

Lista 2 - Exercícios OO – Herança e Polimorfismo

Resolver os exercícios abaixo, criando casos de teste para todas as situações.

1) Formas geométricas

Considere uma hierarquia de formas geométricas, incluindo círculos e retângulos. Cada forma deve ter um método `calcularArea()` que calcula a área da forma e um método `calcularPerimetro()` que calcula o perímetro da forma. Você deve usar polimorfismo para implementar esses métodos nas classes específicas de círculo e retângulo.

Crie uma classe abstrata chamada `FormaGeometrica` com os métodos abstratos `calcularArea()` e `calcularPerimetro()`.

Crie uma classe `Circulo` que herda de `FormaGeometrica` e inclua campos para o raio.

Implemente os métodos `calcularArea()` e `calcularPerimetro()` para o círculo.

Crie uma classe `Retangulo` que herda de `FormaGeometrica` e inclua campos para a largura e altura.

Implemente os métodos `calcularArea()` e `calcularPerimetro()` para o retângulo.

2) Sistema de Pagamento

Crie um sistema de pagamento para uma empresa que possui três tipos de funcionários: funcionários comuns, gerentes e diretores. Cada funcionário recebe um salário mensal fixo, mas os gerentes recebem um bônus adicional e os diretores têm uma participação nos lucros. Crie um programa em Java que calcule e exiba o pagamento mensal para cada funcionário.

Crie uma classe abstrata chamada `Funcionario` com um campo `salarioMensal` e um método abstrato `calcularPagamento()`.

Crie três classes que herdam de `Funcionario`: `FuncionarioComum`, `Gerente` e `Diretor`.

Na classe `FuncionarioComum`, o pagamento mensal é o salário mensal fixo.

Na classe `Gerente`, o pagamento mensal é o salário mensal fixo mais um bônus fixo.

Na classe `Diretor`, o pagamento mensal é o salário mensal fixo mais uma participação nos lucros que é calculada com base nos lucros da empresa.

3) Sistema de Pedidos

Crie um programa Java para um sistema de pedidos em uma loja online. O sistema deve lidar com diferentes tipos de produtos, como eletrônicos, roupas e livros. Cada tipo de produto tem um preço base e pode ter descontos específicos. Crie classes para representar os diferentes tipos de produtos e implemente um método de cálculo de preço com desconto.

Crie uma classe abstrata chamada `Produto` com campos para nome e preço base. Inclua um método abstrato `calcularPreco()`.

Crie três classes que herdam de `Produto`: `ProdutoEletronico`, `ProdutoRoupa` e `ProdutoLivro`.

Na classe ProdutoEletronico, o preço base é o preço de varejo e tem um desconto de 10%.

Na classe ProdutoRoupa, o preço base é o preço de etiqueta e tem um desconto de 20%.

Na classe ProdutoLivro, o preço base é o preço de capa e tem um desconto de 5%.

4) Sistema de passagens

Crie um programa Java para um sistema de reservas de passagens em uma companhia aérea. O sistema deve lidar com diferentes tipos de voos, como voos domésticos e voos internacionais, cada um com suas próprias regras de precificação. Crie classes para representar os diferentes tipos de voos e implemente um método de cálculo de preço.

Crie uma classe abstrata chamada Voo com campos para origem, distancia, destino e data do voo. Inclua um método abstrato calcularPreco().

Crie duas classes que herdam de Voo: VooDomestico e VooInternacional.

Na classe VooDomestico, o preço base é calculado com base na distância entre a origem e o destino multiplicada por um fator de preço específico para voos domésticos.

Na classe VooInternacional, o preço base é calculado com base na distância entre a origem e o destino multiplicada por um fator de preço específico para voos internacionais, além de uma taxa de conversão de moeda.

5) Sistema de Gerenciamento de Produtos

Crie um programa Java para um sistema de gerenciamento de produtos em uma loja. O sistema deve lidar com diferentes tipos de produtos, como eletrônicos, roupas e alimentos, cada um com suas próprias características e métodos de cálculo de preço. Crie classes para representar os diferentes tipos de produtos e implemente métodos para calcular o preço total com base nas quantidades compradas.

Crie uma classe abstrata chamada Produto com campos para nome, preço unitário e quantidade em estoque. Inclua um método abstrato calcularPreco().

Crie três classes que herdam de Produto: ProdutoEletronico, ProdutoRoupa e ProdutoAlimento.

Na classe ProdutoEletronico, o cálculo do preço total leva em consideração o preço unitário e a quantidade comprada.

Na classe ProdutoRoupa, o preço unitário é o preço de etiqueta e o cálculo do preço total leva em consideração a quantidade e um possível desconto.

Na classe ProdutoAlimento, o preço unitário é o preço por quilo e o cálculo do preço total leva em consideração a quantidade em quilos.

Não deve haver venda caso não exista estoque suficiente. Deve baixar o estoque do produto após a venda.

6) Sistema de Contas Bancárias

Crie um programa Java para um sistema de contas bancárias que gerencia diferentes tipos de contas, como contas correntes e contas de poupança. Cada tipo de conta tem

suas próprias regras para depósitos, saques e cálculos de juros. Crie classes para representar os diferentes tipos de contas e implemente métodos para realizar operações bancárias.

Crie uma classe abstrata chamada `ContaBancaria` com campos para número da conta, saldo e titular. Inclua métodos abstratos para `depositar()`, `sacar()`, e `calcularJuros()`.

Crie duas classes que herdam de `ContaBancaria`: `ContaCorrente` e `ContaPoupanca`.

Na classe `ContaCorrente`, o método `depositar()` permite depósitos ilimitados, mas cobra uma taxa de manutenção por depósito. O método `sacar()` permite saques, mas pode verificar se há fundos suficientes. A taxa de manutenção mensal é descontada automaticamente do saldo.

Na classe `ContaPoupanca`, o método `depositar()` permite depósitos ilimitados e o método `sacar()` permite saques, desde que haja saldo suficiente. O método `calcularJuros()` calcula e adiciona juros mensais à conta com base em uma taxa de juros fixa.

7) Sistema de Biblioteca

Crie um programa Java para um sistema de biblioteca que gerencia diferentes tipos de materiais, como livros, revistas e filmes. Cada tipo de material tem seu próprio método de cálculo de empréstimo e datas de devolução. Crie classes para representar os diferentes tipos de materiais e implemente esses métodos.

Crie uma classe abstrata chamada `MaterialBiblioteca` com campos para título e data de publicação. Inclua um método abstrato `calcularDataDevolucao()`.

Crie três classes que herdam de `MaterialBiblioteca`: `Livro`, `Revista` e `Filme`.

Na classe `Livro`, o método `calcularDataDevolucao()` calcula a data de devolução baseada em um prazo fixo, por exemplo, 15 dias após o empréstimo.

Na classe `Revista`, o método `calcularDataDevolucao()` calcula a data de devolução baseada em um prazo fixo diferente, por exemplo, 7 dias após o empréstimo.

Na classe `Filme`, o método `calcularDataDevolucao()` calcula a data de devolução baseada em um prazo fixo, por exemplo, 5 dias após o empréstimo.

8) Sistema de Veículos

Crie um programa Java para um sistema de gerenciamento de veículos que possui diferentes tipos de veículos, como carros, motos e bicicletas, cada um com suas próprias características e métodos de cálculo de custo. Crie classes para representar os diferentes tipos de veículos e implemente esses métodos.

Crie uma classe abstrata chamada `Veiculo` com campos para marca, modelo e preço base. Inclua um método abstrato `calcularCusto()`.

Crie três classes que herdam de `Veiculo`: `Carro`, `Moto` e `Bicicleta`.

Na classe `Carro`, o método `calcularCusto()` calcula o custo com base no preço base multiplicado por um fator fixo para carros.

Na classe `Moto`, o método `calcularCusto()` calcula o custo com base no preço base multiplicado por um fator fixo para motos.

Na classe Bicicleta, o método calcularCusto() calcula o custo com base no preço base multiplicado por um fator fixo para bicicletas.