Lista de Exercicios

**Array**

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

**namespace** CursoCSharp.Exercicios

{

**public** **class** ArrayExercicio

    {

        /\* 1.   Criando e Inicializando um Array

            Crie um array de inteiros chamado numeros com capacidade para 5 elementos.

            Inicialize-o com os valores 2, 4, 6, 8, 10 e imprima todos os elementos. \*/

**public** **static** **void** Exercicio1()

            {

**int**[] numeros **=** **new** **int**[5] { 2, 4, 6, 8, 10};

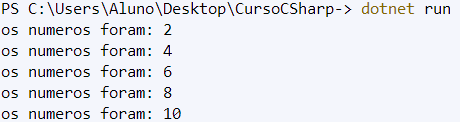
**for**(**int** i **=** 0; i < numeros.Length; i**++**)

                {

                    Console.WriteLine($"os numeros foram: {numeros[i]}");

                }

            }



        /\* 2.   Acessando Elementos de um Array

            Dado o array int[] valores = {1, 3, 5, 7, 9};,

            imprima o terceiro elemento do array. \*/

**public** **static** **void** Exercicio2()

            {

**int**[] valores **=** {1, 3, 5, 7, 9};

                Console.WriteLine($"o terceiro elemento do array valores foi: {valores[2]}");

            }



        /\* 3.    Alterando Valores em um Array

            Crie um array de strings chamado nomes com os valores "Ana", "Bruno", "Carla".

            Mude o valor de "Bruno" para "Beto" e depois imprima todos os elementos do array. \*/

            /\* public static void Exercicio3()

            {

                List<string> nomes = new List<string> { "Ana", "Bruno", "Carla" };

                foreach(var nome in nomes)

                {

                    Console.WriteLine($"O nome na posição {nomes.IndexOf(nome)} foi: {nome}");

                }

                int indiceBruno = nomes.IndexOf("Bruno");

                nomes.Insert(indiceBruno, "Beto");

                nomes.Remove("Bruno");

                foreach(var nome in nomes)

                {

                    Console.WriteLine($"O nome na posição {nomes.IndexOf(nome)} foi: {nome}");

                }

            } \*/

**public** **static** **void** Exercicio3()

            {

**string**[] nomes **=** { "Ana", "Bruno", "Carla" };

**foreach** (var nome **in** nomes)

                {

                    Console.WriteLine($"Nome: {nome}");

                }

                Console.WriteLine("\n");

**for** (**int** i **=** 0; i < nomes.Length; i**++**)

                {

**if** (nomes[i] **==** "Bruno")

                    {

                        nomes[i] **=** "Beto";

                    }

                }

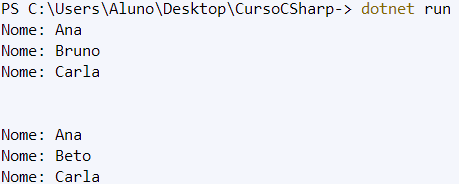
**foreach** (var nome **in** nomes)

                {

                    Console.WriteLine($"Nome: {nome}");

                }

            }



        /\* 4.   Calculando a Soma dos Elementos de um Array

            Crie um array de inteiros chamado notas com os valores 7, 8, 9, 10, 6.

            Calcule e imprima a soma de todos os elementos do array. \*/

**public** **static** **void** Exercicio4()

            {

**int**[] notas **=** { 7, 8, 9, 10, 6};

**int** soma **=** 0;

**for** (**int** i **=** 0; i < notas.Length; i**++**)

                {

                    soma **+=** notas[i];

                }

                Console.WriteLine($"A soma das notas foi: {soma}");

            }



        /\* 5.   Procurando um Elemento em um Array

            Dado o array int[] numeros = {5, 10, 15, 20, 25};,

            escreva um código que verifique se o número 15 está presente no array e

            imprima uma mensagem confirmando. \*/

**public** **static** **void** Exercicio5()

            {

**int**[] numeros **=** { 5, 10, 15, 20, 25};

**bool** contemQuinze **=** numeros.Contains(15);

                Console.WriteLine($"O array contém o número 15: {contemQuinze}");

            }



    }

}

**Lista**

**namespace** CursoCSharp.Exercicios

{

**public** **class** ListExercicio

    {

        /\* 1.   Criando e Adicionando Elementos em uma List

        Crie uma List<string> chamada frutas.

        Adicione as frutas "Maçã", "Banana", "Laranja" e

        imprima todos os elementos da lista.\*/

**public** **static** **void** Exercicio1()

            {

                List<**string**> frutas **=** **new** List<**string**>();

                frutas.Add("Maçã");

                frutas.Add("Banana");

                frutas.Add("Laranja");

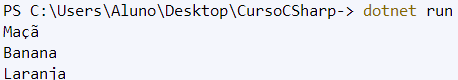
**foreach** (**string** fruta **in** frutas)

                {

                    Console.WriteLine(fruta);

                }

            }



        /\* 2.   Removendo um Elemento de uma List

            Crie uma List<int> chamada numeros e adicione os valores 1, 2, 3, 4, 5.

            Remova o número 3 da lista e depois imprima os elementos restantes. \*/

**public** **static** **void** Exercicio2()

        {

            List<**int**> numeros **=** **new** List<**int**> {1,2,3,4,5};

            numeros.Remove(3);

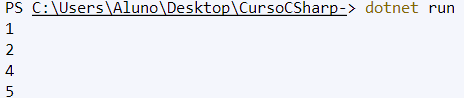
**foreach** (**int** numero **in** numeros)

            {

                Console.WriteLine(numero);

            }

        }



        /\* 3.   Acessando um Elemento Específico em uma List

            Dada a List<string> chamada cidades = new List<string> {"São Paulo", "Rio de Janeiro", "Belo Horizonte"},

            imprima a segunda cidade da lista. \*/

**public** **static** **void** Exercicio3()

        {

            List<**string**> cidades **=** **new** List<**string**> { "São Paulo", "Rio de Janeiro", "Belo Horizonte" };

            Console.WriteLine($"A segunda cidade é: {cidades[1]}");

        }



        /\* 4.   Contando Elementos em uma List

            Crie uma List<string> chamada animais e adicione os valores "Cachorro", "Gato", "Coelho".

            Escreva um código que imprima quantos elementos existem na lista. \*/

**public** **static** **void** Exercicio4()

        {

            List<**string**> animais **=** **new** List<**string**> { "Cachorro", "Gato", "Coelho"};

            Console.WriteLine($"Existem {animais.Count} animais na lista.");

        }



    }

}

Escopo de Variáveis

**namespace** CursoCSharp.Exercicios

{

**public** **class** EscopoVariaveisExercicio

    {

        /\* 1.   Variável Local em um Método

            Escreva um método chamado CalculaSoma que declare uma variável local int soma e

            calcule a soma de dois números passados como parâmetros. \*/

**public** **void** CalculaSoma(**int** num1, **int** num2)

            {

**int** soma **=** num1 + num2;

                Console.WriteLine($"O resultado da soma de {num1} + {num2} = {soma}");

            }



        /\* 2.   Variável Global em uma Classe

            Crie uma classe chamada Carro com uma variável global string marca.

            No método Main, instancie um objeto Carro, defina a marca do carro como "Toyota" e imprima a marca. \*/

**public** **class** Carro

            {

**public** **string**? marca;

**public** **void** Escrever()

                {

                    Console.WriteLine($"Marca: {marca}");

                }



                /\* 3.   Conflito de Escopo

                Escreva um método onde você declare uma variável local com o mesmo nome de uma variável global da classe.

                Mostre como acessar ambas as variáveis no mesmo método. \*/

**public** **void** ConflitoEscopo()

                {

**string**? marca;

                    marca **=** "Fiat";

                    Console.WriteLine($"Variável local: {marca}");

                }



            }

        /\* 4.   Variável Local dentro de um Bloco Condicional

            Escreva um código onde uma variável int x seja declarada dentro de um bloco if.

            Tente acessar essa variável fora do bloco if e observe o que acontece. \*/

**public** **void** XCondicional(**int** y)

            {

**if** (y > 3)

                {

**int** x **=** y + 5;

                    Console.WriteLine($"Valor de x: {x}");

                }

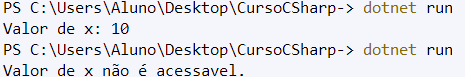
**else**

                {

                    Console.WriteLine("Valor de x não é acessavel.");

                }

            }



          /\* 5. Alterando o Valor de uma Variável Global

            Crie uma variável global int contador em uma classe.

            No método Main, altere o valor de contador dentro de um loop for e depois imprima o valor final de contador.\*/

**public** **class** Contador{

**public** **int** contador **=** 0;

**public** **void** IncrementarPor10(**int** i)

                {

                    Console.Write($"O valor incrementado {i} vezes por 10.");

**while** (i-1 >= 0)

                    {

                        contador **+=** 10;

                        i**--**;

                    }

                    Console.Write($" O total atual é {contador}");

                }

            }





        /\* 6.   Capturando uma Exceção de Conversão de Tipo

            Tente converter uma string não numérica para um inteiro usando int.Parse.

            Use try-catch para capturar a exceção e imprima uma mensagem apropriada. \*/

**public** **void** CapturarExcecao(**string** converterNum)

            {

**try**

                {

**int** num **=** **int**.Parse(converterNum);

                    Console.WriteLine($"Número convertido: {num}");

                }

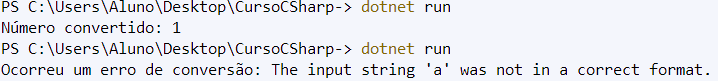
**catch** (FormatException ex)

                {

                    Console.WriteLine($"Ocorreu um erro de conversão: {ex.Message}");

                }

            }



        /\* 7.   Lançando Exceções

            Escreva um método que recebe um número inteiro.

            Se o número for negativo, lance uma ArgumentException.

            Capture a exceção no método Main e imprima uma mensagem apropriada. \*/

**public** **void** NumeroPositivo(**int** numero)

        {

**if** (numero < 0)

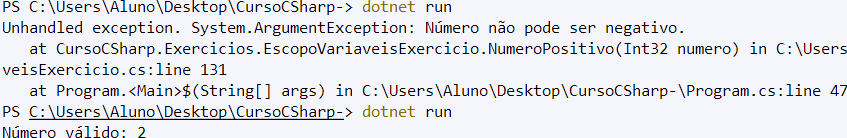
            {

**throw** **new** ArgumentException("Número não pode ser negativo.");

            }

            Console.WriteLine($"Número válido: {numero}");

        }



    }

}