

Exemplo de Classe a utilizando “get/set”, “Construtor” e “Método”:

```
namespace ExercicioCachorro
{
    public class Cachorro
    {
        public string Nome { get; set; }
        public string Raca { get; set; }

        public Cachorro(string nome, string raca)
        {
            Nome = nome;
            Raca = raca;
        }

        public void Latir()
        {
            System.Console.WriteLine("Au au!");
        }
    }
}
```

Exercícios de Classes (Nível Básico)

Instruções Gerais:

- Para cada exercício, crie uma nova classe (.cs).
 - No método `Main` de outra classe (ou na mesma, para simplicidade), crie instâncias dos objetos e teste os métodos.
 - Concentre-se em definir atributos (variáveis), construtores e métodos (funções).
-

Exercício 1: Classe `Pessoa`

Crie uma classe `Pessoa` com os seguintes atributos:

- `Nome` (string)
- `Idade` (int)
- `Altura` (double)

Crie um construtor que receba todos esses atributos.

Crie um método `Apresentar()` que exiba no console uma mensagem como: "Olá, meu nome é [Nome], tenho [Idade] anos e [Altura]m de altura."

Exercício 2: Classe `Produto`

Crie uma classe `Produto` com os seguintes atributos:

- `Nome` (string)
- `Preco` (double)
- `QuantidadeEmEstoque` (int)

Crie um construtor que receba `Nome` e `Preco`. A `QuantidadeEmEstoque` deve ser inicializada como 0.

Crie um método `AdicionarEstoque(int quantidade)` que aumente a `QuantidadeEmEstoque`.

Crie um método `RemoverEstoque(int quantidade)` que diminua a `QuantidadeEmEstoque`, garantindo que não fique negativo.

Crie um método `ExibirDetalhes()` que mostre todos os dados do produto.

Exercício 3: Classe `ContaBancaria`

Crie uma classe `ContaBancaria` com os atributos:

- `NumeroConta` (string)
- `Saldo` (double)

Crie um construtor que receba o `NumeroConta` e inicialize o `Saldo` como 0.

Crie um método `Depositar(double valor)` que adicione valor ao `Saldo`.

Crie um método `Sacar(double valor)` que retire valor do `Saldo`, mas apenas se houver saldo suficiente. Retorne `true` ou `false` para indicar sucesso.

Crie um método `ConsultarSaldo()` que retorne o `Saldo` atual.

Exercício 4: Classe `Retangulo`

Crie uma classe `Retangulo` com os atributos:

- `Largura` (double)
- `Altura` (double)

Crie um construtor que receba `Largura` e `Altura`.

Crie um método `CalcularArea()` que retorne a área do retângulo.

Crie um método `CalcularPerimetro()` que retorne o perímetro do retângulo.

Exercício 5: Classe `Carro`

Crie uma classe `Carro` com os atributos:

- `Marca` (string)
- `Modelo` (string)
- `Ano` (int)
- `VelocidadeAtual` (double) - inicialize com 0.

Crie um construtor que receba `Marca`, `Modelo` e `Ano`.

Crie um método `Acelerar(double incremento)` que aumente a `VelocidadeAtual`.

Crie um método `Frear(double decremento)` que diminua a `VelocidadeAtual`, garantindo que não fique negativa.

Crie um método `ExibirStatus()` que mostre a marca, modelo, ano e velocidade atual do carro.

Exercício 6: Classe `Aluno`

Crie uma classe `Aluno` com os atributos:

- `Nome` (string)
- `Matricula` (string)
- `Nota1`, `Nota2`, `Nota3` (double)

Crie um construtor que receba `Nome` e `Matricula`. As notas devem ser inicializadas com 0.

Crie um método `RegistrarNotas(double n1, double n2, double n3)` para atribuir as notas.

Crie um método `CalcularMedia()` que retorne a média das três notas.

Crie um método `VerificarAprovacao()` que retorne `true` se a média for maior ou igual a 7.0, e `false` caso contrário.

Exercício 7: Classe `Livro`

Crie uma classe `Livro` com os atributos:

- `Titulo` (string)
- `Autor` (string)
- `NumeroPaginas` (int)
- `Disponivel` (bool) - inicialize como `true`.

Crie um construtor que receba `Titulo`, `Autor` e `NumeroPaginas`.

Crie um método `Emprestar()` que mude `Disponivel` para `false` se o livro estiver disponível. Retorne `true` se o empréstimo foi realizado, `false` caso contrário.

Crie um método `Devolver()` que mude `Disponivel` para `true`.

Crie um método `ExibirInfo()` que mostre o título, autor, número de páginas e se está disponível.

Exercício 8: Classe `Circulo`

Crie uma classe `Circulo` com o atributo:

- `Raio` (double)

Crie um construtor que receba o `Raio`.

Crie um método `CalcularArea()` que retorne a área do círculo (use `Math.PI`).

Crie um método `CalcularCircunferencia()` que retorne a circunferência do círculo.

Exercício 9: Classe `Temperatura`

Crie uma classe `Temperatura` com o atributo:

- `ValorCelsius` (double)

Crie um construtor que receba o `ValorCelsius`.

Crie um método `ConverterParaFahrenheit()` que retorne a temperatura em Fahrenheit.

(Fórmula: $F = C * 1.8 + 32$)

Crie um método `ConverterParaKelvin()` que retorne a temperatura em Kelvin.

(Fórmula: $K = C + 273.15$)

Exercício 10: Classe `ItemDePedido`

Crie uma classe `ItemDePedido` com os atributos:

- `NomeProduto` (string)
- `PrecoUnitario` (double)
- `Quantidade` (int)

Crie um construtor que receba todos os atributos.

Crie um método `CalcularSubtotal()` que retorne o preço unitário multiplicado pela quantidade.

Crie um método `ExibirItem()` que mostre o nome do produto, quantidade, preço unitário e o subtotal.
