



# IFRN

Tecnologia em Análise e Desenvolvimento de  
Sistemas

---

## Arrays

Prof. Gilbert Azevedo



# Conteúdo

---

- Arrays
- Declaração e Instanciação
- Iniciação dos Elementos
- Acesso aos Elementos
- Iteração e Cópia



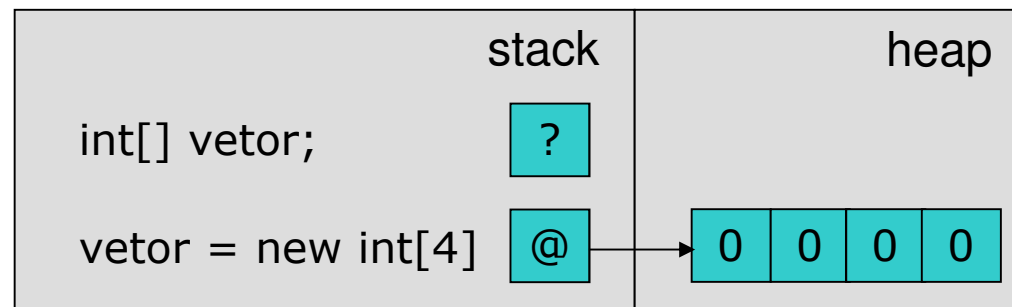
# Arrays em C#

---

- Array é uma seqüência não ordenada de elementos
- Todos os elementos são de um mesmo tipo
- Os elementos residem em um bloco contíguo de memória e são acessados através de um índice inteiro
- No C#, os arrays são tipos por referência
- Todos arrays são descendentes da classe `System.Array`

# 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].ToString());`
- Iteração com foreach
  - `foreach (int i in vetor)`
  - `Console.WriteLine(i.ToString());`
- Iteração com IEnumerator
  - `IEnumerator x = vetor.GetEnumerator();`
  - `while (x.MoveNext())`
  - `Console.WriteLine(x.Current.ToString());`



# 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 Arrays

---

- Copiando com o método estático *System.Array.Copy*
  - `Array.Copy(vetor, copy, 4);`
- Copiando com o método *Clone*, sem instanciar previamente
  - `int[] copy = (int [])vetor.Clone();`



# Referencias Bibliográficas

---

- Introduction to C# Programming with Microsoft .Net
  - Microsoft Official Course 2609A
- Microsoft Visual C# 2005 - Passo a passo
  - John Sharp, Bookman, 2007
- Microsoft Asp.Net – Passo a passo
  - George Sheperd, Bookman, 2007
- Microsoft VS 2005 Express Edition Documentation