



Programação Estruturada e Orientada a Objetos

REVISÃO

2013

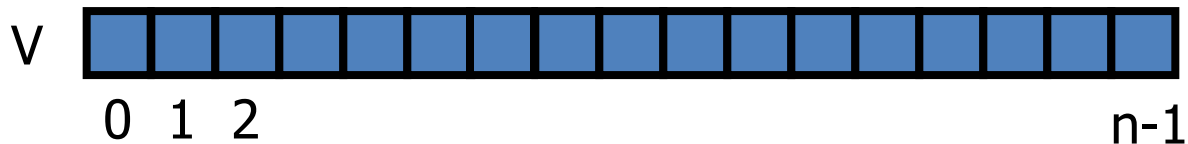
O que veremos hoje?

- Introdução
- Revisão arrays e matrizes
- Exercícios

Transparências baseadas no material do
Prof. Gilbert Azevedo

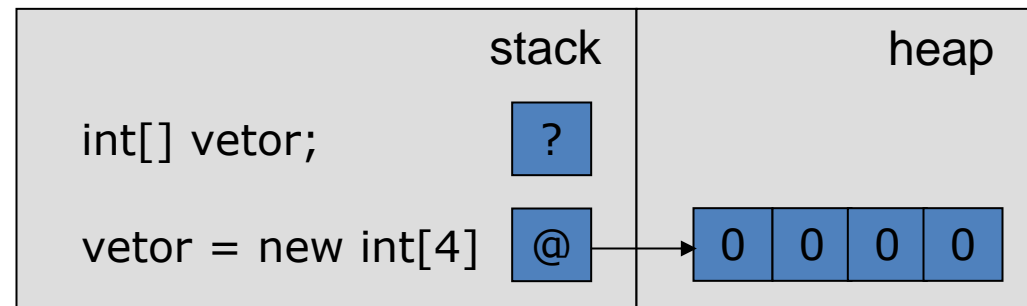
Vetor (Array)

- Tipo de dados utilizado para representar uma coleção de valores de um mesmo tipo
- Uma variável do tipo vetor armazena diversos valores que são referenciados por um número denominado de *índice*
- O *índice* (index) indica a posição de um elemento (valor) dentro do vetor
- Em C#, o *índice* inicia em zero



Declarando e Instanciando Arrays

- Declarando variáveis de arrays
 - É necessário definir o tipo do elemento, seguido de um par de chaves e de uma variável para referenciar o vetor
 - Ex: Declaração de um vetor de inteiros
 - `int[] vetor;`
- Criando instâncias
 - O operador *new* é usado para alocar o vetor na memória
 - Ex: Declarando e alocando um vetor com 4 inteiros
 - `int[] vetor = new int[4];`



Iniciando os Elementos do Array

- Os elementos do array podem ser iniciados na criação do vetor
 - `int[] vetor = new int[4] { 1, 2, 3, 4 };`
 - `int[] vetor = { 1, 2, 3, 4 };`
- Vetor com elementos aleatórios
 - `Random r = new Random();`
 - `int[] vetor = new int[4] { r.Next(10), r.Next(10), r.Next(10), r.Next(10) };`

Acessando os Elementos do Array

- Os elementos são acessados através do operador de indexação []
- O índice dos elementos inicia em zero
- A exceção `IndexOutOfRangeException` é levantada se um índice inválido é usado
- Recuperando o valor de um elemento
 - `int[] vetor = new int[4] { 1, 2, 3, 4 };`
 - `int total = vetor[0] + vetor[1] + vetor[2] + vetor[3];`
- Atribuindo valor a um elemento
 - `vetor[0] = vetor[1] = vetor[2] = vetor[3] = 0;`

Iterando em um Array

- Iteração com repetição for, while, do-while
 - A propriedade Length retorna o número de elementos
 - `for (int i = 0; i < vetor.Length; i++)`
 - `Console.WriteLine(vetor[i]);`
- Iteração com foreach
 - `foreach (int i in vetor)`
 - `Console.WriteLine(i);`

Copiando Arrays

- O operador de atribuição realiza apenas uma cópia da referencia
 - `int[] vetor = { 1, 2, 3, 4 };`
 - `int[] alias = vetor;`
- O método *CopyTo* pode ser utilizado para realizar uma cópia dos elementos
 - `int[] vetor = new int[4] { 1, 2, 3, 4 };`
 - `int[] copy = new int[4];`
 - `vetor.CopyTo(copy, 0);`
- Copiando com o método estático *System.Array.Copy*
 - `Array.Copy(vetor, copy, 4);`

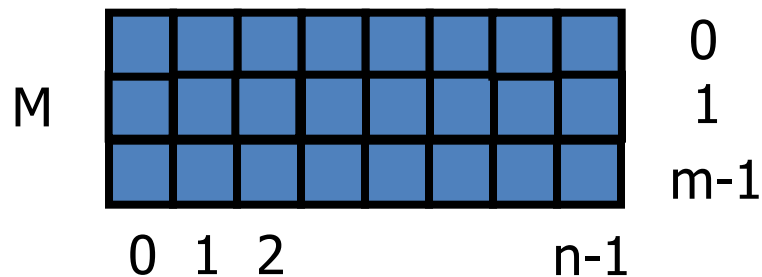
Exemplo de Algoritmo com Vetor

- Mostra um vetor de 10 inteiros na ordem inversa

```
static void Main(string[] args)
{
    int[] x = new int[10];
    int i;
    Console.WriteLine("Digite 10 valores inteiros");
    for (i = 0; i < 10; i++)
        x[i] = int.Parse(Console.ReadLine());
    Console.WriteLine("Ordem inversa");
    for (i = 9; i >= 0; i--)
        Console.WriteLine(x[i]);
    Console.ReadKey();
}
```

Matriz

- Tipo de dados utilizado para representar uma coleção de valores de duas dimensões composta por elementos de um mesmo tipo
- Uma variável do tipo matriz armazena diversos valores que são referenciados por dois índices
- O primeiro índice representa a linha do elemento e o segundo, a coluna
- Os índices de linha e coluna iniciam em zero



Declarando e Instanciando Matrizes

- Declarando variáveis de matrizes
 - É necessário definir o tipo do elemento, seguido de um par de chaves com uma vírgula e de uma variável para referenciar a matriz
 - Ex: Declaração de uma matriz de inteiros
 - `int[,] matriz;`
- Criando instâncias
 - O operador *new* é usado para alocar a matriz
 - Ex: Declarando e alocando uma matriz 3x4 de inteiros
 - `int[,] matriz = new int[3,4];`

Acessando os Elementos da Matriz

- Os elementos podem ser iniciados na criação da matriz
 - Matriz 3x4 de elementos inteiros
 - `int[,] x =`
`{ { 1, 2, 3, 4 }, { 5, 6, 7, 8 }, { 9, 10, 11, 12 } };`
 - `int[,] x = new int[3,4]`
`{ { 1, 2, 3, 4 }, { 5, 6, 7, 8 }, { 9, 10, 11, 12 } };`
- Os elementos são acessados através do operador de indexação [] com os índices de linha e coluna
 - `x[0,1] = 10;`
 - Atribui 10 ao elementos da 1ª linha, 2ª coluna
 - `int i = x[0,0] + x[0,1] + x[0,2] + x[0,3];`
 - Soma os elementos da 1ª linha da matriz x

Exemplo de Algoritmo com Matriz

- Mostra uma matriz 3x4 e sua transposta

```
static void Main(string[] args)
{
    int[,] x = new int[3,4] { { 1, 2, 3, 4 }, { 5, 6, 7, 8 }, { 9, 10, 11, 12 } };
    int i, j;
    Console.WriteLine("Matriz");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 4; j++)
            Console.Write("{0,5}", x[i, j]);
        Console.WriteLine();
    }
    Console.WriteLine("Transposta");
    for (i = 0; i < 4; i++) {
        for (j = 0; j < 3; j++)
            Console.Write("{0,5}", x[j, i]);
        Console.WriteLine();
    }
}
```

Exercícios

- 1. Ler um vetor com 10 inteiros e mostrar os números na ordem direta e inversa a que foram lidos.
- 2. Ler um vetor com 10 inteiros e calcular o maior e o menor elemento do vetor.
- 3. Ler uma matriz 4 x 4 e calcular a soma do maior com o menor elemento da matriz.
- 4. Ler uma matriz 4 x 4 e calcular a soma dos elementos da diagonal principal.
- 5. Ler duas matrizes 4 x 4 e calcular a soma das matrizes.
- 6. Ler duas matrizes 4 x 4 e calcular o produto entre as matrizes.

Dúvidas

