

Exercícios

1. Crie uma hierarquia de classes para representar diferentes tipos de funcionários. Criar instâncias de Funcionario e Gerente, sobrescrever métodos exibir. Crie os atributos que achar necessário.
2. Modele uma relação entre Aluno e Curso onde um aluno pode estar associado a um curso. Associar um aluno a um curso, sem que o curso precise conhecer os alunos. Crie uma instância do objeto aluno e mostre todos os dados de aluno e cursos associados. Crie os atributos que achar necessário.
3. Modele um Pedido que **depende** de uma lista de ItemPedido para existir. Mostrar que ItemPedido só existe dentro de um Pedido. Crie os atributos que achar necessário.
4. Crie um relacionamento muitos-para-muitos entre Aluno e Disciplina, usando uma classe associativa Matricula. Permitir registrar e consultar as disciplinas que um aluno cursa e vice-versa. Crie os atributos que achar necessário.
5. Implemente um pequeno sistema de biblioteca com as seguintes classes:
 - Pessoa (base)
 - Leitor (derivada de Pessoa)
 - Livro
 - Emprestimo (classe associativa entre Leitor e Livro)
6. Simula uma escola com professores e alunos, onde cada professor tem um **array fixo de 3 alunos**. Criar instâncias e preencher o array manualmente. Trabalhar com herança e array fixo. Crie os atributos que achar necessários.
7. Um sistema que calcula a média de um aluno com base em duas notas. Se a quantidade de provas for 0, trata a exceção.
8. Simula o cadastro de um produto com tratamento para entrada inválida e exceções genéricas. Lidar com entrada do usuário, múltiplos catch, e validação de regra de negócio. (FormatException e outros).
9. Conta bancária com saque, depósito e uma exceção personalizada para saldo insuficiente. (Exceção personalizada).
10. Simula uma aplicação que consulta usuários em um banco remoto usando async/await e Task.Delay() para simular tempo de rede. Crie uma classe Usuario contendo nome e id. Outra que simule dados do usuário (repositório).