**Arrays em ADVPL**

**Estrutura de Declaração de Array**

Em ADVPL, arrays são estruturas de dados que permitem armazenar múltiplos valores em uma única variável. A declaração de arrays segue uma estrutura específica.

Para declarar um array, usamos a palavra-chave local para indicar que é uma variável local, seguida pelo nome do array e a atribuição de um array vazio {}.

A forma padrão de declarar um array é:

Local aExemploArray := {}  
  
Os arrays em ADVPL podem conter diversos tipos de dados, incluindo textos, valores numéricos, valores lógicos, blocos de código e nulo.

### Declarando Arrays

A seguir, vamos ver exemplos de como declarar e inicializar arrays em ADVPL:

#### Incluindo a Biblioteca

Primeiro, incluímos a biblioteca necessária:  
  
#include 'totvs.ch'  
  
User Function U\_ARRAY

// Declarando arrays vazios

Local aArray1 := {}

// Declarando arrays com valores pré-definidos

Local aArray2 := {'1', '2', 3, 4, '6'}

// Declarando um array com tamanho inicial de zero

Local aArray3 := array(0)

// Declarando um array com duas posições

Local aArray4 := array(2)

// Declarando um array com uma posição contendo outro array com valores pré-definidos

Local aArray5 := array(1, {"5", "4", "3"})

// Declarando um array com duas posições, cada uma contendo um array diferente

Local aArray6 := array({5, 6, 7}, {'op 5', 'op 6', 'op 7'}) Return

### Explicação

**Array Vazio**:

advpl

Local aArray1 := {}

Aqui, aArray1 é declarado como um array vazio.

**Array com Valores Pré-Definidos**:

Local aArray2 := {'1', '2', 3, 4, '6'}

aArray2 é declarado e inicializado com valores textuais e numéricos.

**Array com Tamanho Inicial de Zero**:

Local aArray3 := array(0)

aArray3 é declarado como um array de tamanho zero. Este pode ser útil quando você precisa de um array dinâmico que crescerá conforme necessário.

**Array com Tamanho Específico**:

Local aArray4 := array(2)

aArray4 é declarado como um array com duas posições, que inicialmente conterão valores nulos.

**Array com Conteúdo Pré-Definido**:

Local aArray6 := array({5, 6, 7}, {'op 5', 'op 6', 'op 7'})

aArray6 é um exemplo de array aninhado, onde a primeira posição contém um array de números e a segunda posição contém um array de textos.

### Adicionando Valores a um Array

É possível adicionar valores a um array após sua declaração usando a função AAdd(). Esta função recebe dois parâmetros: o array ao qual o valor será adicionado e o valor a ser adicionado.

#### Exemplos de Uso da Função AAdd()

aadd(aArray1, 1)

aadd(aArray2,2)

aadd(aArray1,'3')

aadd(aArray2, {'1','2','3'},{'teste1', 'teste2'})

**Adicionando Valores Simples**:  
AAdd(aArray1, 1)

AAdd(aArray1, 2)

AAdd(aArray1, '3')

**Adicionando Matriz:**aadd(aExemplo2, {'1','2','3'},{'teste1', 'teste2'})

**Alteração de valores do array**

**A alteração de valores do array é feito através da seguinte sintaxe:**

aArray4[1] := ‘valor 1’   
  
dessa forma supõem que o dev sabe que o array possui pelo menos uma posição pois o mesmo está sendo especifico na posição que quer inserir o valor no array.

**Função para ordenar o array**