

**Exercício – Polimorfismo**

1) As lojas de comércio eletrônico podem oferecer várias opções de entrega, cada qual com custos específicos. Crie uma hierarquia de herança para representar vários tipos de pacotes a serem entregues aos clientes.

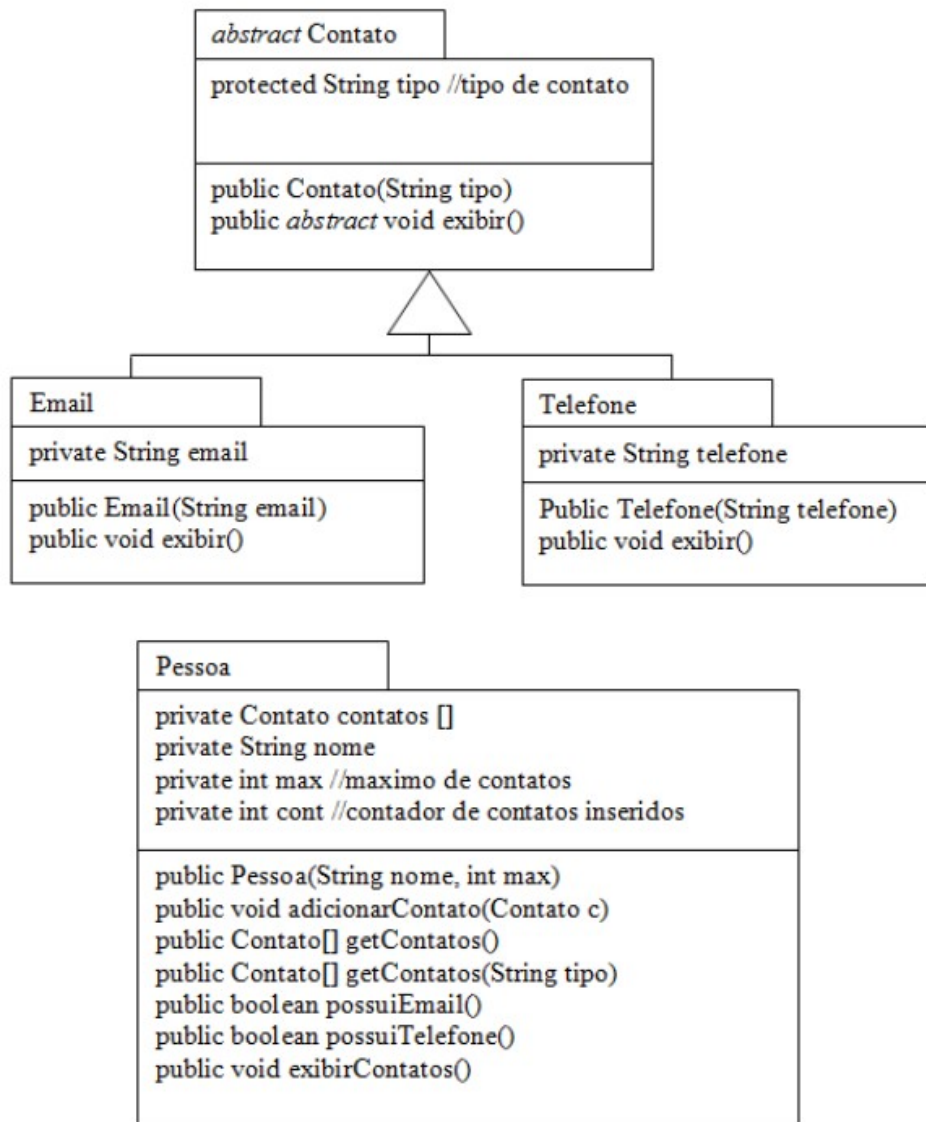
Utilize Pacote como a classe base da hierarquia, então inclua as classes PacoteEntregaNormal e PacoteEntregaExpressa que derivam de Pacote. A classe base Pacote deve ser abstrata e incluir atributos que representam nome, endereço, cidade, estado e CEP tanto do remetente como do destinatário do pacote, além dos membros de dados que armazenam o peso (em quilos) e o custo por quilo para a entrega do pacote. O construtor de Pacote deve inicializar esses atributos. Assegure que o peso e o custo por quilo contêm valores positivos. Pacote deve fornecer um método público calcularCusto que retorna um float indicando o custo associado com a entrega do pacote. O método calcularCusto de Pacote deve determinar o custo multiplicando o peso pelo custo.

A subclasse PacoteEntregaNormal deve herdar a funcionalidade da classe base Pacote, mas também incluir um atributo que representa uma taxa fixa que a empresa de entrega cobra pelo serviço de entrega normal. O construtor de PacoteEntregaNormal deve receber um valor para inicializar esse membro de dados. PacoteEntregaNormal deve redefinir o método calcularCusto para que ela calcule o custo de entrega adicionando a taxa fixa ao custo baseado em peso, calculado pelo método calcularCusto da classe base Pacote.

A classe PacoteEntregaExpressa deve herdar diretamente da classe Pacote e conter um atributo adicional para representar uma taxa adicional por quilo, que é cobrada pelo serviço de entrega mais rápido. PacoteEntregaExpressa deve redefinir o método calcularCusto para que ele acrescente a taxa adicional por quilo ao custo-padrão por quilo antes de calcular o custo da entrega.

Use a hierarquia de Pacote para criar um programa que exibe as informações de endereço e calcula os custos de entrega de vários pacotes. O programa deve conter um vetor do tipo Pacote, que seja inicializado com referências para objetos das classes PacoteEntregaNormal e PacoteEntregaExpressa. Faça um loop pelo vetor para processar os pacotes polimorficamente. Para cada Pacote, invoque as funções get para obter as informações de endereço do remetente e do destinatário, e então imprima os dois endereços da maneira que apareceriam nos pacotes de correio. Além disso, chame o método calcularCusto de cada objeto e imprima o resultado. Monitore o custo de entrega total de todos os pacotes no vetor e exiba esse total quando o loop terminar.

2) Os diagramas abaixo representam classes de parte de um programa de uma agenda que armazena pessoas. Uma pessoa pode ter vários contatos, ou seja, pode ter vários números de telefone ou de email.



Breve descrição dos métodos da classe **Pessoa**:

- O método **Contato[] getContatos()** retorna o vetor de contatos por completo.
- O método **Contato[] getContatos(String tipo)** retorna um vetor com os contatos do tipo especificado.
- O método **boolean possuiEmail()** – verifica se a pessoa tem algum contato do tipo email, retornando verdadeiro ou falso.
- O método **boolean possuiTelefone()** – verifica se a pessoa tem algum contato do tipo telefone, retornando verdadeiro ou falso
- O método **void exibirContatos()** – exibe o nome da pessoa e os seus contatos, ou seja, todos os números de telefone e endereços de email

O programa também deve armazenar uma lista de pessoas em uma agenda. Para isso, implemente uma classe Agenda que é uma abstração de uma agenda de pessoas, as quais são representadas pela classe Pessoa. O diagrama da classe Agenda e algumas de suas especificações estão descritos a seguir.

Agenda	
<pre>private String proprietario ///dono da agenda private Pessoa pessoas[] ///vetor de pessoas private int max /// número máximo de pessoas private int cont //contator de pessoas inseridas na agenda</pre>	
<pre>public Agenda(String proprietario, int max) public void addPessoa(Pessoa p) public Pessoa buscarPessoa(String nome) public void addContato(String nome, Contato c) public void exibirTodasPessoas() public void exibirContatosPessoa(String nome) public void exibirPessoasComEmail() public void recuperarPessoaPorEmail(String email) public void recuperarPessoaPorTelefone(String telefone)</pre>	

Breve descrição dos métodos da classe **Agenda**:

- O método **buscarPessoa(String nome)** retorna a pessoa que possui o nome passado como parâmetro. Se não encontrar, deve retornar null. Esse método deve ser utilizado sempre que for necessário realizar uma busca no programa.
- O método **addContato(String nome, Contato c)** faz uma busca para encontrar a pessoa que tem o nome passado por parâmetro, se encontrar, adiciona o contato c à pessoa que possui o nome passado como parâmetro; caso contrário, se a pessoa não estiver na agenda, não se faz nada. **Utilize** o método **buscarPessoa(String nome)** já implementado para encontrar a pessoa.
- O método **exibirTodasPessoas()** exibe os dados das pessoas cadastradas na agenda e seus contatos.
- O método **exibirContatosPessoa(String nome)** exibe os contatos da pessoa cujo nome é aquele passado no parâmetro nome.
- O método **exibirPessoasComEmail()** exibe somente o nome das pessoas que possuem email.

Os métodos **recuperarPessoaPorEmail** e **recuperarPessoaPorTelefone** passam, respectivamente, um email ou um telefone por parâmetro. Esse método busca tal informação em cada uma das pessoas cadastradas (inseridas) na agenda. O método deve exibir o nome das pessoas que possuem o email (no caso do método **recuperarPessoaPorEmail**) ou o telefone (no caso do método **recuperarPessoaPorTelefone**).