





Programação Orientada a Objetos

Ms. Eng. Claudio Ferreira de Carvalho



AULA 09
Variáveis Indexadas

Variáveis indexadas

- ✓ Muitas vezes, um programa precisa receber diversos dados do mesmo tipo como, por exemplo, o nome e a idade de diversas pessoas.
- ✓ Para receber estes dados o programador precisa criar diversas variáveis. Neste caso, a tendência dos desenvolvedores é criar variáveis com o mesmo nome possuindo no final um número, tal como: nome1, nome2, nome3. etc.
- ✓ O exemplo a seguir demonstra este procedimento.

Variáveis indexadas

```
1. namespace Aula09_Ex01
2. {
3.     internal class Program
4.     {
5.
6.         static void Main(string[] args)
7.         {
8.             // Declara variáveis
9.             string nome1, nome2, nome3;
10.            byte idade1, idade2, idade3;
11.
12.            // Solicita os nomes e idade
13.            Console.WriteLine("Forneca o 1o nome: ");
14.            nome1 = Console.ReadLine();
15.            Console.WriteLine("Forneca a idade: ");
16.            idade1 = byte.Parse(Console.ReadLine());
17.            Console.WriteLine("Forneca o 2o nome: ");
18.            nome2 = Console.ReadLine();
19.            Console.WriteLine("Forneca a idade: ");
20.            idade2 = byte.Parse(Console.ReadLine());
21.            Console.WriteLine("Forneca o 3o nome: ");
22.            nome3 = Console.ReadLine();
23.            Console.WriteLine("Forneca a idade: ");
24.            idade3 = byte.Parse(Console.ReadLine());
25.
26.            // Fornece os nomes e idades
27.            Console.WriteLine("\nRelacao de nomes e idades");
28.            Console.WriteLine("Nome: {0} está com {1} anos", nome1, idade1);
29.            Console.WriteLine("Nome: {0} está com {1} anos", nome2, idade2);
30.            Console.WriteLine("Nome: {0} está com {1} anos", nome3, idade3);
31.
32.            // Mantém a tela aberta esperando a digitação de uma tecla
33.            Console.ReadKey();
34.        }
35.    }
36. }
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Projeto
Forneca o 1o nome: Claudio
Forneca a idade: 75
Forneca o 2o nome: Pedro
Forneca a idade: 70
Forneca o 3o nome: Antonio
Forneca a idade: 66

Relacao de nomes e idades
Nome: Claudio está com 75 anos
Nome: Pedro está com 70 anos
Nome: Antonio está com 66 anos
```

Conforme pode ser observado, a programação se torna muito trabalhosa e extensa, visto que o programa acima só recebe 3 nomes e 3 idades, imagine se a necessidade fosse receber 100 nomes. Para facilitar esta tarefa, o C#, assim como a maioria das linguagens de programação possui o que se chama de Variável Indexada ou arranjos . Estes índices podem ser unidimensionais ou multidimensionais

Variáveis indexadas Unidimensional

✓ Definição:

- ✓ Consiste em criar um conjunto de variáveis indexadas numericamente, deste 0 até o valor desejado.

✓ Sintaxe:

tipo[] identificador = new tipo:[dimensão];

tipo:

é o tipo de variável que será criada;

identificador:

é o identificador da variável;

dimensão:

número de itens que serão criados a partir do 0 até o número indicado.

Variáveis indexadas Unidimensional – Exemplo

```
1. namespace Aula09_Ex02
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             string[] nome = new string[10];
8.             int[] idade = new int[10];
9.             int c, f;
10.            char cont;
11.
12.            c = 0;
13.            // Solicita os nomes e idade
14.            do
15.            {
16.                // Declara variáveis
17.                Console.Write("Forneça o {0}o nome: ", c + 1);
18.                nome[c] = Console.ReadLine();
19.                Console.Write("Forneça a idade do {0}o: ", c + 1);
20.                idade[c] = int.Parse(Console.ReadLine());
21.                Console.Write("Deseja fornecer outro nome? (s/n) ");
22.                cont = char.Parse(Console.ReadLine().ToUpper());
23.                if (cont == 'N')
24.                    break;
25.                c = c + 1;
26.            }
27.            while (c <= 9);
```

```
28.
29.            // Atribui ao f o valor de c que encerrou o loop
30.            f = c;
31.            c = 0;
32.
33.            // Fornece os nomes e idades
34.            Console.WriteLine("\nRelação de nomes e idades:");
35.            do
36.            {
37.                Console.WriteLine("Nome {0} , está com {1} anos de
idade", nome[c], idade[c]);
38.                c = c + 1;
39.            }
40.            while (c <= f);
41.
42.            // Mantém a tela aberta esperando a digitação de uma
43.            tecla
44.            Console.ReadKey();
45.        }
46.    }
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Proj

```
Forneça o 1o nome: Claudio
Forneça a idade do 1o: 75
Deseja fornecer outro nome? (s/n) s
Forneça o 2o nome: Pedro
Forneça a idade do 2o: 70
Deseja fornecer outro nome? (s/n) s
Forneça o 3o nome: Antonio
Forneça a idade do 3o: 66
Deseja fornecer outro nome? (s/n) n

Relação de nomes e idades:
Nome Claudio , está com 75 anos de idade
Nome Pedro , está com 70 anos de idade
Nome Antonio , está com 66 anos de idade
```


Variáveis indexadas Unidimensional – Exemplo (discutir índices)

```
1. namespace Aula09_Ex03
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara variáveis
8.             double[] valor = new double[4];
9.             double soma = 0;
10.            double media = 0;
11.            byte c;
12.            // Solicita os valores
13.            for (c = 0; c <= 3; c++)
14.            {
15.                Console.WriteLine("Forneça o {0}o Valor: ", c + 1);
16.                valor[c] = double.Parse(Console.ReadLine());
17.                soma = soma + valor[c];
18.            }
19.            // Fornece a média
20.            media = soma / c;
21.            Console.WriteLine("A média dos {0} valores é {1} ", c,
media);
22.
23.            // Mantem a tela aberta aguardando a digitação de uma
tecla
24.            Console.ReadKey();
25.        }
26.    }
27. }
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_O
Forneça o 1o Valor: 10
Forneça o 2o Valor: 20
Forneça o 3o Valor: 30
Forneça o 4o Valor: 40
A média dos 4 valores é 25
```

Neste programa deve ser observado:

1) Quando é criada a variável indexada com a instrução:
`double[] valor = new double[4];`
são criadas 4 variáveis do tipo `double` com os seguintes identificadores:

`valor[0]`, `valor[1]`, `valor[2]`, `valor[3]`, portanto o primeiro valor fornecido é atribuído à variável `valor[0]` e o quarto valor fornecido à variável `valor[3]`, motivo pelo qual na linha `Console.WriteLine("Forneça o {0} o Valor: ", c + 1);`, o valor a ser atribuído à `{0}` é `c+1`

2) O "loop for", é executado 4 vezes com o contador `c` variando de 0 até 3, **mas o contador `c` é incrementado no final do loop, então, ao sair do loop o `c` vale 4**. Isto pode ser confirmado pois a média é obtida pela expressão
`media = soma / c;`

Posicionar o cursor na tela

- ✓ Posicionamento do cursor em coordenadas
 - ✓ A utilização dos comandos Write e WriteLine permitem escrever sequencialmente na tela.
 - ✓ É possível também posicionar o cursor com a instrução SetCursorPosition, seguido das coordenadas de tela.

- ✓ Sintaxe

`SetCursorPosition(c,l);`

c é o número da coluna;
l é o número da linha.

- ✓ Sintaxe

`SetCursorPosition(15,10);`

Coloca o cursor na coluna
15 e linha 10

Variáveis indexadas Multidimensional

✓ Definição:

- ✓ Consiste em criar um conjunto de variáveis indexadas numericamente com diversas dimensões.

✓ Sintaxe:

tipo[,] identificador = new tipo:[dimensão1,dimesnsão2];

tipo:

é o tipo de variável que será criada;

identificador:

é o identificador da variável;

dimensões:

número de itens que serão criados de cada uma das dimensões a partir do 0 até o número indicado.

Variáveis indexadas Multidimensional – Exemplo

```
1. namespace Aula09_Ex04
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração de Variáveis
8.             int[,] A = new int[2, 2];
9.             byte c, l;
10.
11.             // Recebe os valores da matriz
12.             for (l = 0; l <= 1; l++)
13.             {
14.                 for (c = 0; c <= 1; c++)
15.                 {
16.                     Console.Write("Forneça o valor
17. A {0}{1} ", l+1, c+1);
18.                     A[l, c] =
19.                     int.Parse(Console.ReadLine());
20.                 }
21.             }
22.         }
23.     }
24. }
```

```
20.
21.     //Monta a matriz
22.     // Barra esquerda
23.     for (l = 9; l <= 13; l++)
24.     {
25.         Console.SetCursorPosition(8, l);
26.         Console.WriteLine("|");
27.     }
28.     // Barra direita
29.     for (l = 9; l <= 13; l++)
30.     {
31.         Console.SetCursorPosition(18, l);
32.         Console.WriteLine("|");
33.     }
34.
35.     // Coloca os números na matriz
36.     Console.SetCursorPosition(10, 10);
37.     Console.Write(A[0, 0]);
38.     Console.SetCursorPosition(15, 10);
39.     Console.Write(A[0, 1]);
40.     Console.SetCursorPosition(10, 12);
41.     Console.Write(A[1, 0]);
42.     Console.SetCursorPosition(15, 12);
43.     Console.Write(A[1, 1]);
44.
45.     //Mantém a tela aberta esperando a digitação de
46.     uma tecla
47.     Console.ReadKey();
48. }
49. }
50. }
```

Variáveis indexadas Multidimensional – Exemplo

```
1. namespace Aula09_Ex04
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração de Variáveis
8.             int[,] A = new int[2, 2];
9.             byte c, l;
10.
11.             // Recebe os valores da matriz
12.             for (l = 0; l <= 1; l++)
13.             {
14.                 for (c = 0; c <= 1; c++)
15.                 {
16.                     Console.Write("Forneça o valor
17. A {0}{1} ", l+1, c+1);
18.                     A[l, c] =
19. int.Parse(Console.ReadLine());
20.                 }
21.             }
22.         }
23.     }
24. }
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Obj

```
Forneça o valor A 11 1
Forneça o valor A 12 2
Forneça o valor A 21 3
Forneça o valor A 22 4
```

```
| 1  2 |
| 3  4 |
```

```
20.
21. //Monta a matriz
22. // Barra esquerda
23. for (l = 9; l <= 13; l++)
24. {
25.     Console.SetCursorPosition(8, l);
26.     Console.WriteLine("|");
27. }
28. // Barra direita
29. for (l = 9; l <= 13; l++)
30. {
31.     Console.SetCursorPosition(18, l);
32.     Console.WriteLine("|");
33. }
34.
35. // Coloca os números na matriz
36. Console.SetCursorPosition(10, 10);
37. Console.Write(A[0, 0]);
38. Console.SetCursorPosition(15, 10);
39. Console.Write(A[0, 1]);
40. Console.SetCursorPosition(10, 12);
41. Console.Write(A[1, 0]);
42. Console.SetCursorPosition(15, 12);
43. Console.Write(A[1, 1]);
44.
45. //Mantém a tela aberta esperando a digitação de
46. Console.ReadKey();
47. }
```

