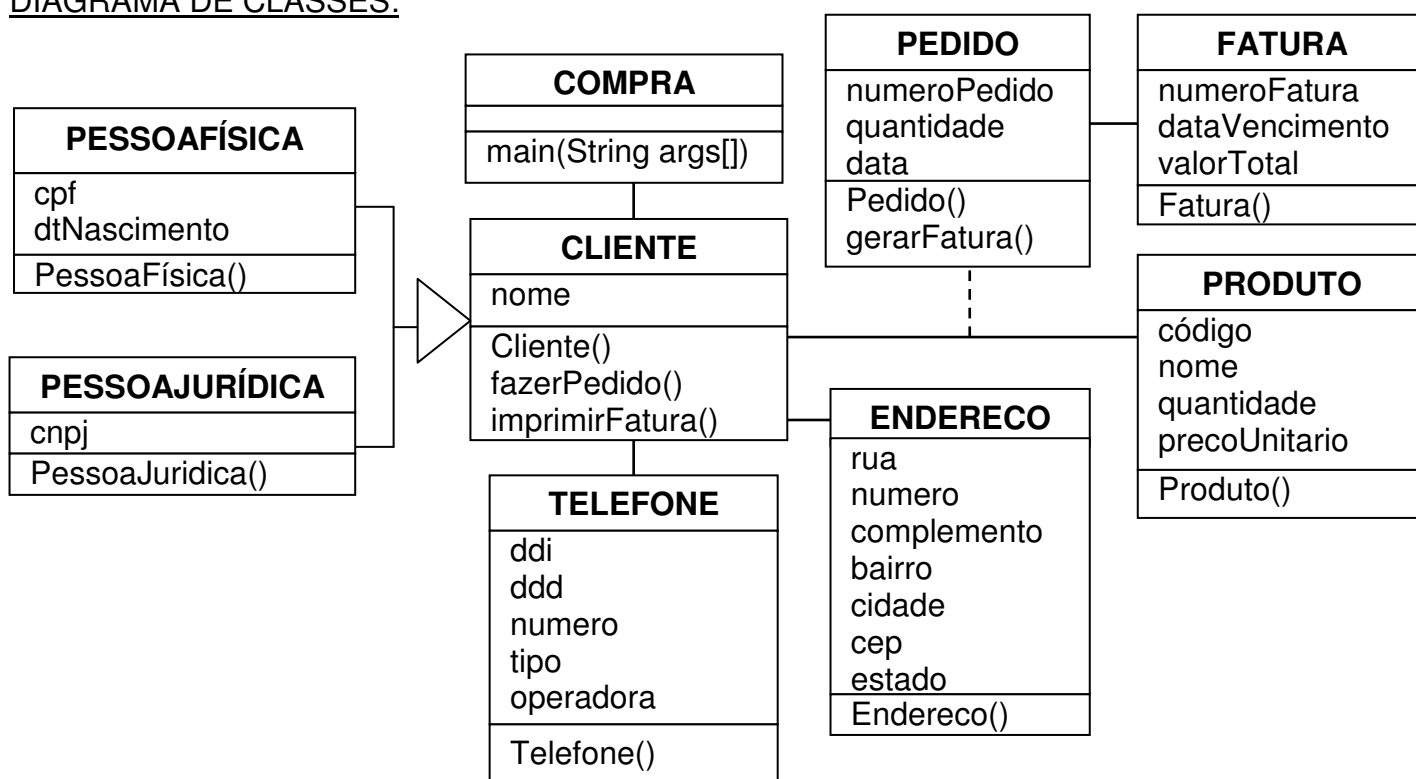


CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO – 3º PERÍODO DISCIPLINA INTRODUÇÃO À PROGRAMAÇÃO ORIENTADA A OBJETOS

LISTA DE EXERCÍCIOS 06 – CONCEITOS AVANÇADOS DA ORIENTAÇÃO A OBJETOS (ESTUDO DE CASO)

O diagrama de classes abaixo representa, de forma simplificada, uma loja que trabalha com venda de produtos. Cada produto solicitado pelo cliente (pessoa física ou pessoa jurídica) em uma determinada quantidade gera um pedido. Cada pedido gera uma fatura que contém o valor total a ser pago. Codifique em Java a aplicação que representa este cenário de acordo com as recomendações.

DIAGRAMA DE CLASSES:



ATENÇÃO: é obrigatória a implementação dos métodos de acesso para os atributos das classes, bem como a definição da visibilidade de todos os membros das classes. Os valores gravados nos atributos devem ser informados pelo usuário através de *JOptionPane*. DICA: inicie a codificação pelas classes mais simples: **ENDERECO**, **TELEFONE**, **PRODUTO** e **FATURA**.

CLASSE CLIENTE

A superclasse **CLIENTE** deve ser abstrata e por relacionar-se com as classes **ENDERECO**, **TELEFONE**, **PEDIDO** e **PRODUTO** deve conter um atributo de referência para cada delas. É preciso implementar os métodos de acesso para os atributos de referência. Considere que cada cliente (jurídica ou física) tem um único telefone e um único endereço e só pode fazer um pedido de uma determinada quantidade de um único produto.

O método construtor deve solicitar ao usuário o valor do atributo *nome* através do componente gráfico *JOptionPane*. Além disso, dentro do construtor deve-se criar uma instância da classe

CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO – 3º PERÍODO **DISCIPLINA INTRODUÇÃO À PROGRAMAÇÃO ORIENTADA A OBJETOS**

LISTA DE EXERCÍCIOS 06 – CONCEITOS AVANÇADOS DA ORIENTAÇÃO A OBJETOS **(ESTUDO DE CASO)**

ENDereco é uma instância da classe **TELEFONE**, relacionando-as aos respectivos atributos de referência.

Os métodos *fazerPedido()* e *imprimirFatura()* são concretos e devem ser implementados na própria superclasse.

Um **CLIENTE** faz um **PEDIDO** de um **PRODUTO**. Logo, o método *fazerPedido()* deve conter uma instância da classe **PEDIDO** e outra da classe **PRODUTO**, relacionando-as aos respectivos atributos de referência. Também é preciso relacionar a instância do **PRODUTO** com a instância do **PEDIDO**, vinculando o **PRODUTO** escolhido pelo **CLIENTE** ao **PEDIDO**. Por último, deve ser chamado o método *gerarFatura()* da classe **PEDIDO**.

O método *imprimirFatura()* deve mostrar na tela todos os atributos de **CLIENTE** (comuns e de referência), de **PEDIDO**, do **PRODUTO** e da **FATURA**.

CLASSE PESSOAFISICA e CLASSE PESSOA JURIDICA

Os métodos construtores destas subclasses devem chamar o método construtor da superclasse e depois solicitar ao usuário os valores dos respectivos atributos específicos de cada uma delas através do componente gráfico *JOptionPane*. Por último, ainda dentro dos métodos construtores, deve-se chamar os métodos *fazerPedido()* e *imprimirFatura()*.

CLASSE PEDIDO

A classe associativa **PEDIDO** é oriunda do relacionamento entre as classes **CLIENTE** e **PRODUTO**. Além disso, ela também relaciona-se com a classe **FATURA**. Sendo assim, a classe **PEDIDO** deve ter um atributo de referência para cada uma das classes com as quais se relaciona: **CLIENTE**, **PRODUTO** e **FATURA**.

O método *gerarFatura()* deve conter uma instância da classe **FATURA**, relacionando-a ao respectivo atributo de referência. O atributo *valorTotal* da classe **FATURA** não deve ser informado pelo usuário no construtor desta classe e sim calculado, multiplicando-se o atributo *quantidade* da classe **PEDIDO** pelo atributo *precoUnitario* da classe **PRODUTO**.

CLASSE COMPRA

A classe **COMPRA** contém o método **MAIN**, dentro do qual deve-se instanciar uma das subclasses **PESSOAFISICA** ou **PESSOA JURIDICA** a partir da escolha feita pelo usuário através do uso do componente gráfico *JOptionPane*.