

Cap. 8 - Vetores e Matrizes

Aprenda a Programar com C# - 2ª Edição (2020)

Edições Sílabo

<https://bit.ly/36nyKFm>

António Trigo, Jorge Henriques

{antonio.trigo,jmvhenriques}@gmail.com

1 de outubro de 2020

Vetores

Exemplos

Matrizes

Exemplos

Vetores e funções

Definição

- ▶ Os vetores armazenam num só identificador vários valores do mesmo tipo;
- ▶ É um bloco consecutivo de espaços na memória contendo valores do mesmo tipo e referenciado com um nome
- ▶ Cada um dos elementos do vetor é identificado através do nome do vetor e da posição do elemento no vetor;
- ▶ Sintaxe:

```
tipo_de_dados[] nome;
```

- ▶ Vetor “a” de inteiros

```
int[] a;
```

Inicialização

- ▶ Em C# os vetores são objetos pelo que devem ser inicializados com a palavra “new”. Caso se queiram atribuir valores iniciais ao vetor pode-se omitir a parte `new int[]`.

```
tipo_de_dados[] nome = new tipo_de_dados[dimensao];  
tipo_de_dados[] nome = new  
    tipo_de_dados[dimensao]{valor1, valor2, ..., valorn};  
tipo_de_dados[] nome = new  
    tipo_de_dados[] {valor1, valor2, ..., valorn};  
tipo_de_dados[] nome = {valor1, valor2, ..., valorn};
```

- ▶ Exemplo de um vetor inicializado com 5 valores:
`int[] a = new int[5];`
- ▶ Exemplo de um vetor inicializado com os seguintes 5 valores iniciais 10, 20, 30, 40, 50:

```
int[] a = new int[5]{10, 20, 30, 40, 50};
```

- ▶ Os valores do vetor que não forem explicitamente inicializados, serão inicializados a 0.

Utilização

- ▶ Para aceder a cada elemento do vetor é preciso indicar a sua posição, sendo que em C# todos os vetores começam pela posição 0.
- ▶ Exemplo de impressão no ecrã do elemento na quarta posição:

```
int[] a = new int[]{10,20,30,40,50};  
Console.WriteLine(a[3]); //Mostra o elemento 40
```

- ▶ Exemplo de leitura a partir do teclado de um valor para a terceira posição:

```
int[] a = new int[]{10,20,30,40,50};  
Console.Write("Insira um valor para a terceira  
posicao:");  
a[2] = Convert.ToInt32(Console.ReadLine());  
Console.WriteLine(a[2]);
```

Apresentar todos os valores de um vetor

```
static void Main(string[] args)
{
    int TAM = 10, i;
    int[] v = {23,24,25,26,27,13,14,15,16,17};
    for(i=0;i<TAM;i++){
        Console.WriteLine(v[i]);
    }
}
```

Apresentar todos os valores de um vetor, utilizando a propriedade Length

```
static void Main(string[] args)
{
    int i;
    int[] v = { 23, 24, 25, 26, 27, 13, 14, 15, 16,
               17 };
    for (i = 0; i < v.Length; i++)
    {
        Console.WriteLine(v[i]);
    }
}
```

Apresentar todos os valores de um vetor, utilizando a estrutura *foreach*

```
static void Main(string[] args)
{
    int[] v = {23,24,25,26,27,13,14,15,16,17};
    foreach (int num in v)
    {
        Console.WriteLine(num);
    }
}
```


Apresentar todos os valores de um vetor que são superiores a 100

```
static void Main(string[] args)
{
    int TAM = 10, i;
    int[] v = {23,24,125,26,127,13,214,15,16,217};
    for(i=0;i<TAM;i++){
        if (v[i]>100)
            Console.WriteLine(v[i]);
    }
}
```

Preencher um vetor com valores introduzidos pelo utilizador

```
static void Main(string[] args)
{
    int TAM = 10, i;
    int[] v = new int[10];
    for(i = 0; i < TAM ; i++){
        Console.Write("Insira um numero para  
v[{0}]:", i);
        v[i] = Convert.ToInt32(Console.ReadLine());
    }
    for (i = 0; i < TAM; i++)
    {
        Console.WriteLine(v[i]);
    }
}
```

Preencher e apresentar um vetor com n valores introduzidos pelo utilizador

```
static void Main(string[] args)
{
    int TAM, i;
    Console.WriteLine("Quantos valores deseja
        inserir?");
    TAM = Convert.ToInt32(Console.ReadLine());
    int[] v = new int[TAM];
    for(i = 0; i < TAM ; i++){
        Console.Write("Insira um numero para
            v[{0}]:", i);
        v[i] = Convert.ToInt32(Console.ReadLine());
    }
    for (i = 0; i < TAM; i++)
    {
        Console.WriteLine(v[i]);
    }
}
```

Definição

- ▶ Vetor é uma designação para o caso particular da matriz que só possui uma dimensão. Nos outros casos utilizamos a designação Matriz;
- ▶ As matrizes como duas ou mais dimensões designam-se de matrizes multidimensionais;
- ▶ A semelhança dos vetores ou matrizes unidimensionais, cada elemento é identificado com um nome e a sua posição na matriz;
- ▶ Sintaxe:

```
tipo_de_dados[,] nome;
```

- ▶ Matriz “a” de inteiros

```
int[] a;
```

Inicialização

- ▶ É similar à dos vetores, mas com a indicação das várias dimensões..

```
tipo_de_dados[,] nome = new
    tipo_de_dados[dimensao1, dimensao2, ..., dimensaoN];
tipo_de_dados[,] nome = new
    tipo_de_dados[dimensao1, dimensao2, ..., dimensaoN]{valor1
tipo_de_dados[,] nome = new
    tipo_de_dados[, ..., ]{{valor1, valor2, ..., valorn}, ..., {va
tipo_de_dados[,] nome =
    {{valor1, valor2, ..., valorn}, ..., {valor1, valor2, ..., valo
```

- ▶ Exemplo de uma matriz 3x4:

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

- ▶ Exemplo de uma matriz 3x2 com os seguintes 6 valores iniciais 1, 2, 3, 4, 5 e 6:

```
int[,] m = new int[3, 2] { {1, 2}, {3, 4}, {5, 6} };
```

Utilização

- ▶ A numeração dos elementos, à semelhança dos vetores começa sempre na posição 0.
- ▶ Exemplo de impressão no ecrã do elemento na posição 2,1 de uma matriz 3,2:

```
int[,] m = new int[3, 2] { {1, 2}, {3, 4}, {5, 6} };  
Console.WriteLine(m[2, 1]); //Mostra o elemento 6
```

- ▶ Exemplo de leitura a partir do teclado de um valor para o primeiro elemento (0,0):

```
int[,] m = new int[3, 2] { { 1, 2 }, { 3, 4 }, { 5,  
    6 } };  
Console.Write("Insira um valor para a primeira  
    posicao:");  
m[0, 0] = Convert.ToInt32(Console.ReadLine());  
Console.WriteLine(m[0, 0]);
```

Apresentar todos os valores de uma matriz bidimensional

```
static void Main(string[] args)
{
    int L = 2, C = 3, i, j;
    int[,] m = { { 1, 2, 3 }, { 4, 5, 6 } };
    for (i = 0; i < L; i++)
    {
        for (j = 0; j < C; j++)
            Console.Write("{0} ", m[i, j]);
        Console.WriteLine();
    }
}
```

Passagem por valor de vetores

```
static void Alterar(int[] pArray){  
    pArray[0] = 888; // Esta alteracao muda o valor  
                    original do vetor  
    pArray = new int[5] {-3, -1, -2, -3, -4}; //  
                    Esta alteracao e local  
    System.Console.WriteLine("Dentro deste metodo o  
                             valor de arr[0] e: {0}", pArray[0]);  
}  
  
static void Main() {  
    int[] arr = {1, 4, 5};  
    System.Console.WriteLine("Valor do primeiro  
                             elemento do vetor antes de ser chamado o metodo:  
                             {0}", arr [0]);  
    Alterar(arr);  
    System.Console.WriteLine("Valor do primeiro  
                             elemento do vetor depois de ser chamado o  
                             metodo: {0}", arr [0]);  
}
```


Passagem por referência de vetores

```
static void Alterar(ref int[] pArray){  
    // No caso da passagem por referencia ambas as  
    // instrucoes irao afetar as variaveis originais:  
    pArray[0] = 888;  
    pArray = new int[5] {-3, -1, -2, -3, -4};  
    System.Console.WriteLine("Dentro deste metodo o  
        valor de arr[0] e: {0}", pArray[0]);  
}  
  
static void Main() {  
    int[] arr = {1, 4, 5};  
    System.Console.WriteLine("Valor do primeiro  
        elemento do vetor antes de ser chamado o metodo:  
        {0}", arr [0]);  
    Alterar(arr);  
    System.Console.WriteLine("Valor do primeiro  
        elemento do vetor depois de ser chamado o  
        metodo: {0}", arr [0]);  
}
```