

• UEMS •

Universidade Estadual
de Mato Grosso do Sul

Universidade Estadual de Mato Grosso do Sul

Ciência da Computação

Algoritmos e Estruturas de Dados I (AED-I)

Matriz

Prof. Nilton

nilton@comp.uems.br

Sumário

- Construção de algoritmos usando matrizes

Matriz

- Uma estrutura para guardar **vários dados**;
- Os dados devem ser do **mesmo tipo**;
- A matriz possui um tamanho **pré-definido**;
- Um dado é acessado usando **dois índices**;
- Um índice é uma **posição da linha** e outro **posição da coluna** da matriz.

Matriz

	Aluno		
	1	2	3
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

- No exemplo, **Aluno** é uma matriz 10x3.
- As 10 linhas podem representar a **quantidade de alunos** e a 1º coluna a 1ª nota de uma avaliação, a 2º coluna a 2ª nota e a 3º coluna a **média** entre a duas avaliações.

Declarando uma matriz

```
var  
aluno:vetor[1..10,1..3] de real
```

Guardando um valor em uma posição da matriz

```
//armazena as notas do PRIMEIRO aluno  
aluno[1,1] <- 8.0 //primeira nota  
aluno[1,2] <- 9.0 //segunda nota  
//calcula a media das notas  
aluno[1,3] <- (aluno[1,1] + aluno[1,2]) / 2  
  
//armazena as notas do SEGUNDO aluno  
aluno[2,1] <- 7.5 //primeira nota  
aluno[2,2] <- 7.5 //segunda nota  
//calcula a media das notas  
aluno[2,3] <- (aluno[2,1] + aluno[2,2]) / 2
```

Mostrando um valor de uma posição da matriz

```
var
  aluno:vetor[1..10,1..3] de real

inicio
  aluno[1,1] <- 8.0    //primeira nota
  aluno[1,2] <- 9.0    //segunda nota
  //calcula a média das notas
  aluno[1,3] <- (aluno[1,1] + aluno[1,2]) / 2

  //mostrando a media calculada das notas
  escreva(aluno[1,3])
fimalgoritmo
```

Acessando todas as posições de uma matriz: usando uma estrutura repetição

```
var
  aluno:vetor[1..10,1..3] de real
  i:inteiro
inicio
  para i de 1 ate 10 faca
    //obtendo a primeira nota do I aluno
    leia(aluno[i,1])
    //obtendo a segunda nota do I aluno
    leia(aluno[i,2])
    //calculando e armazenando a média do I aluno
    aluno[i,3] <- (aluno[i,1] + aluno[i,2]) / 2
  fimpara
fimalgoritmo
```


Acessando todas as posições de uma matriz: usando duas estruturas de repetição

```
var
  aluno:vetor[1..10,1..3] de real
  i,j:inteiro
inicio
  para i de 1 ate 10 faca //controla a quantidade de alunos
    para j de 1 ate 2 faca //controla a quantidade de notas
      //obtendo cada nota de um aluno
      leia(aluno[i,j])
    fimpara
    //calculado a média de todos os alunos
    aluno[i,3] <- (aluno[i,1] + aluno[i,2])/2
  fimpara
fimalgoritmo
```


Exemplo 1

- Faça um algoritmo que preencha uma matriz 15x2 com números inteiros e mostre os números entre 15 e 20.

```
algoritmo "matriz.numeros.entre.15.e.20"
var
    n:vetor[1..15,1..2] de inteiro
    i,j:inteiro
inicio
    //preenche a matriz
    para i de 1 ate 15 faca
        para j de 1 ate 2 faca
            leia(n[i,j])
        fimpara
    fimpara
    //mostra os números entre 15 e 20
    para i de 1 ate 15 faca
        para j de 1 ate 2 faca
            se (n[i,j]>=15) e (n[i,j]<=20) entao
                escreva(n[i,j])
            fimse
        fimpara
    fimpara
finalgoritmo
```

Exemplo 2

- Faça um algoritmo que preencha uma matriz 20x5 com números inteiros e mostre os números positivos.

```
algoritmo "matriz.numeros.positivos"  
var  
    n:vetor[1..20,1..5] de inteiro  
    i,j:inteiro  
inicio  
    //preenche a matriz  
    para i de 1 ate 20 faca  
        para j de 1 ate 5 faca  
            leia(n[i,j])  
        fimpara  
    fimpara  
    //mostra os números positivos  
    para i de 1 ate 20 faca  
        para j de 1 ate 5 faca  
            se (n[i,j]>0) entao  
                escreva(n[i,j])  
            fimse  
        fimpara  
    fimpara  
fimalgoritmo
```


Exemplo 3

- Faça um algoritmo que preencha uma matriz 10x10 com números inteiros e mostre quantos números são positivos.

```
algoritmo "matriz.qtos.numeros.positivos"
var
  n:vetor[1..10,1..10] de inteiro
  i,j,cont:inteiro
inicio
  //preenche a matriz
  para i de 1 ate 10 faca
    para j de 1 ate 10 faca
      leia(n[i,j])
    fimpara
  fimpara
  //conta quantos números são positivos
  cont <- 0 //contador
  para i de 1 ate 10 faca
    para j de 1 ate 10 faca
      se (n[i,j]>0) entao
        cont <- cont + 1
      fimse
    fimpara
  fimpara
  //informa quantos sao positivos
  escreva(cont)
finalgoritmo
```

Exemplo 4

- Faça um algoritmo que preencha uma matriz 8x5 com números inteiros e mostre quantos números são divisíveis por 4.

```
algoritmo "matriz.qtos.numeros.divisiveis.4"  
var  
  n:vetor[1..8,1..5] de inteiro  
  i,j,cont:inteiro  
inicio  
  //preenche a matriz  
  para i de 1 ate 8 faca  
    para j de 1 ate 5 faca  
      leia(n[i,j])  
    fimpara  
  fimpara  
  //conta quantos números são divisíveis por 4  
  cont <- 0 //contador  
  para i de 1 ate 8 faca  
    para j de 1 ate 5 faca  
      se (n[i,j] mod 4 = 0) entao  
        cont <- cont + 1  
      fimse  
    fimpara  
  fimpara  
  //informa quantos são divisíveis por 4  
  escreva(cont)  
fimalgoritmo
```


Exemplo 5

- Faça um algoritmo que receba 3 notas de um grupo de 50 alunos, calcule e mostre a média de cada aluno.

```
algoritmo "3.notas.media.50.alunos"  
var  
    nota:vetor[1..50,1..4] de real //na 1° 2° e 3° colunas as notas e  
                                   //na 4° coluna a média das 3 notas  
    i:inteiro  
inicio  
    //obtendo as 3 notas  
    para i de 1 ate 50 faca  
        leia(nota[i,1])  
        leia(nota[i,2])  
        leia(nota[i,3])  
    fimpara  
    //calculando a média a partir de 3 notas  
    para i de 1 ate 50 faca  
        nota[i,4] <- (nota[i,1]+nota[i,2]+nota[i,3])/3  
    fimpara  
    //mostrando a média calculada  
    para i de 1 ate 50 faca  
        escreva(nota[i,4])  
    fimpara  
fimalgoritmo
```

Resolvendo 1

- Faça um algoritmo que preencha uma matriz 3x4, calcule e mostre:
 - A quantidade de elementos pares.
 - A média de todos os elementos.

Atividades

- Listas 5 e 6.