**DESCRIÇÃO DETALHADA DOS EXERCÍCIOS**

**Descrição Detalhada do Exercício 7 – Soma de Números Pares de um Array**

Este exercício tem como objetivo desenvolver um programa em C# que solicita ao usuário a entrada de 10 números inteiros, armazena-os em um array e calcula a soma apenas dos números pares inseridos.​

**Objetivos do Exercício**

Entrada de Dados: Solicitar ao usuário que digite 10 números inteiros.

Armazenamento: Utilizar um array para armazenar os números fornecidos.

Processamento: Identificar os números pares e calcular sua soma.

Saída de Dados: Exibir a soma dos números pares no console.​

**Componentes do Código**

1. **Importação de Bibliotecas:**

*using System;*

*using System.Globalization;*

* System: Necessária para operações básicas, como entrada e saída de dados.
* System.Globalization: Importada, mas não utilizada neste código específico.​

1. **Classe Principal:**

*class Exerc7 {*

* Define a classe principal do programa.​

1. **Método Main:**

*static void Main(){*

* Ponto de entrada do programa.​

1. **Leitura da Entrada do Usuário:**

*Console.WriteLine("Digite um número: ");*

*int num = int.Parse(Console.ReadLine() ?? "0");*

* Solicita ao usuário que digite um número.
* Lê a entrada do usuário e a converte para um inteiro.
* O operador de coalescência nula (??) assegura que, se Console.ReadLine() retornar null, o valor padrão "0" será utilizado, evitando exceções.

1. **Cálculo e Exibição da Tabuada:**

*for (int i = 1; i < 11; i++) {*

*Console.WriteLine($"\n{num} x {i} = " + num\*i);*

*}*

* Utiliza um loop for que varia i de 1 a 10.
* Em cada iteração, calcula o produto de num por i e exibe o resultado.
* A interpolação de strings ($"\n{num} x {i} = ") é usada para formatar a saída de maneira legível.

**Conceitos de Programação Aplicados**

* **Entrada e Saída de Dados**: Uso de Console.WriteLine e Console.ReadLine para interagir com o usuário.
* **Conversão de Tipos**: int.Parse para converter a entrada de string para inteiro.
* **Estrutura de Repetição**: Uso do loop for para iterar de 1 a 10.
* **Interpolação de Strings**: Facilita a formatação da saída no console.​

**Descrição Detalhada do Exercício 8 – Cálculo do IMC em C#**

Este exercício tem como objetivo desenvolver um programa em C# que solicita ao usuário a entrada de sua altura e peso, calcula o Índice de Massa Corporal (IMC) e classifica o resultado conforme faixas predefinidas.​

**Objetivos do Exercício**

Entrada de Dados: Solicitar ao usuário que digite sua altura (em metros) e peso (em quilogramas).

Processamento: Calcular o IMC utilizando a fórmula:​ IMC = Peso / (altura \* altura)

Classificação: Determinar a categoria do IMC com base em faixas estabelecidas.

Saída de Dados: Exibir a classificação correspondente ao IMC calculado.​

**Componentes do Código**

1. **Importação de Bibliotecas:**

*using System;*

* System: Necessária para operações básicas, como entrada e saída de dados.

1. **Classe Principal**:

*static void Main() {*

* Ponto de entrada do programa.

1. **Leitura da Altura e Peso**:

*Console.WriteLine("Digite a sua altura: ");*

*float altura = float.Parse(Console.ReadLine() ?? "0");*

*Console.WriteLine("Digite o seu peso: ");*

*float peso = float.Parse(Console.ReadLine() ?? "0");*

* Solicita ao usuário que digite sua altura e peso.
* Lê as entradas do usuário e as converte para o tipo float.
* O operador de coalescência nula (??) assegura que, se Console.ReadLine() retornar null, o valor padrão "0" será utilizado, evitando exceções.

1. **Cálculo do IMC**:

*float imc = peso / (altura \* altura);*

* Aplica a fórmula do IMC utilizando os valores fornecidos pelo usuário.

1. **Classificação do IMC**:

*if(imc < 18.5){*

*Console.WriteLine("Abaixo do peso.");*

*} else if(imc >= 18.5 && imc < 25){*

*Console.WriteLine("Peso normal.");*

*} else if(imc > 25 && imc < 30){*

*Console.WriteLine("Sobrepeso.");*

*} else if(imc > 30){*

*Console.WriteLine("Obesidade.");*

*}*

* Utiliza estruturas condicionais if e else if para determinar a categoria do IMC com base em faixas estabelecidas.
* Exibe a classificação correspondente no console

**Conceitos de Programação Aplicados**

* **Entrada e Saída de Dados**: Uso de Console.WriteLine e Console.ReadLine para interagir com o usuário.
* **Conversão de Tipos**: float.Parse para converter a entrada de string para número de ponto flutuante.
* **Operadores Aritméticos**: Utilização da fórmula do IMC para cálculo.
* **Estruturas Condicionais**: Uso de if e else if para classificar o IMC.