



Fundamentos de Programação

Matrizes – Vetores Multidimensionais

Estruturas de dados matriciais

- Assim como os vetores, as matrizes são estruturas de dados **homogêneas**. Podem ser construídas dos diversos tipos básicos primitivos (real, inteiro, caractere).
- Principal diferença em relação aos vetores (unidimensionais): possui uma ou mais dimensões adicionais.
- Maioria dos casos: utiliza-se matrizes bidimensionais.

Matrizes

- São utilizadas quando os dados homogêneos necessitam de uma estruturação com mais de uma dimensão.
- Exemplo:
 - Programar um jogo de xadrez
 - o tabuleiro é naturalmente bidimensional



Matrizes

- São utilizadas quando os dados homogêneos necessitam de uma estruturação com mais de uma dimensão.
- Exemplo:
 - Imagens



Matrizes

- São utilizadas quando os dados homogêneos necessitam de uma estruturação com mais de uma dimensão.
- Exemplo:
 - Cesta de compras

Pão	Leite	Açúcar	Sal	Arroz	
3	1	0	0	0	Venda 1
0	3	1	0	1	Venda 2
7	0	1	0	0	Venda 3
0	0	0	1	2	Venda 4

Matrizes

- São utilizadas quando os dados homogêneos necessitam de uma estruturação com mais de uma dimensão.
- Exemplo:
 - Cesta de compras (Venda diária – 3 dimensões)

Pão	Leite	Açúcar	Sal	Arroz	
3	0	0	0	0	
0	0	0	0	0	
7	1	0	0	0	
0	10	4	0	0	
Pão	Leite	Açúcar	Sal	Arroz	
7	1	0	0	0	Venda 1
1	0	1	0	0	Venda 2
0	0	2	0	0	Venda 3
0	0	0	1	1	Venda 4

Matrizes: declaração

- A sintaxe para declaração de uma variável deste tipo é semelhante a declaração dos vetores.
- Considera-se porém a quantidade de elementos da outra dimensão:

`<tipo_primitivo>[,] <identificador> = new <tipo_primitivo>[numLinhas,numColunas];`

- Exemplo:

```
int[,] mat = new int[3,4];  
// matriz 3 linhas e 4 colunas do tipo inteiro  
// esta matriz tem 12 elementos
```

OBS: Os índices variam de 0 a 2 para as linhas e de 0 a 3 para as colunas neste caso.

Matrizes: declaração

matriz n-dimensional

- Exemplo:

```
double[, ,] mat = new double[3, 6, 5];  
// matriz real 3x6x5  
// esta matriz tem 90 elementos
```

*OBS: Os índices variam de 0 a 2 para a 1ª dimensão,
0 a 5 para a 2ª dimensão e
0 a 4 para a 3ª dimensão.*

Matrizes: declaração

Representação:

	0	1	2	3
0	0,0	0,1	0,2	0,3
1	1,0	1,1	1,2	1,3
2	2,0	2,1	2,2	2,3

mat[1,2]

Matrizes: atribuição

- A atribuição a um valor na matriz é feito explicitando a posição em que o valor será armazenado.
- Exemplo:

```
double [, ] num = new double [2, 3];
```

```
num [0, 0] = 3.6;
```

```
num [0, 1] = 2.7;
```

```
num [0, 2] = 1.5;
```

```
num [1, 0] = 5.0;
```

```
num [1, 1] = 4.1;
```

```
num [1, 2] = 2.3;
```



3.6	2.7	1.5
5.0	4.1	2.3

Matrizes: manipulação

- Os elementos das matrizes são manipulados individualmente por meio de índices (iniciando de zero) entre colchetes.

Exemplo:

	0	1	2
0	3	8	5
1	9	2	1

Acesso aos elementos da matriz acima:

```
a = mat[0,0];  
b = mat[1,2];
```

*A instrução ao lado atribui um valor ao elemento da linha zero e coluna um da matriz **mat**:*

```
i = 0;  
j = 1;  
mat[0,1] = 15;  
// ou  
mat[i,j] = 15;
```

Matrizes: exemplo

- O programa a seguir, inicializa com zeros os elementos de uma matriz inteira **n** de 5 linhas e 4 colunas e imprime.

```
static void Main(string[] args)
{
    int i, j;
    int[,] n = new int[5,4];
    for (i=0; i< 5; i++)
        for (j=0; j< 4; j++)
            n[i,j] = 0;

    Console.Write("{0}\n", "Matriz");
    for (i=0; i < 5; i++)
    {
        Console.WriteLine("\nLinha {0}\n", i);
        for (j=0; j < 4; j++)
            Console.Write("{0} ", n[i,j]);
    }
}
```

```
Matriz
Linha 0
0 0 0 0
Linha 1
0 0 0 0
Linha 2
0 0 0 0
Linha 3
0 0 0 0
Linha 4
0 0 0 0
```

Matrizes: exemplo

- O programa abaixo inicializa os elementos de uma matriz **m** com os valores iguais a soma dos índices de cada elemento e imprime cada valor.

```
static void Main(string[] args)
{
    int i, j;
    int[,] m = new int[3, 2];
    for (i=0; i < 3; i++)
        for (j=0; j < 2; j++)
        {
            m[i, j] = i + j;
            Console.Write("i={0} j={1} elemento={2}\n", i, j, m[i, j]);
        }
}
```

```
i= 0 j= 0 elemento=0
i= 0 j= 1 elemento=1
i= 1 j= 0 elemento=1
i= 1 j= 1 elemento=2
i= 2 j= 0 elemento=2
i= 2 j= 1 elemento=3
```

Matrizes e sub-rotinas

- Matrizes serão passadas para sub-rotinas da mesma forma como vetores.
- A propriedade Length retorna o número total de elementos em todas as dimensões da matriz. Dessa forma, é aconselhável passarmos por parâmetro o tamanho de cada dimensão da matriz.
- Um procedimento para imprimir uma matriz teria a seguinte declaração:

```
static void imprimeMatriz (double[, ] mat, int linhas, int colunas)
```

Exercício resolvido

- Problema: Criar uma função que receba uma matriz 2 x 3 de números reais e retorne a média dos valores da matriz. Crie uma função principal que chame a sub-rotina e imprima a média.
- Solução:
 - Todos os valores da matriz serão acumulados em uma variável real. A função retornará o valor dessa variável dividido pelo número de elementos de matriz.

Exercício - Solução

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }
    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3]{{3.4,5.6,4.0},{2.0,1.1,4.9}};
    double media = mediaMatriz(mat);
    Console.WriteLine("Media = {0}", media);
}
```


Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};
    double media = mediaMatriz(mat);
    Console.WriteLine("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:
i =
j =
media =

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[, ] m)
```

```
{  
    int i, j;  
    double media = 0;  
    for (i = 0; i < 2; i++)  
    {  
        for (j = 0; j < 3; j++)  
        {  
            media = media + m[i,j];  
        }  
    }  
    return media / 6.0;  
}
```

```
static void Main(string[] args)
```

```
{  
    double[, ] mat = new double[2,3] {{3.4,5.6,4.0},  
                                       {2.0,1.1,4.9}};  
    double media = mediaMatriz(mat);  
    Console.Write ("Media = {0}", media);  
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i =
j =
media =

		0	1	2
0		3.4 4	5.6	4.0
1		2.0 0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }
    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};
    double media = mediaMatriz(mat);
    Console.WriteLine("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = ?
j = ?
media = 0

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = 0
j = ?
media = 0

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = 0
j = 0
media = 0

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = 0
j = 0
media = 0 + 3.4 = 3.4

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:
i = 0
j = 1
media = 3.4

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = 0
j = 1
media = 3.4 + 5.6 = 9.0

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:
i = 0
j = 2
media = 9.0

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = 0
j = 2
media = 9.0 + 4.0 = 13.0

		0	1	2
0		3.4 4	5.6	4.0
1		2.0 0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:
i = 0
j = 3
media = 13.0

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = 1
j = 3
media = 13.0

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:
i = 1
j = 0
media = 13.0

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = 1
j = 0
media = 13.0 + 2.0 = 15.0

		0	1	2
0		3.4 4	5.6	4.0
1		2.0 0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:
i = 1
j = 1
media = 15.0

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = 1
j = 1
media = 15.0 + 1.1 = 16.1

		0	1	2
0		3.4 4	5.6	4.0
1		2.0 0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = 1
j = 2
media = 16.1

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = 1
j = 2
media = 16.1 + 4.9 = 21

		0	1	2
0		3.4 4	5.6	4.0
1		2.0 0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:
i = 1
j = 3
media = 21

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:

i = 2
j = 3
media = 21

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }
    return media / 6.0;
}
```

```
static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                       {2.0,1.1,4.9}};
    double media = mediaMatriz(mat);
    Console.Write ("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:
i = 2
j = 3
media = 21

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Exercício – Teste de Mesa

```
static double mediaMatriz(double[,] m)
{
    int i, j;
    double media = 0;
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 3; j++)
        {
            media = media + m[i,j];
        }
    }

    return media / 6.0;
}

static void Main(string[] args)
{
    double[,] mat = new double[2,3] {{3.4,5.6,4.0},
                                      {2.0,1.1,4.9}};

    double media = mediaMatriz(mat);
    Console.WriteLine("Media = {0}", media);
}
```

Entrada:
matriz com 6 elementos
2 linhas e 3 colunas

Variáveis da Sub-Rotina:
i = 2
j = 3
media = 21

		0	1	2
0		3.4	5.6	4.0
1		2.0	1.1	4.9

Saída:
Media = 3.5

Exercícios

1) Quais são os elementos do vetor referenciados pelas expressões ?

MAT

3.2	4.1	2.7
5.9	0.6	9.0
1.1	8.3.	6.4

a) `mat[2, 0];`

b) `mat[1, 1];`

c) `mat[3, 1];`

2) Qual é a diferença entre os números “3” das duas instruções abaixo ?

```
int[, ] mat = new int[6, 3];
```

```
mat[6, 3] = 5;
```

Exercícios

- 3) Faça um programa para exibir a soma de duas matrizes quadradas 3 x 3. Deverá ser criado um procedimento para ler uma matriz (será chamado duas vezes com parâmetros diferentes) e um segundo procedimento que irá imprimir a soma das matrizes passadas como parâmetro.
- 4) Faça um programa que leia uma matriz quadrada de tamanho 10 e um procedimento que inverta as linhas pelas colunas em uma segunda matriz de mesmo tamanho. Imprima ao final a segunda matriz.
- 5) Faça um procedimento que receba uma matriz quadrada 5 x 5 e crie uma matriz identidade. Imprima a matriz após sua inicialização em um outro procedimento.

Exercícios

- 6) Crie uma função que receba uma matriz de números reais e retorne a soma dos elementos desta matriz.
- 7) Faça um programa que leia um vetor de dimensão 5 e uma matriz quadrada de dimensão 5. Crie um procedimento que multiplique o vetor pela matriz. Imprima o resultado.
- 8) Faça um programa que leia duas matrizes quadradas de dimensão 3. Crie um procedimento que multiplique as duas matrizes. Imprima o resultado.

Matrizes

Vetores multidimensionais

Aula de Exercícios

Declaração de matrizes

```
int[, ] mat = new int[3, 4];
```

	0	1	2	3
0	0,0	0,1	0,2	0,3
1	1,0	1,1	1,2	1,3
2	2,0	2,1	2,2	2,3

mat[1,2] →

- Declaração de matrizes com mais de 2 dimensões

```
double[, , ] mat2 = new double[3, 5, 2];
```

- Operações em matrizes

```
mat2[0, 0] = 7;
```

```
mat2[2, 3] = mat2[2, 3] * 8.6;
```

Inicialização de matrizes

- Pode-se fornecer valores de cada elemento de uma matriz na declaração, da mesma forma que nos vetores.

- Exemplo:

```
double[, ] num = new double[2, 3] = { { 3.6, 2.7, 1.5 },  
                                       { 5.0, 4.1, 2.3 } };
```

- Ou seja, fornece-se os valores linha a linha.

Inicialização de matrizes

- Observação:
- Em C# as matrizes são inicializadas automaticamente com o valor padrão para o tipo de dado: 0 para tipos numéricos ou o caractere nulo ('\0') para o tipo char.

Matrizes: exemplo

- O programa abaixo inicializa os elementos de uma matriz **m** com os valores iguais a soma dos índices de cada elemento e imprime cada valor.

```
static void Main(string[] args)
{
    int i, j;
    int[,] m = new int[3, 2];
    for (i=0; i < 3; i++)
        for (j=0; j < 2; j++)
        {
            m[i, j] = i + j;
            Console.WriteLine("i={0} j={1} elemento={2}\n", i, j, m[i, j]);
        }
}
```

```
i= 0 j= 0 elemento=0
i= 0 j= 1 elemento=1
i= 1 j= 0 elemento=1
i= 1 j= 1 elemento=2
i= 2 j= 0 elemento=2
i= 2 j= 1 elemento=3
```

Matrizes e Sub-rotinas

- Matrizes serão passadas para subrotinas da mesma forma como os vetores são passados, a menos de um detalhe: **os tamanhos de cada dimensão, se não previamente definidos, precisam ser passados por parâmetro.**

- **Exemplo:**

```
void imprimeMatriz(double [, ] m, int linhas, int colunas)
```

Exercícios

- 1) Faça um procedimento para calcular a multiplicação de uma matriz 3×4 por um escalar. Faça também um procedimento capaz de imprimir esta matriz. Ao final, desenvolva uma função principal onde será criada e lida uma matriz 3×4 . Faça nesta função a chamada dos dois procedimentos criados anteriormente.
- 2) Faça um programa que leia uma matriz 7×5 e imprima a terceira coluna.
- 3) Faça um programa que leia uma matriz quadrada de dimensão 10, uma função que encontre o maior valor desta matriz e uma função que encontre o menor valor. Imprima os valores encontrados na função principal.
- 4) Faça um programa que leia uma matriz 6×3 e um procedimento que gere duas matrizes 3×3 , a primeira com as 3 primeiras linhas e a outra com as restantes.

Exercícios

- 5) Faça um programa que leia uma matriz de caracteres 5 x 10 e uma função que conte quantas letras “a” aparecem na matriz. Você poderá fazer a leitura letra a letra ou considerar que cada linha da matriz é uma *string*.
- 6) Faça um programa que leia 3 vetores reais de dimensão 10 e um procedimento que gere uma matriz (10 x 3) onde cada coluna é dada por um destes vetores.
- 7) Crie uma matriz tridimensional onde as linhas indicam as notas de matemática, história e geografia em três provas de 10 alunos e crie uma função que verifique quantos alunos passaram, ou seja, os que tenham média aritmética > 60 nas 3 disciplinas.

Exercícios

8) Faça um programa para ler a quantidade de um total de 5 produtos que uma empresa tem em suas 7 lojas e imprimir em uma tabela:

- a) o total de cada produto nestas lojas
- b) a loja que tem menos produtos

9) Idem ao exercício 1, agora com o seguinte “Menu”:

Escolha uma opção de cálculo para matrizes:

1) Soma

2) Diferença

3) Transposta

4) Produto

5) Sair

Opção: ____

Na opção (3) o usuário só precisa fornecer uma matriz, nas opções (1), (2) e (4) o usuário deve fornecer duas matrizes.

Observação, a matriz deve ser 3x3.