

Controle de Fluxo

Comandos de controle de fluxo de um programa servem para especificar a ordem em que os comandos devem ser executados. Nos capítulos anteriores foi suposta a execução linear de todos os comandos de um programa; mostraremos comandos que permitirão:

- Executar condicionalmente um grupo de comandos;
- Executar repetidamente um grupo de comandos conforme uma condição;
- Executar repetidamente um grupo de comandos um certo número de vezes;
- Abandonar um subprograma no meio da execução.

Comando SE

O comando SE executa um conjunto de comandos se o resultado de uma expressão lógica for verdadeira. O formato do comando é:

```
SE expressão lógica
.... comandos.....
.....
FIM
```

A expressão lógica pode resultar em um valor verdadeiro ou falso; se for verdadeiro, o conjunto de comandos encontrados até a palavra FIM será executado; caso contrário não. Por exemplo:

```
SE %BITOLA = 0
N NBIT = 2
N BITOLA = 8
FIM
```

A variável BITOLA foi comparada com zero. Sendo igual, os 2 comandos seguintes serão executados. Todo o comando SE termina numa palavra FIM.

A expressão lógica contém somente constantes, e por isto as variáveis envolvidas precisam sempre ser substituídas por seu valor através do sinal "%". Os tipos de expressões lógicas possíveis são:

=	Igual a
---	---------

<>	Diferente de
>	Maior que
<	Menor que
>=	Maior que ou igual a
<=	Menor que ou igual a

O sinal != pode ser usado também no lugar de <>.

Expressões lógicas e aritméticas

O programa permite a construção de expressões lógicas bem mais complexas. Para entender como isto é possível, é preciso conhecer as regras de interpretação de expressões lógicas:

A expressão lógica interpretada pelo comando SE na verdade é uma expressão aritmética, que resulta em valor numérico. Este valor será considerado falso se valer zero e verdadeiro se diferente de zero.

Os operadores lógicos funcionam como operadores aritméticos, que atuam sobre um ou mais operandos, e resultam no valor 1 quando a condição for verdadeira ou 0 se falsa. Os operadores, assim como outros operadores aritméticos são interpretados da esquerda para a direita, a menos da precedência dos parênteses.

Os operadores E, OU e NAO, podem ser usados em expressões lógicas, com os seguintes resultados:

Operação	Resultado
0 E 0	0
1 E 0	0
1 E 1	1
0 OU 0	0
1 OU 0	1
1 OU 1	1
NAO 0	1
NAO 1	0

onde 0 representa um valor falso, e 1 verdadeiro. Como exemplo, suponha que uma linha entre os pontos 1 e 2 deva

ser desenhada se A e B forem não nulos ou se C e D forem nulos:

```
SE ((%A <> 0) E (%B <> 0)) OU -  
( (%C = 0) E (%D = 0) )  
LINHA 1 2  
FIM
```

Os parênteses usados são estritamente necessários; se não fossem utilizados, as expressões passariam a ser interpretadas da esquerda para a direita, o que resultaria em um valor lógico incorreto.

Expressões lógicas com cadeias de caracteres

O comando SE interpreta apenas expressões aritméticas. Para comparar cadeias de caracteres, a função CMP retorna o valor (0) se duas cadeias de caracteres forem iguais ou (1) se forem diferentes. Por exemplo, colocaremos o texto armazenado em TITULO no desenho, a menos que seu valor seja 'INVALIDO':

```
SE CMP ('%TITULO', 'INVALIDO') <> 0  
TEXTO 0,0 '%TITULO'  
FIM
```

SENAO

O comando SE executa um conjunto de comandos a seguir se o resultado da expressão lógica for verdadeiro. A palavra SENAO colocada antes do FIM do comando SE permite que um conjunto de comandos seja executado no caso da condição ser falsa. Por exemplo:

```
SE %BITOLA = 0  
N NBIT= 2  
N BITOLA= 8  
SENAO  
N NBIT= 4  
FIM
```

Resultará em NBIT=2 e BITOLA=8 se BITOLA valer zero, ou em NBIT=4 caso contrário.

Aninhamento de SEs

O comando SE permite que um conjunto de comandos seja executado condicionalmente. Neste conjunto de comandos pode haver outros comandos SE, que poderão executar outro conjunto e assim por diante. Independente

da quantidade de SEs usados, cada um deles deverá terminar por um comando FIM. Por exemplo:

```
SE LEN ('%TITULO') > 0
SE LEN ('%TITULO') < 10
NUM HTEX 0.4
DEFINE NÍVEL 0
SENAO
NUM HTEX 0.2
DEFINE NÍVEL 1
FIM
TEXTO 100,200 HTEX %HTEX '%TITULO'
FIM
```

O primeiro SE executa um conjunto de comandos se o número de caracteres da variável alfanumérica TITULO for maior que zero. Entre os comandos executados, existe um outro SE, que define o tamanho de um texto e o nível de desenho, conforme o número de caracteres de TITULO menor ou maior que 10.

Quando o DP encontra uma palavra FIM, ela se refere sempre ao último comando SE "aberto" (ou ainda, ao ENQUANTO ou REPETE, que veremos adiante). Se não houver SE aberto, então o FIM se referirá ao fim do subprograma ou desenho, que será terminado.

Note também que na medida que os SE foram sendo abertos, fomos deslocando os comandos agrupados sob o SE para a direita, de modo a deixar claro o seu agrupamento. Esta medida tem o efeito meramente estético, mas é extremamente importante para quem lê o programa, que enxerga como os comandos estão agrupados.

SENAO SE

Uma última variante do comando SE é o SENAO SE. Suponha que necessitemos analisar uma série de hipóteses mutuamente exclusivas, como no exemplo:

```
SE %A = 1
A TIPO = 'Tijolo de maciço'
SENAO
SE %A = 2
A TIPO = 'Tijolo cerâmico'
SENAO
SE %A = 3
```

```

A TIPO = 'Argamassa'
SENAO
SE %A = 4
A TIPO = 'Azulejo'
SENAO
A TIPO = 'INVALIDO'
FIM
FIM
FIM
FIM

```

O comando SENAO SE permite analisar casos deste tipo, sem aninhamento adicional dos SEs. Quando o DP encontra um SENAO SE, o nível de aninhamento não é aumentado, e somente um FIM continua necessário. Usando este comando, o exemplo acima ficaria:

```

SE %A = 1
A TIPO = 'Tijolo de maciço'
SENAO SE %A = 2
A TIPO = 'Tijolo cerâmico'
SENAO SE %A = 3
A TIPO = 'Argamassa'
SENAO SE %A = 4
A TIPO = 'Azulejo'
SENAO
A TIPO = 'INVALIDO'
FIM

```

REPETE

Este comando repete um conjunto de comandos um número determinado de vezes. A repetição é controlada por uma variável, dita de controle, que tem o seu valor incrementado ou decrementado após cada iteração. A sintaxe é:

```

REPETE nome [=] valor1 [A] valor2 [PASSO valor]
.... (comandos) ....
FIM

```

nome se refere ao nome de uma variável numérica de controle, que não deve ser macro-substituído; valor1 é o valor inicial assumido pela variável de controle e valor2 o valor final; o incremento da variável de controle após cada repetição é de menos que um valor diferente seja especificado sinal de "=" e a letra "A" entre valor1 e valor2 são opcionais, e servem apenas de comentário.

O conjunto de comandos a ser repetido deve ser terminado pela palavra FIM. Como exemplo, vamos determinar o peso médio de um conjunto de N tipos de elementos, onde:

N Número de tipos de elementos
Pi Peso de cada elemento: P1, P2, ... Pn
Qi Quantidade de cada elemento: Q1, Q2, ... Qn
MEDIA Média ponderada
NTOT Número total de elementos

O programa será:

```
N MEDIA= 0
N NTOT= 0
REPETE I= %N
N MEDIA= %MEDIA + (%P%I * %Q%I)
N NTOT= %NTOT + %Q%I
FIM
N MEDIA= %MEDIA / %NTOT
```

A lógica do comando REPETE é:

Atribuir o valor inicial à variável de controle;

Verificar se a variável de controle é superior ao valor limite (se o passo for negativo verifica se é inferior); se for, sair do comando;

Caso contrário, executar todos os comandos até o próximo fim;

Encontrando o FIM, somar o incremento à variável de controle e voltar para (b).

O DP permite que a variável de controle seja alterada dentro do conjunto de comandos em repetição; no entanto

deve-se evitar este tipo de procedimento, que dificulta a compreensão do programa.

Entre os comandos que podem ser repetidos incluem-se outros comandos de repetição, SEs e ENQUANTOS.

ENQUANTO

Este comando repete um conjunto de comandos terminados por FIM, enquanto for válida uma condição. A sintaxe é:

```
ENQUANTO expressão lógica
.... (comandos) ....
FIM
```

Por exemplo, suponha que devamos desenhar linhas entre os nós 1 e 2 até que a abscissa X do nó 1 atinja 1000:

```
N DELTAX50
10,0
20,200
ENQUANTO%X1 < 1000
POL1 2
11 @ %DELTAX, 0
22 @ %DELTAX, 0
FIM
```

O DP verifica a validade da expressão antes de executar o conjunto de comandos; se for verdadeira, os comandos serão executados, caso contrário não. Ao encontrar o FIM correspondente, volta-se à expressão lógica e faz-se o teste novamente, e assim por diante, até que a expressão se torne falsa.

Podem fazer parte do conjunto de comandos repetidos por este comando outros ENQUANTOs, SEs e REPETEs.

QUEBRA

É conveniente, às vezes, controlarmos a saída de um REPETE ou ENQUANTO de outro modo além do teste feito no início. O comando QUEBRA permite uma saída antecipada da repetição; ele termina imediatamente o laço mais interno.

Suponha que precisemos desenhar 10 linhas com incremento de coordenadas DELTAX, DELTAY, mas a abscissa X do primeiro nó não pode ser superior a 1000:

```
N DELTAX150
N DELTAY100
10,0
20,200
REPETE I= 10
SEX1 > 1000
QUEBRA
FIM
POL1 2
11 @ %DELTAX, %DELTAY
21 @ %DELTAX, %DELTAY
FIM
```

CONTINUA

Este comando funciona de forma semelhante ao QUEBRA, iniciando imediatamente a próxima iteração do REPETE ou ENQUANTO atual. Por exemplo, no problema de cálculo da média ponderada, suponha que não desejemos que os valores de peso negativo entrem no cálculo:

```
N MEDIA= 0
N NTOT= 0
REPETE I= %N
SE%P%I < 0
CONTINUA
FIM
N MEDIA= %MEDIA + (%P%I * %Q%I)
N NTOT= %NTOT + %Q%I
FIM
N MEDIA= %MEDIA / %NTOT
```

SAÍDA

Em algumas situações é preciso abandonar o processamento de um subprograma antes do seu fim. O comando SAÍDA termina imediatamente a execução. Por exemplo, se um subprograma não puder ser executado se a variável COMPR valer zero:

```
SE %COMPR = 0
```


SAIDA

FIM

Aninhamento de SE, REPETE, ENQUANTO

Os comandos acima controlam a execução de um conjunto de comandos. Neste conjunto pode haver outros comandos SE, REPETE e ENQUANTO, que controlarão outros conjuntos e assim por diante. As seguintes regras são válidas:

O comando FIM se aplica ao último SE, REPETE ou ENQUANTO executado;

Os comandos QUEBRA e CONTINUA se aplicam ao último REPETE ou ENQUANTO executado.