

Lista de Exercícios 12

Classe Abstrata

Cada aluno deverá, individualmente, resolver os exercícios abaixo:

Orientações:

- Para cada exercício faça:
 - a) Elabore o diagrama de classes utilizando UML quando o diagrama não for dado;
 - b) Implemente ou complemente o código em Java quando o código não for dado;
 - c) Implemente um aplicativo específico para demonstração dos códigos.

1. Sistema de Gestão de Funcionários

Descrição

Você deve criar um sistema para gerenciar diferentes tipos de funcionários de uma empresa. A classe abstrata **Funcionario** servirá como base, definindo atributos e métodos comuns. As subclasses concretas (por exemplo, **FuncionarioHorista**, **FuncionarioMensalista**, **Estagiario**) devem sobrescrever métodos para calcular o salário de forma específica.

Estrutura sugerida

A. **Funcionario** (classe abstrata)

- a. Atributos: **nome**, **cpf**, **salarioBase**
- b. Métodos abstratos: **calcularSalario()**
- c. Métodos concretos: **imprimirDados()**

B. **FuncionarioHorista**

- a. Atributos específicos: **horasTrabalhadas**, **valorHora**
- b. Implementação de **calcularSalario()**
 - i. $\text{horasTrabalhadas} * \text{valorHora}$

C. FuncionarioMensalista

- a. Atributo específico: **beneficios**
- b. Implementação de **calcularSalario()**
 - i. $\text{salarioBase} + \text{beneficios}$

D. Estagiario

- a. Atributos específicos: **cargaHoraria**, **valorBolsa**
- b. Implementação de **calcularSalario()**
 - i. $\text{cargaHoraria} * \text{valorBolsa}$

E. FolhaPagamento

- a. Método **adicionarFuncionario(Funcionario f)**
- b. Método **processarFolha()** que itera pela lista de funcionários e soma os salários.

F. Aplicativo

- a. Criar objetos **FuncionarioHorista**, **FuncionarioMensalista**, **Estagiario**
- b. Adicionar à Folha **FolhaPagamento**
- c. Imprimir resultados.

2. Sistema de Loja de Roupas

Descrição

Implemente um sistema de uma loja de roupas que tenha diferentes categorias de produtos. Crie uma classe abstrata **Produto** com atributos e métodos comuns. Cada tipo de roupa (Camiseta, Calça, etc.) deve ter sua forma de calcular o preço final (podendo incluir descontos, impostos específicos).

Estrutura sugerida

A. Produto

- a. Atributos: **descricao**, **precoCusto**
- b. Métodos abstratos: **calcularPrecoVenda()**

B. Camiseta

- a. Atributos adicionais: **tamanho**, **estampa**
- b. Implementação de **calcularPrecoVenda()**
 - i. incluir margem de lucro e possível desconto

C. Calca

- a. Atributos adicionais: **tamanho, material**
- b. Implementação de **calcularPrecoVenda()**
 - i. incluir margem de lucro distinta

D. Sapato

- a. Atributos adicionais: **tipo, numero**
- b. Implementação de **calcularPrecoVenda()**
 - i. defina outro critério de cálculo

E. CarrinhoDeCompras

- a. Coleção de **Produto**
- b. Método **adicionarProduto(Produto p)**
- c. Método **calcularTotal()**

F. Aplicativo

- a. Instanciar produtos
- b. Adicionar ao carrinho
- c. Exibe valor total.

3. Sistema de Reservas de Quarto de Hotel

Descrição

Implemente um sistema que gerencie reservas de quarto em um hotel. A classe abstrata **Quarto** deve definir atributos e um método abstrato para calcular o valor da diária. As subclasses (**QuartoSimples**, **QuartoLuxo**, etc.) definem cálculos diferentes.

Estrutura sugerida

A. Quarto

- a. Atributos: **numero, capacidade, valorBase**
- b. Método abstrato: **calcularDiaria()**

B. QuartoSimples

- a. Implementação de **calcularDiaria()**
 - i. valorBase para 2 pessoas

C. QuartoLuxo

- a. Implementação de **calcularDiaria()**
 - i. valorBase + taxas extras

D. Reserva

- a. Atributos: **Quarto** **quartoReservado**,
dataEntrada, **dataSaida**
- b. Métodos: **calcularValorTotal()**
 - i. diferença de dias * **calcularDiaria()**

E. GerenciadorReservas

- a. Lista de **Reserva**
- b. Métodos para criar, cancelar, listar reservas

F. Aplicativo

- a. Criar quartos de diferentes tipos e reservas
- b. Imprimir valores de diárias e resumo das reservas.

4. Biblioteca de Mídias (Livros e Filmes)

Descrição

Crie uma biblioteca que armazena diferentes tipos de mídia: livros e filmes, por exemplo. A classe abstrata **Midia** definirá atributos e métodos básicos. As subclasses **Livro** e **Filme** terão comportamentos ou atributos específicos conforme descrição a seguir.

Estrutura sugerida

A. Midia

- a. Atributos: **titulo**, **anoLancamento**
- b. Métodos abstratos: **exibirDetalhes()**

B. Livro

- a. Atributos: **autor**, **editora**
- b. Implementação de **exibirDetalhes()**
 - i. incluir autor e editora

C. Filme

- a. Atributos: **diretor**, **duracao**
- b. Implementação de **exibirDetalhes()**
 - i. inclui diretor e duração

D. ItemBiblioteca

- a. Associa-se a **Midia** (composição ou agregação)
- b. Atributos específicos de controle

- i. código de localização, status de empréstimo (emprestado, devolvido, atradado)

E. **Biblioteca**

- a. Coleção de **Midia** (ou **ItemBiblioteca**)
- b. Métodos para adicionar, remover, buscar mídias

F. **Main**

- a. Criar livros e filmes,
- b. Adicionar à biblioteca
- c. Imprimir detalhes de todos os itens.

Bom trabalho.