Arrays

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 = { 2, 4, 6, 8, 10 };

            foreach (int numero in numeros)

            {

                Console.WriteLine($"Os numeros foram {numero}");

            }

        }

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 é: {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()

        {

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

            nomes[1] = "Beto";

            foreach (string 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;

            foreach (int nota in notas)

            {

                soma += nota;

                }

                Console.WriteLine($"A soma dos elementos do array é: {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 encontrado = false;

            foreach (int numero in numeros)

            {

                if (numero == 15)

                {

                    encontrado = true;

                    break;

                }

            }

            if (encontrado)

            {

                Console.WriteLine("O número 15 está presente no array.");

            }

            else

            {

                Console.WriteLine("O número 15 não está presente no array.");

            }

        }

List

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 Exercicio6()

        {

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

        frutas.Add(" Maçã ");

        frutas.Add(" Banana ");

        frutas.Add(" Laranja ");

        Console.Write("Lista de frutas:");

        foreach (string fruta in frutas)

        {

            Console.Write(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 Exercicio7()

{

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

    Console.WriteLine();

    Console.Write("Lista com todos os valores: ");

    foreach (int numero in numeros)

    {

        Console.Write(numero + " ");

    }

     Console.WriteLine();

    numeros.Remove(3);

    Console.WriteLine();

    Console.Write("Números sem o 3: ");

    foreach (int numero in numeros)

    {

        Console.Write(numero + " ");

    }

}

3. Acessando um Elemento Específico em uma List

Dada a List <string> chamada cidades = new List <string> {"São Paulo", "Brasilia", "Belo Horizonte"}, imprima a segunda cidade da lista.

 public static void Exercicio8()

 {

    List<string> cidades = new List<string> { "São Paulo", "Brasilia", "Belo Horizonte" };

    Console.WriteLine();

    Console.WriteLine(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 Exercicio9()

{

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

    Console.WriteLine();

    Console.WriteLine("Quantidade de animais na lista: " + animais.Count);

}

Escopo de Variáveis

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. A variável

soma deve estar acessível apenas dentro deste método.

public static void CalculaSoma()

{

    int numero1 = 5;

    int numero2 = 3;

    int soma = numero1 + numero2;

    Console.WriteLine($"A soma de {numero1} e {numero2} é: {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 static void VariavelLocalBlocoCondicional()

        {

        int x = 0;

        if (x > 0)

        {

            int y = x + 5;

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

        }

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

        }

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 static void ConversaoTipo()

        {

            string str = "10";

            try

            {

                int numero = int.Parse(str);

                Console.WriteLine($"O número convertido é: {numero}");

            }

            catch (FormatException)

            {

                Console.WriteLine("A string não é um número válido.");

            }

            catch (OverflowException)

            {

                Console.WriteLine("O número é muito grande para ser convertido.");

            }

        }

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 static void LancandoExcecao(int numero)

        {

            if (numero < 0)

            {

                throw new ArgumentException("O número não pode ser negativo.");

            }

            Console.WriteLine("Número válido.");

        }