

Definição de Vetor e Matriz em Linguagem C#

As matrizes em geral são caracterizadas por se tratarem de uma única variável de um determinado tamanho que armazena várias informações do mesmo tipo. Essas informações são gravadas na memória seqüencialmente e são referenciadas através de índices. As matrizes podem ser tanto unidimensionais (vetores) como multidimensionais.

Matrizes Unidimensionais

São matrizes de uma única dimensão. Essas matrizes também são chamadas de **vetores**. A declaração de vetores em C deve obedecer a seguinte sintaxe:

```
Tipo [ ] nome_vetor = new Tipo[tamanho];
```

O tipo deve ser especificado de acordo com o tipo de informação que será armazenado no vetor (ex. int float, char, string...). E o tamanho representa a quantidade de elementos que este vetor irá conter. É importante dizer que na linguagem C# as matrizes começam pelo índice 0 que guarda o primeiro elemento da matriz. Para entender melhor, considere que seja necessário declarar um vetor do tipo inteiro que contenha 10 elementos. Isto é feito da seguinte forma:

```
int [ ] vetor_exemplo = new int[10];
```

Isso por que a matriz "vetor_exemplo" vai de 0 a 9, ou seja, contém 10 elementos. Também é possível inicializar o vetor no momento de sua declaração. Para isso veja a sintaxe abaixo:

```
Tipo [ ] nome_vetor = { lista_de_valores };
```

Sendo que todos os elementos da lista de valores devem ser separados por vírgula e serem todas do mesmo tipo de dados especificado. A seguir temos a declaração do "vetor_exemplo" com os valores atribuídos.

```
Int [ ] vetor_exemplo = { 0,1,2,3,4,5,6,7,8,9 };
```

Exemplo 23

/* Exemplo utilizando vetor do tipo int */

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            int[] valores = new int[10];
            Console.Clear();
            for (i = 0; i < valores.Length; ++i) // Atribuição
                valores[i] = i + 1;

            for (i = 0; i < valores.Length; ++i) // Exibição
                Console.WriteLine("valores[" + i + "] = " + valores[i]);

            Console.ReadKey();
        }
    }
}
```

Exemplo 24

/* Exemplo utilizando vetor do tipo double */

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            double sal_minimo = 510.00;
            double[] salarios = new double[10];
            Console.Clear();

            for (i = 0; i < salarios.Length; ++i) // Atribuição
                salarios[i] = (i+1) * sal_minimo;

            for (i = 0; i < salarios.Length; ++i) // Exibição
                Console.WriteLine("Salario Minimo[{0}] = {1}", (i + 1),
                    salarios[i].ToString("#0.00"));

            Console.ReadKey();
        }
    }
}
```

Exemplo 25

/* Exemplo utilizando vetor do tipo string */

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            string[] nomes = new string[10];

            Console.Clear();

            for (i = 0; i < nomes.Length; ++i) // Atribuição
            {
                Console.Write("Digite o {0}º nome: ", i+1);
                nomes[i] = Console.ReadLine();
            }

            for (i = 0; i < nomes.Length; ++i) // Exibição
                Console.WriteLine("Nomes[{0}] = {1}", i, nomes[i]);

            Console.ReadKey();
        }
    }
}
```

Exercícios

- 1) Escreva um programa que leia um vetor de 15 números inteiros e exiba ao final apenas os números pares que foram armazenados neste vetor.
- 2) Escreva um programa que leia um vetor de 15 números inteiros e exiba ao final apenas a quantidade de números ímpares que foram armazenados neste vetor.
- 3) Escreva um programa que leia um vetor de 15 notas de alunos de uma turma e exiba ao final a média das notas.
- 4) Escreva um programa que leia um vetor de 15 notas de alunos de uma turma e exiba ao final a média e a quantidade de alunos que estão com nota acima da média e a quantidade de alunos que estão com nota abaixo da média.
- 5) Escreva um programa que leia o nome e o preço de 10 produtos. Logo após cadastros os produtos, peça para o usuário digitar um valor, no qual deverá ser realizada uma pesquisa e exibir apenas os produtos que possuem preço até o valor digitado pelo usuário.
- 6) Escreva um programa que cadastre o nome e a nota de 20 alunos. Ao final o programa deverá exibir os dados dos alunos e a informação de foi "Aprovado" ou "Reprovado", considerando nota sete para aprovação.