死循环。为了避免进入死循环，要确保E_CONDITION在某些情况下不会等于1。

另外，为了在某种情况下能退出循环，可以使用EXIT语句；

EXIT语句可以立即退出当前的循环，比如下面的代码：

```
56 WHILE #E_CONDITION=1 DO
57     // Statement section WHILE
58     // www.founderchip.com
59     #TMP_C := #TMP_C + 1;
60     IF #TMP_C=32767 THEN
61         // Statement section IF
62         EXIT;
63     END_IF;
64 END_WHILE;
65
```

当TMP_C达到最大值32767时，退出WHILE循环。

2.3 REPEAT语句

REPEAT 语句用来重复执行某段代码直到满足某种条件退出循环。

例如下面的代码：

```
66 REPEAT
67     // Statement section REPEAT
68     // www.founderchip.com
69     #TMP_C := #TMP_C + 1;
70 UNTIL #TMP_C=3000 END_REPEAT;
71
```

重复执行TMP_C:=TMP_C+1直到TMP_C的值等于3000。


同样可以使用EXIT语句退出REPEAT循环。

循环语句的内部也可以再嵌套循环语句，限于篇幅就不展开描述了。

好了，西门子SCL语言的语句与结构就先介绍到这里。欢迎扫描下方二维码关注我们的微信公众号。

方正智芯
Founder Chip

[长按扫码关注我们](#)



方正智芯

公众号：founderchip

官方网站：www.founderchip.com

原创工业智能控制领域（PLC、单片机、通信）的技术分享