

Interpretador Hall – Operação com Matrizes

Veja abaixo um exemplo de algumas funções para se trabalhar com matrizes no *interpretador Hall*. O trabalho com matrizes no interpretador poderá ser realizado com dois enfoques:

- lógica de programação
- matemático

Enfoque de Lógica de Programação

Vejamos, inicialmente, o exemplo com o enfoque da lógica de programação. Nosso exemplo consistirá na rotinas de cálculo para se adicionar duas matrizes. No enfoque da lógica de programação é necessário descrever para o computador o modo como ele irá realizar o cálculo. A propósito, em um curso de lógica de programação, o objetivo é esse mesmo, o de que o aluno aprenda a raciocinar de forma algorítmica.

Veja abaixo o código

```
Matriz – Lógica de Programação
Algoritmo()
{
    // declaracao de variaveis

    matriz inteiro A[3:4];
    matriz inteiro B[3:4];
    matriz inteiro C[3:4];
    inteiro i,j;;

    // inicializando a matriz A[]

    para ( i := 1 ate 3 passo 1 )
    {
        para ( j := 1 ate 4 passo 1 )
        {
            leia("A: ", A[i][j] );
        }
    }

    // inicializando a matriz B[]

    para ( i := 1 ate 3 passo 1 )
    {
        para ( j := 1 ate 4 passo 1 )
        {
            leia("B: ", B[i][j] );
        }
    }
}
```

Matriz - Lógica de Programação

```
    }  
  }  
  
  // calculando a soma das matrizes A[] e B[]  
  
  para ( i := 1 ate 3 passo 1 )  
  {  
    para ( j := 1 ate 4 passo 1 )  
    {  
      C[i][j] := A[i][j] + B[i][j];  
    }  
  }  
  
  // exibindo o resultado  
  
  para ( i := 1 ate 3 passo 1 )  
  {  
    para ( j := 1 ate 5 passo 1 )  
    {  
      escreva(C[i][j], " ");  
    }  
  }  
}
```

Comentário

A grande vantagem do enfoque “lógica de programação” é fazer com que o aluno aprenda a raciocinar com os índices dos elementos das matrizes. É um excelente aprendizado para um futuro cálculo numérico que o aluno poderá enfrentar um pouco mais adiante nos estudos.

Vejamos o enfoque matemático.

Enfoque Matemático

Vejamos abaixo a resolução do mesmo algoritmo pela abordagem do enfoque matemático, isto é, utilizando os recursos pré-definidos do interpretador para o tratamento de matrizes.

Veja o código.

```
Matriz – Lógica de Programação
Algoritmo()
{
    // declaracao de variaveis

    matriz inteiro A[3:4];
    matriz inteiro B[3:4];
    matriz inteiro C[3:4];

    // inicializando as matrizes A[] e B[]

    A[] := (0,2,4,6,8,10,12,14,16,18,20,22,24,26,28);
    B[] := (1,3,5,7,9,11,13,15,17,19,21,23,25,27,29);

    // adicionando as matrizes A[] e B[]

    C[] := AddMatriz(A[], B[]);

    // exibindo o resultado

    escreva(C[]);
}
```

Comentário:

Facilmente observamos que o código no enfoque matemático é bem menor e mais conciso do que a correspondente versão algorítmica. Caso o objetivo do professor seja o de usar o interpretador para ilustrar conceitos matemáticos, o enfoque matemático é mais direto e objetivo. Há várias funções disponíveis para o tratamento de matrizes, como por exemplo:

- AddMatriz()
- DifMatriz()
- ProdMatriz()
- Transposta()
-
-