### **Questão 1 – Gerenciamento de Funcionários**

A empresa “TechSolutions” precisa de um sistema para controlar seus funcionários. A classe Funcionario contém nome, cpf, salario e cargo.

**Crie métodos na classe Empresa para:**

* Adicionar um novo funcionário (armazenar em um ArrayList<Funcionario>).
* Remover um funcionário pelo CPF.
* Aumentar o salário de um funcionário específico.
* Verificar se um determinado cargo existe na empresa (pelo menos um funcionário com esse cargo).

### **Questão 2– Loja de Eletrodomésticos**

A classe Produto representa um item à venda, com codigo, nome, preco e quantidadeEstoque. A classe Loja possui um ArrayList<Produto> com todos os produtos disponíveis.

**Implemente os métodos na classe Loja:**

* Buscar um produto pelo código.
* Verificar se há um produto em estoque (quantidade maior que 0).
* Atualizar o estoque ao realizar uma venda.
* Exibir todos os produtos com estoque abaixo de 5 unidades.

### **Questão 3 – Sistema de Transporte Escolar**

A classe Aluno possui nome, matricula e endereco. A classe OnibusEscolar armazena os alunos em um ArrayList<Aluno>.

**Implemente métodos para:**

* Verificar se um aluno está cadastrado para o transporte.
* Adicionar um aluno.
* Remover um aluno pela matrícula.
* Listar todos os alunos de um determinado endereço.

### **Questão 4 – Clínica Veterinária**

A classe Animal possui nome, tipo (cachorro, gato, etc.), idade e nomeDono. A classe Clinica mantém uma lista de Animal.

**Crie métodos para:**

* Verificar se já existe um animal com determinado nome e nome do dono.
* Adicionar um novo animal.
* Remover um animal pelo nome e dono.
* Contar quantos animais de um certo tipo há cadastrados (por exemplo: quantos gatos?).

### **Questão 5 – Cadastro de Clientes por CPF**

Você está desenvolvendo um sistema de cadastro de clientes para uma empresa. Use um HashMap<String, Cliente>, onde a **chave é o CPF** e o valor é o objeto Cliente.

**Crie os métodos:**

* Adicionar um novo cliente.
* Verificar se um cliente está cadastrado (pelo CPF).
* Remover um cliente pelo CPF.
* Recuperar os dados de um cliente específico.

### **Questão 6 – Estoque de Produtos por Código**

Uma loja utiliza um HashMap<String, Produto> para gerenciar o estoque, onde a chave é o **código do produto**.

**Implemente os métodos:**

* Verificar se um produto está cadastrado no estoque.
* Atualizar a quantidade de um produto específico.
* Listar todos os produtos com estoque zerado.
* Remover um produto do sistema.

### **Questão 7 – Sistema de Notas de Alunos**

Você está criando um sistema de lançamento de notas. Use um HashMap<String, Double> para armazenar a **nota final de cada aluno**, onde a chave é o nome do aluno.

**Crie métodos para:**

* Adicionar ou atualizar a nota de um aluno.
* Verificar se um aluno está cadastrado.
* Calcular a média geral da turma.
* Listar todos os alunos com nota acima de 7.0.

### **Questão 8 – Agenda de Contatos**

Implemente uma agenda de contatos usando HashMap<String, String>, onde a **chave é o nome da pessoa** e o valor é o **número de telefone**.

**Crie métodos para:**

* Adicionar um contato.
* Atualizar o número de telefone de um contato.
* Remover um contato.
* Verificar se um nome está na agenda.

### **Questão 9 – Sistema de Pagamento com Herança**

A empresa “PayMaster” realiza pagamentos para diferentes tipos de funcionários. Existem dois tipos principais:

* **FuncionarioHorista**, que recebe por hora trabalhada.
* **FuncionarioMensalista**, que recebe um salário fixo por mês.

Crie uma superclasse Funcionario com os atributos: nome e cpf.  
 As subclasses devem conter os atributos específicos:

* FuncionarioHorista: valorHora e horasTrabalhadas
* FuncionarioMensalista: salarioMensal

Implemente na superclasse um método abstrato calcularPagamento() que será sobrescrito nas subclasses.

**Na classe principal (ou classe Empresa):**

* Cadastre diferentes tipos de funcionários em um ArrayList<Funcionario>.
* Implemente um método para exibir o pagamento de todos os funcionários.

### **Questão 10 – Cadastro de Veículos com Herança**

Você está desenvolvendo um sistema de cadastro para uma empresa de transporte. Todos os veículos possuem placa, marca e ano.

Crie uma superclasse Veiculo com esses atributos e subclasses para:

* Carro (adicional: número de portas)
* Moto (adicional: cilindradas)

Implemente:

* Um método exibirDetalhes() sobrescrito em cada subclasse para mostrar as informações específicas do veículo.
* Armazene os veículos em um ArrayList<Veiculo> e implemente um método para exibir os detalhes de todos.

### **Questão 11 – Sistema Escolar com Pessoas**

Crie uma superclasse Pessoa com os atributos nome, cpf e endereco.  
 Crie as subclasses:

* Aluno (atributos: matricula, curso)
* Professor (atributos: siape, disciplina)

Implemente os métodos:

* exibirDados() na superclasse e sobrescreva nas subclasses para mostrar informações completas.
* Armazene todas as pessoas em uma lista (ArrayList<Pessoa>) e implemente um método para listar somente os alunos ou somente os professores.