

PROVA PRÁTICA - ORIENTAÇÃO A OBJETOS EM C#

1) Sistema de Cadastro de Pessoas

Você foi contratado para desenvolver um pequeno módulo de cadastro.

Crie uma classe Pessoa contendo:

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

Regras:

- Idade não pode ser negativa.
- Crie um método Apresentar() que retorne: "Meu nome é X, tenho Y anos e Z metros."
- Instancie 3 pessoas no Main e imprima os dados.

Objetivo: entender classe, objeto, método e validação básica.

2) Controle de Produto com Desconto Percentual

Uma loja precisa calcular descontos automaticamente.

Crie uma classe Produto contendo:

- Nome
- Preco (double)
- Estoque (int)

Implemente:

- Método AplicarDesconto(double percentual)
- O percentual deve ser aplicado matematicamente ($\text{Preco} - (\text{Preco} * \text{percentual}/100)$)
- Não permitir percentual negativo.

Teste aplicando 15% e depois 5% novamente.

Objetivo: encapsulamento + matemática básica.

3) Sistema de Conta Bancária

Crie uma classe ContaBancaria:

- NumeroConta
- Titular
- Saldo (privado)

Métodos:

- Depositar(double valor)
- Sacar(double valor)
- Não permitir saque maior que o saldo
- Não permitir valores negativos

Extra:

- Criar método Transferir(ContaBancaria destino, double valor)

Objetivo: encapsulamento real + regras de negócio.

4) Sistema Escolar com Média Manual

Crie uma classe Aluno:

- Nome
- Lista de notas (List)

Métodos:

- AdicionarNota(double nota)
- CalcularMedia()

Regra:

- NÃO usar .Average()
- Calcular manualmente somando e dividindo pela quantidade

Extra:

- Se média $\geq 7 \rightarrow$ Aprovado
- Senão \rightarrow Reprovado

Objetivo: trabalhar listas e lógica matemática.

5) Herança - Sistema de Funcionários

Crie classe base Funcionario:

- Nome
- SalarioBase
- Método virtual CalcularSalario()

Crie:

- Desenvolvedor \rightarrow bônus de 20%
- Gerente \rightarrow bônus de 35%

Calcular:

$$\text{SalarioFinal} = \text{SalarioBase} + (\text{SalarioBase} * \text{percentual})$$

Objetivo: herança + sobreescrita.

6) Polimorfismo Real

Crie uma lista List contendo:

- 2 Desenvolvedores
- 1 Gerente

Percorra a lista chamando CalcularSalario().

Sem if ou switch.

Objetivo: entender polimorfismo de verdade.

7) Sistema de Formas Geométricas (Classe Abstrata)

Crie classe abstrata Forma:

- Método abstrato CalcularArea()

Crie:

- Círculo (Área = $\pi * r^2$)
- Retângulo (base * altura)

Instancie em uma lista de Forma e calcule todas as áreas.

Objetivo: abstração + matemática.

8) Sistema de Pagamentos (Interface)

Crie interface IPagamento:

- ProcessarPagamento(double valor)
- CalcularTaxa(double valor)

Implemente:

- Pix (taxa 1%)
- CartãoCredito (taxa 5%)
- Boleto (taxa fixa 2 reais)

Calcular valor final após taxa.

Objetivo: interface + polimorfismo.

9) Sistema de Notificações

Crie interface INotificação:

- Enviar(string mensagem)

Implemente:

- Email
- SMS
- WhatsApp

Simule envio usando Console.WriteLine().

Objetivo: desacoplamento.

10) Sistema de RPG (Simulação Real)

Crie classe base Personagem:

- Nome
- Vida
- AtaqueBase

Método virtual Atacar()

Crie:

- Guerreiro → dano = AtaqueBase * 1.5
- Mago → dano = AtaqueBase * 2, porém perde 5 de vida por ataque

Simule batalha entre dois personagens.

Objetivo: herança + regra matemática + estado.

11) Sistema de Veículos

Classe base Veiculo:

- VelocidadeAtual
- Método virtual Mover()

Carro → aumenta 20 km/h

Moto → aumenta 30 km/h

Bicicleta → aumenta 10 km/h

Simular 3 movimentos consecutivos.

Objetivo: estado interno + polimorfismo.

12) Sistema de Biblioteca Completo

Classe Livro:

- Titulo
- Autor
- Ano
- Emprestado (bool)

Classe Biblioteca:

- Lista de livros
- Método Emprestar(string titulo)
- Método Devolver(string titulo)

Não usar LINQ.

Objetivo: manipulação manual de lista.

13) Sistema de Pedido (E-commerce Parcial)

Classe Produto:

- Nome
- Preco

Classe Pedido:

- Lista de produtos

- Método AdicionarProduto
- Método CalcularTotal()

Extra:

- Se total > 500 → desconto 10%
- Se total > 1000 → desconto 15%

Objetivo: regras condicionais reais.

14) Sistema de Carrinho com Interface de Pagamento

Integre exercício 13 com IPagamento.

O pedido deve receber uma forma de pagamento e calcular:

$$\text{TotalFinal} = \text{Total} + \text{Taxa}$$

Objetivo: composição + polimorfismo.

15) Sistema de E-commerce COMPLETO (Desafio Final Grande)

Modele:

Classe Usuario:

- Nome
- Email
- Senha (privada)
- Método Autenticar(string senha)

Classe Produto:

- Nome
- Preco
- Estoque
- Método DiminuirEstoque(int quantidade)

Classe Carrinho:

- Lista de produtos
- Método AdicionarProduto
- Método RemoverProduto
- Método CalcularSubtotal()

Classe Pedido:

- Usuario
- Lista de produtos
- Forma de pagamento (IPagamento)
- Método FinalizarPedido()

Regras:

- Não permitir finalizar se estoque for insuficiente
- Aplicar taxa da forma de pagamento
- Aplicar desconto progressivo:
 - acima de 1000 → 5%
 - acima de 2000 → 10%

- Mostrar resumo final detalhado

Objetivo: aplicar TODOS os conceitos:

Encapsulamento

Herança

Interface

Polimorfismo

Validação

Regras de negócio reais

Matemática aplicada

Organização de código limpa.
