

**Questão 1.** Defina os tipos abstratos de dados `Periférico`, `PeriféricoDeEntrada`, `PeriféricoDeSaída`. Um periférico é um dispositivo que possui um nome, uma marca um modelo e um tipo de conexão. Um periférico de entrada deve permitir a transmissão de dados do meio externo para o computador, isso será feito através de um método chamado `entrada` que deve receber um `Object` como parâmetro. Um periférico de saída deve permitir a transmissão de dados do computador para o meio externo, isso será feito através de um método chamado `saida` que deve receber um `Object` como parâmetro.

Obs.: Os métodos `entrada` e `saida` são a interface de comunicação. Por exemplo, se o computador precisa imprimir uma folha numa impressora ele chamaria o método `saida` de um periférico adequado passando como parâmetro um objeto que representasse a folha que ele deseja imprimir. Caso o usuário deseje enviar um comando pelo teclado ao computador ele deve chamar o método `entrada` do periférico adequado passando como parâmetro um objeto que representasse a sequência de teclas digitadas.

Obs.: Sempre que um dispositivo for ligado ele deve se identificar, por exemplo, ao ligar uma impressora ela deve imprimir uma mensagem na tela do tipo : "Ligando impressora MARCA MODELO conectada a Porta Pararela"

**Questão 2.** Implemente o diagrama de classes abaixo.

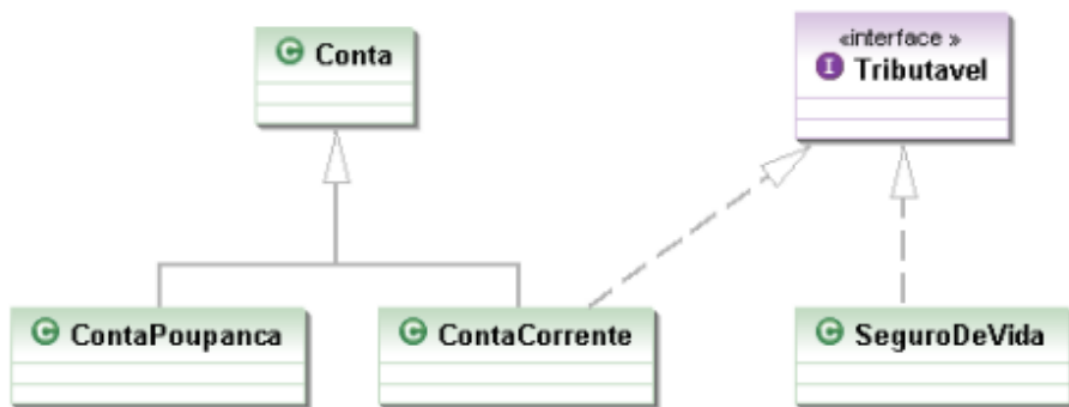


Figura 1: Diagrama de Classes.

- Crie uma interface `Tributavel` que possui o método `calculaTributos()`, que retorna um `double`.
- Alguns bens são tributáveis e outros não, `ContaPoupanca` não é tributável, já para `ContaCorrente` você precisa pagar 1% da conta e o `SeguroDeVida` tem uma taxa fixa de 42 reais.
- As classes `ContaCorrente` e `ContaPoupanca` herdam de uma classe `Conta`. Essa classe `Conta` possui um saldo e os métodos `sacar(double)`, `depositar(double)` e `obterSaldo()` que retorna o saldo da conta.
- Vamos criar uma classe `TestaTributavel` com um método `main` para testar o nosso exemplo.

**Questão 3.** Crie um `GerenciadorDeImpostoDeRenda`, que recebe todos os tributáveis de uma pessoa e soma seus valores e inclua nele um método para devolver seu total. Essa classe deve ter um atributo para calcular a soma total dos tributos e um método `adicionar(Tributavel)` que recebe como parâmetro um `Tributavel` e soma os tributos dele ao total. Crie um `main` para instanciar diversas classes que implementam `Tributavel` e passar como argumento para um `GerenciadorDeImpostoDeRenda`. Repare que você não pode passar qualquer tipo de conta para o método `adiciona`, apenas as que implementam `Tributavel`.