

Estrutura Geral de um Programa Procedural:

Declaração de Subrotinas e Funções:

DECLARE SUB procedimento1 (**args**)

DECLARE FUNCTION procedimento2 (**args**) **AS TYPE**

Início do Programa Principal:

Declaração de constantes:

CONST nomeconst **AS TYPE** = valor da constante

Declaração de variáveis:

DIM nomevar **AS TYPE**

Declaração de variáveis indexadas:

DIM variavel() **AS TYPE**

Declaração e dimensionamento de variáveis indexadas unidimensionais (vetores):

DIM variavel(**n**) **AS TYPE**

Declaração e dimensionamento de variáveis indexadas multidimensionais (matrizes):

DIM variavel(**n, m, ...**) **AS TYPE**

onde

TYPE = **INTEGER, SINGLE, DOUBLE, STRING**, etc.

n, m, ... = os índices máximos do vetor ou matriz

Início do programa principal com chamadas aos procedimentos (Sub e Function):

procedimento1 (args1) <----- procedimento1 é uma Sub

var = procedimento2 (args2) <----- procedimento2 é uma Function (retorna valor que é
 armazenado na variável var

etc...

END <----- Fim do programa principal

Início das Definições dos Procedimentos:

SUB procedimento1 (**args**) <-----**args** contém as variáveis que serão passadas entre
 o programa de chamada e este procedimento

DIM var **AS TYPE** <-----var é uma variável local do procedimento1

 etc...

END SUB

FUNCTION procedimento2 (**args**) **AS TYPE** <-----este último TYPE se refere ao tipo de valor
 que será retornado pela função
 no comando **RETURN**

DIM var1 **AS TYPE** <-----var1 é uma variável local do procedimento3

 etc...

RETURN var1 <-----var1 é uma variável local e deve ter o mesmo tipo do
 valor de retorno descrito na declaração da Function

END FUNCTION

onde **args** são os argumentos dos procedimentos, que podem ser passados por referência (**BYREF**) ou por valor (**BYVAL**) e tem a seguinte sintaxe:

(**BYVAL** var1 **AS TYPE**, **BYREF** var2 **AS TYPE**, var3() **AS TYPE**, var4(,,) **AS TYPE**)

com as seguintes observações:

var1 é uma variável passada por valor (comando **BYVAL**). Isto quer dizer que se o procedimento alterar o valor desta variável var1, este novo valor **não** será retornado ao programa de chamada;

var2 é uma variável passada por referência (comando **BYREF**), ou seja, **é referenciada pelo seu endereço na memória do computador**. Isto significa que, caso o procedimento altere o valor desta variável, este novo valor **será** retornado ao programa de chamada;

var3() é uma variável indexada unidimensional (vetor). Tais variáveis são **sempre passadas por referência** e portanto não se deve colocar os comandos **BYREF** ou **BYVAL** na declaração

var4(,,) é uma variável indexada tridimensional (matriz 3D). como vetores e matrizes **são sempre passadas por referência**, não se deve colocar **BYREF** ou **BYVAL** na declaração

Fim das definições dos procedimentos

=====

Fim do programa

Observação:

DIM var() **AS TYPE** <---- Declaração da variável indexada (ou vetor) unidimensional var()

REDIM var(**n**) <---- Dimensionamento ou alocação da variável indexada var() com **n+1** elementos na memória do computador, apresentando os seguintes elementos: var(**0**), var(**1**), var(**2**), ... var(**n**)