

[illegible]

# Sistema de Pedidos em Restaurante

Desenvolva um sistema para gerenciar pedidos em um restaurante. Crie uma classe Pedido com os seguintes atributos:

- numero (int): número do pedido.
- itens (ArrayList<ItemPedido>): lista de itens no pedido, onde:
  - ItemPedido possui os atributos nomeDoPrato (String),
  - quantidade (int) e
  - precoUnitario (double).

**Adicione métodos para aplicar regras de negócios, como:**

- `calcularTotalPedido()`: método que calcula o valor total do pedido somando os preços de todos os itens e a taxa de entrega.
- `adicionarItem(ItemPedido item)`: método que adiciona um item ao pedido.
- `removerItem(ItemPedido item)`: método que remove um item do pedido.
- `reservarMesa(int numeroMesa)`: método que reserva uma mesa específica para o pedido.

**Crie uma classe Restaurante com os seguintes atributos e métodos:**

- pedidos (ArrayList<Pedido>): lista de pedidos realizados.
- Métodos para adicionar, remover e buscar pedidos na lista.
- Método para exibir todos os pedidos realizados.

**Ao final, crie objetos de cada classe, faça teste com 2 métodos na Classe Main.**

			<b>Notas</b>	
Campus	FATEC PRAIA GRANDE	Período	VESPERTINO	Prova
Disciplina	TÉCNICA DE PROGRAMAÇÃO I	Semestre	2º	Trabalho
Professor	ALESSANDRO FERREIRA PAZ LIMA	Sala		Total
Prova	P2-B	Data	13/06/2024	Revisada
Duração	<b>120min</b>	Conteúdo	2º BIMESTRE	Visto
Permanência	15 minutos no mínimo	Valor	0 a 10	
RA		Turma		

**LETRA  
LEGÍVEL**

Nome

# Sistema de Reservas de Passagens Aéreas

Imagine que você está desenvolvendo um sistema para reservas de passagens aéreas. Crie uma classe Voo com os seguintes atributos:

- numeroVoo (String): número do voo.
- origem (String): cidade de origem.
- destino (String): cidade de destino.
- assentosDisponiveis (int): número de assentos disponíveis.

**Adicione métodos para aplicar regras de negócios, como:**

- realizarReserva(int quantidadeAssentos): método que realiza a reserva de uma determinada quantidade de assentos.
- verificarDisponibilidade(int quantidadeAssentos): método que verifica se há assentos disponíveis.
- realizarPagamento(String tipoViagem, boolean pontosTuristicos): método que realiza o pagamento da passagem, considerando se é somente ida, ida e volta, e se inclui pontos turísticos (taxa adicional).
- imprimirPassagem(): método que imprime os detalhes da passagem.

**Crie uma classe Aeroporto com os seguintes atributos e métodos:**

- voos (ArrayList<Voo>): lista de voos disponíveis.
- Métodos para adicionar, remover e buscar voos na lista.
- Método para exibir todos os voos disponíveis.

**Ao final, crie objetos de cada classe, faça teste com 2 métodos na Classe Main.**

[illegible]

# Sistema de Treinamento em Tecnologia

Imagine que você está desenvolvendo um sistema para gerenciar o treinamento de programadores em uma empresa. Crie uma classe `Treinamento` com os seguintes atributos:

- id (int): identificador único do treinamento.
- nomeInstrutor (String): nome do instrutor do treinamento.
- linguagemEnsina (String): linguagem de programação ensinada no treinamento.
- alunos (ArrayList<Aluno>): lista de alunos inscritos no treinamento, onde Aluno possui os atributos nome (String) e notaFinal (double).

**Adicione métodos para aplicar regras de negócios, como:**

- `verificarDisponibilidade()`: método que verifica a disponibilidade do instrutor para o treinamento.
- `definirCargaHoraria(int horas)`: método que define a carga horária do treinamento.
- `verificarUltimoTreinamento(Aluno aluno)`: método que verifica se um programador pode fazer o treinamento, considerando o limite de 80 horas a cada 2 meses.
- `calcularMediaAlunos()`: método que calcula a média das notas finais dos alunos após o treinamento.

Utilize o conceito de polimorfismo para criar subclasses específicas para treinamentos presenciais e online. Cada uma dessas subclasses deve ter métodos específicos para o tipo de treinamento.

- Crie a classe `TreinamentoPresencial` com um atributo adicional local (String) e a classe `TreinamentoOnline` com um atributo adicional `linkAcesso` (String).

**Ao final, crie objetos de cada classe, faça teste com 2 métodos na Classe Main.**