

Curso:	Jovem Programador
Módulo:	Programador Web
Unidade curricular:	Desenvolver aplicações backend para web
Docente:	Ricardo Köhler

Exercícios de Fixação - Programação orientado à objetos

1. Crie uma classe Termômetro com o atributo temperatura. Adicione métodos para definir a temperatura, converter para Celsius, Fahrenheit e exibir a temperatura atual. Encapsule o atributo temperatura.
2. Crie uma classe Cofre com os atributos senha e valorGuardado. Adicione métodos para depositar e retirar valores, garantindo que a senha correta seja fornecida. Use o encapsulamento para proteger a senha e o valor.
3. Crie uma classe Elevador com os atributos andarAtual, totalAndares e capacidade. Adicione métodos para subir, descer e exibir o andar atual. Encapsule os atributos para evitar valores inválidos (andar fora dos limites ou capacidade negativa).
4. Crie uma classe Calculadora com métodos para realizar operações matemáticas básicas (soma, subtração, multiplicação e divisão). Implemente o encapsulamento para controlar o acesso a um histórico de operações realizadas.
5. Crie uma classe Relógio com os atributos horas, minutos e segundos. Adicione métodos para ajustar o horário e exibir o horário atual no formato HH:MM. Encapsule os atributos para garantir que os valores estejam dentro dos limites corretos.
6. Crie uma classe CarrinhoDeCompras que permita adicionar, remover e listar produtos. Cada produto deve ter um nome e um preço. Adicione um método para calcular o valor total do carrinho. Utilize encapsulamento para proteger a lista de produtos.
7. Crie uma classe SensorDeUmidade com os atributos umidadeAtual e umidadeMaxima. Adicione métodos para atualizar e exibir a umidade. Implemente um método que avise quando a umidade máxima for atingida. Encapsule os atributos para evitar modificações diretas.

8. Crie uma classe Agenda com um atributo que armazene uma lista de contatos. Cada contato deve ter nome e telefone. Adicione métodos para adicionar, remover, buscar e exibir contatos. Encapsule a lista de contatos para que só possa ser manipulada pelos métodos da classe.
9. Crie uma classe CaixaRegistradora com os atributos totalVendas e quantidadeItens. Adicione métodos para registrar uma venda, calcular o troco e exibir o total de vendas e itens registrados. Encapsule os atributos para garantir que o total de vendas seja sempre positivo.
10. Crie uma classe Lampada com os atributos estado (ligada/desligada) e potencia. Adicione métodos para ligar, desligar e verificar o estado atual da lâmpada. Encapsule os atributos para evitar manipulações diretas.
11. Crie uma classe Banco com os atributos nome e uma lista de contas. Cada conta tem número, titular e saldo. Adicione métodos para criar conta, fechar conta, depositar e sacar dinheiro. Use o encapsulamento para proteger a lista de contas.
12. Crie uma classe Inventario que gerencie uma lista de itens. Cada item deve ter nome, quantidade e preço. Adicione métodos para adicionar, remover, buscar e listar itens, além de calcular o valor total do inventário. Encapsule a lista de itens.
13. Crie uma classe QuartoHotel com atributos número, capacidade e ocupacaoAtual. Adicione métodos para reservar, liberar e verificar a disponibilidade do quarto. Use o encapsulamento para proteger os atributos, garantindo que a ocupação não ultrapasse a capacidade.
14. Crie uma classe Biblioteca que gerencie uma coleção de livros. Cada livro deve ter título, autor e status (disponível/empréstimo). Adicione métodos para adicionar, emprestar, devolver e listar livros. Encapsule a lista de livros para proteger a integridade dos dados.