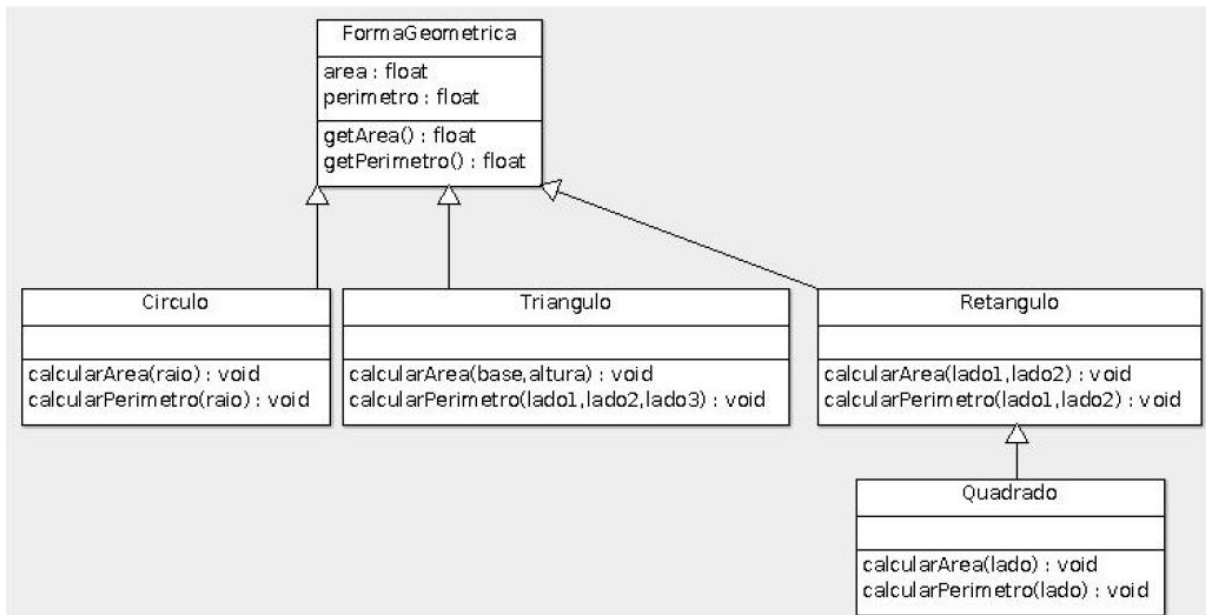


Exercícios - Classes abstratas, interfaces, classes internas, enum e genéricos

1. Escreva um programa em Java em conformidade com o diagrama apresentado na figura abaixo, usando herança e classes abstratas



2. Há uma preocupação atual com as pegadas de carbono (*carbon footprints*, emissões anuais de gás carbônico na atmosfera) a partir de instalações que queimam vários tipos de combustíveis para aquecimento, veículos que queimam combustíveis para se mover, e assim por diante. Nesse cenário:
 - a. Crie três pequenas classes não relacionadas por herança - classes **Construcao**, **Carro** e **Bicicleta**. Dê a cada classe alguns atributos e comportamentos (métodos) únicos que ela não tem em comum com as outras classes. Sugestões:
 - **Construcao**: número de pessoas (int), uso de energia renovável (boolean), número de lâmpadas (int), uso de ar-condicionado (boolean).
 - **Carro**: combustível (string), cilindrada (float).
 - b. Escreva uma interface **CarbonFootprint** com um método `getCarbonFootprint`. Faça cada uma das suas classes implementar essa interface, para que seu método `getCarbonFootprint` calcule uma pegada de carbono apropriada a cada classe (usando os atributos sugeridos ou outros – consulte na internet como calcular).

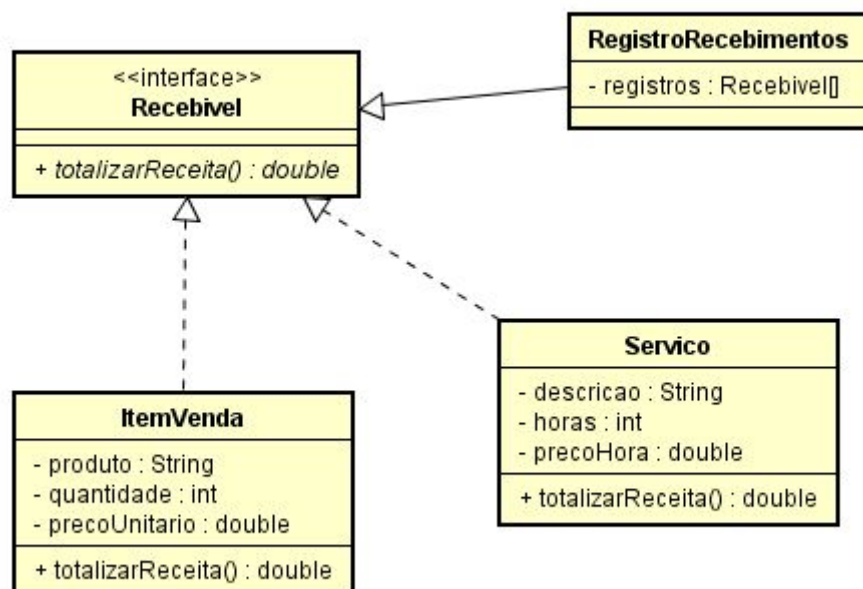
Escreva um programa que crie 2 objetos de cada uma das três classes. Crie um objeto `ArrayList<CarbonFootprint>` e insira as referências dos objetos instanciados nessa lista. Finalmente, itere pela lista, chamando polimorficamente o método `getCarbonFootprint` de cada objeto. Para cada objeto, imprima algumas informações de identificação e as pegadas de carbono do objeto.

3. Modifique o exercício 2, tornando `Edificio` uma classe abstrata e implementando as duas novas subclasses concretas `Casa` e `Escola`.
 - a. O programa que cria a lista de objetos vai continuar funcionando após a modificação na estrutura de classes? (responda como comentário no código)
 - b. Modifique o programa para que passe a instanciar diretamente objetos `Casa` e `Escola`, incluindo-os na lista.

4. Uma empresa paga seus funcionários semanalmente. Os funcionários são de quatro tipos: Funcionários assalariados recebem salários fixos semanais independentemente do número de horas trabalhadas, funcionários que trabalham por hora são pagos da mesma forma e recebem horas extras (isto é, 1,5 vezes sua taxa de salário por hora) por todas as horas trabalhadas além das 40 horas normais, funcionários comissionados recebem uma porcentagem sobre suas vendas e funcionários assalariados/ comissionados recebem um salário-base mais uma porcentagem sobre suas vendas. Para o período salarial atual, a empresa decidiu recompensar os funcionários assalariados/comissionados adicionando 10% aos seus salários-base. A empresa quer escrever um programa em Java que realiza os cálculos da folha de pagamento polimorficamente.

5. Você deve desenvolver um sistema para empresas que prestam serviços de manutenção de computadores e que comercializam peças e equipamentos de informática. O sistema precisa permitir que sejam registrados todos os valores recebidos dos clientes, tanto os relativos à prestação de serviços quanto os oriundos de vendas de mercadorias. Construa este sistema em conformidade com a estrutura apresentada na figura abaixo.

OBS: A classe `RegistroRecebimentos` é a classe executável do sistema e deverá permitir o registro de recebimentos e a impressão da lista desses registros (coloque como opções no menu para o usuário decidir). A classe `RegistroRecebimentos` deverá criar instâncias das classes `ItemVenda` e `Servico` para representarem cada recebimento realizado e deverá gravá-las no atributo `registros`. A interface `Recebivel` deve ser utilizada como tipo na declaração desse atributo.



6. Crie uma enumeração `DiaDaSemana` com sete valores entre `DOMINGO` e `SABADO`. Adicione um método `ehDiaUtil()` à classe `DiaDaSemana` que retorna `true` se o valor em que for chamado estiver entre `SEGUNDA` e `SEXTA` e `false`, caso contrário.
7. Escreva uma classe genérica `Par` que tem dois parâmetros de tipo - `F` e `S` - cada um representando o tipo do primeiro e segundo elemento do par, respectivamente. Adicione os métodos `get` e `set` ao primeiro e ao segundo elemento do par (Dica: o cabeçalho da classe deve ser `public class Par<F, S >`)
8. Defina uma classe que contenha um método genérico capaz de efetuar a soma das áreas de arrays contendo apenas objetos dos tipos apresentados no exercício 1.