

Variáveis e Macro-Substituição

Nos capítulos anteriores vimos como é possível especificar um desenho completo por meio da linguagem DP. Nosso objetivo agora é especificar desenhos em função de parâmetros, e para isto é fundamental a introdução do conceito de variáveis.

Variáveis no DP são posições de memória que podem assumir qualquer valor. Os tipos de valores reconhecidos pelo programa são:

Numéricos	Valores em ponto flutuante com 15 dígitos significativos
Alfanuméricos	Cadeias de até 80 caracteres
Coordenadas	Par de valores numéricos

Variáveis podem ter nomes formados por até 8 caracteres ou números (caracteres adicionais serão ignorados). Variáveis numéricas devem ter nome alfanumérico começando por uma letra; variáveis tipo coordenadas podem também ter um nome numérico.

Atribuição de valores a variáveis numéricas

O comando NUM permite atribuir um valor a uma variável numérica. A sintaxe é:

```
NUM nome [=] valor]
```

A palavra NUM pode para maior comodidade ser abreviada para N. Por exemplo, para atribuir o valor de 10 para a variável RAI0:

```
NUM RAI0 = 10
```

O sinal de = é opcional, e serve somente de comentário. Outras formas aceitáveis para o mesmo comando:

```
N RAI0 = 10
```

```
N RAI0 10
```

Se o valor de uma variável numérica não for definido, zero será assumido. "valor", como vimos no capítulo [Localção](#)

Geométrica (pág. 16), pode ser qualquer expressão aritmética que resulte em um número, incluindo funções e operadores geométricos.

Macro-substituição de variáveis numéricas

Agora que já sabemos como definir uma variável numérica, vamos tentar usá-la. Como exemplo, definiremos um círculo de raio RAIO, nas coordenadas 0,0:

```
CIRCULO C 0,0 R RAIO
```

```
*
```

```
***** ERRO: Valor esperado
```

O programa não aceita a variável RAIO no lugar do valor do raio. O mesmo acontece em toda a definição de geometria; se nomes de variáveis fossem aceitos em qualquer lugar, interfeririam com as palavras chaves da linguagem geométrica (que não foi alterada, para manter compatibilidade com outros programas).

A solução utilizada no DP foi o uso do recurso de macro-substituição: consiste em trocar o nome da variável pelo seu valor, antes da linha de comando ser passada para interpretação. A macro-substituição é feita colocando-se o sinal % antes do nome da variável:

```
CIRCULO C 0,0 R %RAIO
```

Ao ler esta linha, a rotina de leitura verifica a existência de um sinal % e procura o valor da variável RAIO, substituindo-a:

```
CIRCULO C 0,0 R 10
```

A linha obtida é finalmente enviada para interpretação, resultando no esperado: um círculo de centro 0,0 e raio 10.

Precisão da macro-substituição

Variáveis numéricas são substituídas com a máxima precisão possível, limitada a 15 dígitos significativos. O programa automaticamente elimina zeros redundantes à direita da vírgula.

Nem sempre a substituição com a precisão máxima é interessante. O DP permite controlar o número de casas depois da vírgula, com a notação:

```
%.Nnome
```

onde N é o número de casas fixadas depois da vírgula, com arredondamento da última casa. N pode valer de 9. Por exemplo, atribuindo-se o valor

```
N PI = 3.141593
```

teremos como resultado das substituições:

```
codificação resultado
```

```
-----
```

```
%PI 3.141593
```

```
%.0PI 3
```

```
%.3PI 3.142
```

```
%.4PI 3.1416
```

```
%.5PI 3.14159
```

Note que um número com zero casas depois da vírgula é arredondado para o inteiro mais próximo, enquanto que a função FIX (num) trunca o número. Por exemplo:

```
N VAL 3.6
```

```
N A FIX (%VAL)
```

```
N B %.0VAL
```

resultará em A valendo 3 e B valendo 4. O sinal de (-) dentro da macro-substituição permite fazer também com que os zeros não sejam suprimidos, e o número de dígitos mostrado seja sempre exato. Por exemplo, se atribuirmos o valor 3.1 para a variável "A":

```
N A3.1
```

podemos substituir "A" por:

```
codificação resultado
```

```
-----
```

```
%A 3.1
```

```
%.2A 3.1
```

```
%-.2A 3.10
```

```
%.6A 3.1
```

%-.6A 3.100000

Atribuição de variáveis alfanuméricas

Variáveis alfanuméricas armazenam cadeias de caracteres alfanuméricos. Cadeias de caracteres são definidas sempre entre apóstrofes:

```
A TESTE 'Teste Alfanumérico'
```

O comando acima define a variável TESTE com uma cadeia de caracteres. São definições aceitáveis também:

```
ALF TESTE 'Teste Alfanumerico'
```

```
ALF TESTE='Teste Alfanumerico'
```

```
A TESTE='Teste Alfanumerico'
```

Variáveis alfanuméricas são usadas em desenho principalmente para armazenar e manipular textos, colocados com o comando TEXTO. A definição de uma variável alfanumérica sem valor, tal como em:

```
A TESTE
```

faz com que a variável assuma valor nulo, isto é, uma cadeia com zero caracteres.

Macro-substituição de variáveis alfanuméricas

Variáveis alfanuméricas são substituídas da mesma forma que variáveis numéricas. Um detalhe importante é que a macro-substituição não coloca apóstrofes; se necessários, devem ser colocados manualmente. Vamos por exemplo colocar o texto contido na variável TESTE nas coordenadas (0,0) do desenho:

```
TEXTO 0,0 '%TESTE'
```

Durante a leitura, o DP substituirá o comando por:

```
TEXTO 0,0 'Teste Alfanumerico'
```

Note que se esquecêssemos os apóstrofes, resultaria em:

```
TEXTO 0,0 %TESTE
```

```
TEXTO 0,0 Teste Alfanumerico
```

Neste caso o DP acusaria erro na palavra Teste, que não faz parte da sintaxe do comando TEXTO.

Outros usos para variáveis alfanuméricas

Variáveis alfanuméricas não estão restritas ao comando TEXTO. Como a substituição é sempre sem apóstrofes, pode-se usar uma variável alfanumérica para alterar a sintaxe de um comando. Por exemplo,

```
A TLIN 'POLIGONAL'
```

```
%TLIN 1 2 3 4
```

resultará em uma poligonal por 4 pontos, enquanto que

```
A TLIN 'CURVA'
```

```
%TLIN 1 2 3 4
```

resultará numa curva por 4 pontos.

Manipulação de cadeias de caracteres

Cadeias de caracteres podem ser manipuladas através de operações de concatenação, extração de sub-cadeias e pesquisa de posição. Como decorrência, as variáveis alfanuméricas podem ser igualmente manipuladas.

Concatenação

Cadeias de caracteres podem ser concatenadas naturalmente pelo processo de macro-substituição. Exemplo:

```
AVAR1 'C=1000'
```

```
AVAR2 'C/16'
```

```
AVAR3 '%VAR1 %VAR2'
```

resultará em VAR3 valendo 'C=1000 C/16'. O espaço entre %VAR1 e %VAR2 é necessário, como veremos adiante. Qualquer tipo de variável pode ser usado na macro-substituição, e não apenas variáveis alfanuméricas.

Extração de sub-cadeias

As funções LEFT, RIGHT e MID permitem extrair um subconjunto de uma cadeia de caracteres. Estas funções podem ser colocadas em qualquer lugar onde o programa espere por uma cadeia de caracteres.

A função LEFT extrai os "n" primeiros caracteres de uma cadeia. Por exemplo,

```
A VAR LEFT ('ABCDEFGH', 3)
```

resultará na variável VAR valendo 'ABC'. A função RIGHT extrai os últimos "n" caracteres:

```
A CAD 'ABCDEFGH'
```

```
A VAR RIGHT ('%CAD', 3)
```

resultará na variável VAR valendo 'FGH'. Por último, a função MID extrai um subconjunto de caracteres em qualquer posição de uma cadeia:

```
A VAR MID ('%CAD', 3, 2)
```

resultará na variável VAR valendo 'CD', isto é, 2 caracteres a partir da 3a posição da cadeia original. A função MID pode ser chamada também sem o último parâmetro (o número de caracteres). Neste caso, todos os caracteres a partir de uma determinada posição serão tomados. Por exemplo,

```
A VAR MID ('%CAD', 3)
```

resultará no valor 'CDEFGH' para a variável VAR.

Pesquisa de posição

A função POS busca a posição de uma sub-cadeia dentro de uma cadeia maior de caracteres. Por exemplo,

```
N IPOS POS ('ABCDEFGH', 'DEF')
```

resultará em IPOS valendo 4, pois a cadeia 'DEF' é uma sub-cadeia de 'ABCDEFGH' a partir da 4a posição, enquanto que:

```
N IPOS POS ('ABCDEFGH', 'XYZ')
```

resultará em IPOS igual a zero, pois a cadeia 'XYZ' não é um subconjunto de 'ABCDEFGH'. Façamos um exemplo mais complexo: dada a cadeia

```
'5 P7 { 6.3 C=200'
```

extrair o número que vem depois da letra 'P' na variável POSIC e o número que vem depois das letras C= na variável COMPR:

- a) ACAD'5 P7 { 6.3 C=200'
- b) NIPOS ('%CAD', 'P')
- c) ATMPMID ('%CAD', %I+1)
- d) NIPOS ('%TMP', ' ')
- e) ATMPMID ('%TMP', 1, %I-1)
- f) NPOSIC%TMP
- g) NIPOS ('%CAD', 'C=')
- h) ATMPMID ('%CAD', %I+1)
- i) NCOMPR%TMP

Neste exemplo:

- a) A variável CAD recebeu a cadeia de caracteres original;
- b) A variável numérica I tem a posição da letra 'P' na cadeia CAD;
- c) A variável TMP tem todos os caracteres a direita da cadeia CAD;
- d) A variável I agora tem a posição do espaço em branco após o número 7 (em P7");
- e) A variável TMP agora tem somente o número após a letra P, que começa no primeiro caractere de TMP e vai até o próximo espaço em branco menos um caractere;
- f) Transportamos a variável TMP, alfanumérica para a variável POSIC, numérica;
- g) Achamos agora a posição do 'C=' dentro da cadeia CAD;
- h) TMP agora vale todos os caracteres à direita de C=; eles já são o número desejado;
- i) Transportamos finalmente o número obtido para a variável numérica COMPR.

Note que:

Embora os valores alfanuméricos tenham sido transportados para variáveis numéricas, eles poderiam ter sido usados diretamente em expressões aritméticas; isto por que o que importa não é o conteúdo da variável, mas o resultado da macro-substituição;

O DP não permite que o resultado das funções LEFT, RIGHT e MID sejam atribuídos diretamente a variáveis numéricas;

Variáveis do lado direito são sempre precedidas do sinal "%", enquanto que do lado esquerdo não.

Foi criada uma variável de trabalho, I, que na verdade poderia ter sido eliminada se tivéssemos colocado o resultado da função POS diretamente dentro da função MID. O exemplo acima seria equivalente a:

```
A CAD '5 P7 { 6.3 C=200'  
A TMP MID ('%CAD', POS ('%CAD', 'P')+1)  
A TMP MID ('%TMP', 1, POS ('%TMP', ' ')-1)  
N POSIC %TMP  
A TMP MID ('%CAD', POS ('%CAD', 'C=')+1)  
N COMPR %TMP
```

Comprimento de uma cadeia de caracteres

A função LEN retorna o número de caracteres de uma cadeia, e pode ser usada dentro de uma expressão aritmética. Por exemplo:

```
N NCLEN ('ABCDEFGH')
```

fará com que a variável NC valha 8.

Comparação de duas cadeias

A função CMP compara duas cadeias de caracteres, retornando o valor 0 se as duas forem iguais ou 1 se forem diferentes:

```
N ACMP ('ABC', 'ABC')  
N BCMP ('ABC', 'DEF')
```

fará com que A valha 0 e B função CMP permite que cadeias de caracteres possam ser comparadas em expressões condicionais, a serem vistas em um capítulo posterior.

Atribuição de coordenadas

O terceiro tipo de variável suportado pelo DP é o tipo coordenadas X,Y. Coordenadas podem ser definidas do seguinte modo:

```
COORD VAR 100,50
```

que significa, variável VAR com coordenadas (100,50). São aceitáveis também:

```
COORD VAR = 100,50
```

```
C VAR 100,50
```

```
C VAR = 100,50
```

Qualquer valor composto de coordenadas pode ser usado na definição de uma variável do tipo par de coordenadas. É aceitável, por exemplo:

```
C VAR 10 @ 100 + 20 + 20 < DIR 15 16 - 180
```

A definição de uma variável do tipo coordenadas é bastante semelhante à definição geométrica de nós, vista anteriormente.

Macro-substituição de coordenadas

O processo de macro-substituição de coordenadas é idêntico ao dos outros tipos de variáveis. Assim, os comandos:

```
C VAR 100,50
```

```
1 %VAR
```

resultarão no segundo comando sendo interpretado como

```
1 100,50
```

Nós e variáveis tipo coordenadas

A definição de nós vista no capítulo *Locação Geométrica* nada mais é do que a criação de variáveis tipo par de coordenadas, cujo nome é numérico. A macro-substituição de coordenadas de nós é permitida, e o comando:

```
C VAR2 %1
```

será interpretada como

```
C VAR2 100,50
```

se o nó 1 valer 100,50. A criação de variáveis deste tipo e a sua manipulação são na maioria dos casos mais vantajosa através da sintaxe vista no capítulo Locação Geométrica. A extensão da linguagem DP para tratar nós como variáveis tem como objetivo permitir a passagem de parâmetros tipo nós entre subprogramas DP, assunto a ser tratado em outro capítulo.

Caractere nulo

O nome de uma variável normalmente é delimitado por um caractere não alfabético, tal como um espaço a seguinte construção:

```
A VAR1 'ABCD'  
A VAR2 '%VAR1 EFGH'
```

VAR2 neste caso assumirá o valor 'ABCD EFGH'. Como fazemos para retirar o espaço em branco entre a letra D e a letra E? Se fizermos:

```
A VAR2 '%VAR1EFGH'
```

resultará no DP acusando um erro de variável indefinida: VAR1EFGH. Para solucionar o problema de encostar um texto do lado direito do nome de uma variável, usamos dois caracteres %%. O DP, quando encontra um sinal de por cento seguido do outro, joga os dois fora. Para resolvermos o problema anterior, faremos então:

```
A VAR2 '%VAR1%%EFGH'
```

que resultará em VAR2 valendo 'ABCDEFGH'. A seqüência %% é chamada aqui de Caractere Nulo.

Dupla Substituição

O DP analisa de forma diferente duas variáveis "encostadas" uma na outra. Por exemplo,

```
A VAR %A%I
```

será interpretado da seguinte forma: primeiro, a variável "I" é substituída. O resultado alterará o nome da variável "A", e a substituição final será feita sobre o nome obtido. Por exemplo:

```
N I 1  
A A1 'CA25'  
A A2 'CA50A'  
A A3 'CA50B'
```

```
A A4 'CA60B'
```

```
A VAR '%A%I'
```

Primeiro, a variável "I" será substituída, resultando em:

```
A VAR '%A1'
```

O resultado será novamente substituído:

```
A VAR 'CA25'
```

A dupla substituição é muito útil para a manipulação de tabelas, principalmente dentro de laços de programa, como veremos adiante.

Quando a dupla substituição não é desejada, deve-se incluir o caractere nulo separando as variáveis, tal como:

```
A VAR '%A%%I'
```

onde o programa substituirá separadamente a variável "A" e a variável "I".

Variáveis do sistema

Existe um conjunto de variáveis que são sempre definidas pelo próprio DP. O objetivo destas variáveis é permitir maior controle do programa. Todas as variáveis do sistema começam pelo caractere traço "_" (também chamado de underscore). Abaixo estão listadas as variáveis do sistema, seus tipos e seus defaults, considerando o INSTAL.LDF distribuído com o DP:

Nome	Tipo	Default	Significado
_IAPLIC	N	9	Tipo de sistema:(1) genérico(2) TQS-Formas(3) TQS-Lajes(4) TQS-Fundações(7) TQS-Vigas(8) TQS-Pilar(9) TQS-AGC&DP(11) TQS-Madeira(15) CORBAR(16) CORMAD(17) TQS-Alvest
_ISUBAPLIC	N	1	Tipo de subsistema. Para cada sistema temos vários subsistemas. Veja na tabela a seguir, a correlação entre os sistemas e subsistemas
_ERROS	N	0	Numero de erros detectados

_ESCALA	N	50	Escala atual
_TAMTTX	N	0.24	Tamanho de texto, comando TEXTO
_TAMTEI	N	0.4	Tamanho de texto para o comando EIXOS
_NIVEL	N	0	Nível de desenho atual
_TAMTCO	N	0.22	Tamanho do texto de cotagem
_COTNIV	N	221	Nível de cotagem
_COTLCH	N	1	Cotagem c/ linha de chamada (0) não (1) sim
_COTMUL	N	1	Multiplicador de dimensões de cotagem
_COTBLO	A	'TICK'	Bloco de cotagem
_COTNIL	N	-1	Nível da linha de cotagem (-1=COTNIV)
_COTNIC	N	-1	Nível da linha de chamada (-1=COTNIV)
_TABPLT	A	''	Tabela de plotagem
_DATA	A	data atual	Data do sistema
_HORA	A	hora atual	Hora do sistema
_DIRET	A	pasta atual	Pasta atual de trabalho
_BIBDP	A	'%_SUPORTE\DP\DPS'	Biblioteca de subprogramas DPS
_BIBBLO	A	'%_SUPORTE\BLOCOS\GERAIS'	Biblioteca de blocos
_BIBINC	A	'.' (PASTA ATUAL)	Biblioteca de arquivos de inclusão
_SUPORTE	A	'C:\TQSW\SUPORTE'	Pasta de arquivos de critérios

Tabela de Sistemas e subsistemas:

Sistema (_IAPLIC)	Subsistema (_Isubaplic)	Descrição
(1) Genérico	1	Desenho qualquer
	2	Moldura / Carimbo para plotagem
	3	Tabela de ferros
	4	Layout de plantas

	5	Elementos extras sobre as plantas
	6	Plotagem em desenho
	7	Interpretação de plotagem
(2) TQS-Formas	1	Entrada gráfica de formas
	2	Planta de formas
	3	Verificação de formas
	4	Visualização de Pórtico / Grelha
	5	Entrada gráfica de grelha
	6	Superposição de cargas em pilares
	7	Reações de pórtico em desenho
	8	Esquema gráfico do edifício
	9	Planta de locação de pilares
(3) TQS-Lajes	1	Armação positiva de lajes
	2	Armação negativa de lajes
	3	Esforços, processo simplificado
	4	Armação de cisalhamento / punção
	5	Diagrama do editor de esforços
	6	Faixas do editor de esforços
	7	Armação protendida
	8	Tabela de Aço Protendido
	9	Elevação de cabos de protensão
(4) TQS-Fundações	1	Armação de sapatas
	2	Armação de blocos
(7) TQS-Vigas	1	Armação de vigas
	2	Esquema gráfico de vigas
(8) CADPilar	1	Armação de pilares
	2	Locação de pilares no piso

	3	Seções de pilares
(9) TQS-AGC&DP	1	Desenho genérico de armaduras
	2	Biblioteca de tipos de ferros
	3	Lista de ferros desenhada
	4	Detalhes de armação (sem tabela de ferros)
(11) TQS-Madeira	1	Desenho de formas de madeira
(15) CORBAR	1	Desenho genérico
(16) CORMAD	1	Desenho genérico
(17) TQS-Alvest	1	Desenho de alvenarias

As variáveis do sistema não podem ser modificadas diretamente pelo comando de atribuição de variáveis NUM e ALF, mas a maioria é controlada por DEFINES. Por exemplo,

```
DEFINE ESCALA 20
```

faz com que imediatamente, a variável `_ESCALA` passe a valer 20. Para tornar um desenho independente de escala, poderíamos multiplicar medidas pela escala atual, tal como na definição do nó 8:

```
8 1 @ 0, -1.3*%_ESCALA
```

Variáveis e macro substituição: Exemplo

Vamos agora refazer o exemplo visto no capítulo anterior, do desenho do paralelepípedo. Desta vez, colocaremos os parâmetros do desenho nas seguintes variáveis:

```
COMPRIM Comprimento do paralelepípedo
```

```
LARGURA Largura
```

```
ALTURA Altura
```

```
TITULO Título do paralelepípedo
```

Para tornar o desenho independente da escala, usaremos a variável `_ESCALA` para calcular as posições relativas das vistas e das cotagens em relação ao resto do desenho. O resultado será o arquivo EXEMPLO2.DP, mostrado a seguir:

```

DESENHO 'EXEMPLO2'
DEFINE ESCALA 50
$
$Definição dos parâmetros do desenho
$
N COMPRIM = 400
N LARGURA = 150
N ALTURA = 100
A TITULO = 'PARALELEPIPEDO'
$
$Vista em Planta
$
1 0,0
3 1 @%COMPRIM,%LARGURA
2 X3,Y1
4 X1,Y3
POLIGONAL 1 2 3 4 1
$
$Vista frontal
$
8 1 @0,-1.3*%_ESCALA
6 8 @%COMPRIM,-%ALTURA
5 X8,Y6
7 X6,Y8
POLIGONAL 5 6 7 8 5
$
$Vista lateral
$
9 2 @2.5*%_ESCALA,0
11 9 @%ALTURA,%LARGURA
10 X11,Y9
12 X9,Y11
POLIGONAL 9 10 11 12 9

```

```

$
$Nos auxiliares
$
13 5 @-1*%_ESCALA,0
14 4 @0,1*%_ESCALA
15 (X1 + X3)/2, (Y1 + Y3)/2
$
$Cotagens
$
COTAGEM VERTICAL 5 8 13
COTAGEM VERTICAL 1 4 13
COTAGEM HORIZONTAL 4 3 14
COTAGEM HORIZONTAL 12 11 14
$
$Título do paralelepípedo, centrado
$
TEXTO15 CENTRADO '%TITULO'
FIM

```

Listagem de saída

O processamento de um arquivo .DP resulta em outro arquivo, de mesmo nome e com tipo .LST, que é chamado de listagem de saída. Este arquivo contém uma cópia do arquivo .DP com as linhas numeradas e com a indicação de erros, se ocorrerem.

Um ponto interessante é que as linhas listadas não são as codificadas originalmente, mas as resultantes do processo de macro-substituição. Por exemplo, a seqüência de linhas:

```

N COMPRIM = 400
N LARGURA = 150
3 1 @ %COMPRIM, %LARGURA

```

aparecerá na listagem de saída como:

```

N COMPRIM = 400

```

```
N LARGURA = 150
```

```
3 1 @ 400, 150
```

Isto facilita a correção de erros , pois mostram claramente quando uma linha não foi aceita por erro de substituição.

A listagem de saída pode ser suprimida a partir do comando

```
DEFINE NLISTA
```

e novamente habilitada pelo comando

```
DEFINE LISTA
```

O DP emite listagem de saída para o programa DP, mas não o faz nos subprogramas. A este respeito trataremos no próximo capítulo.