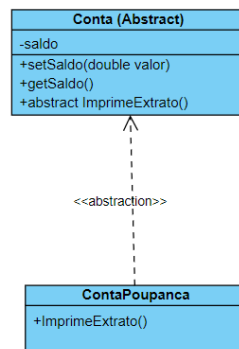


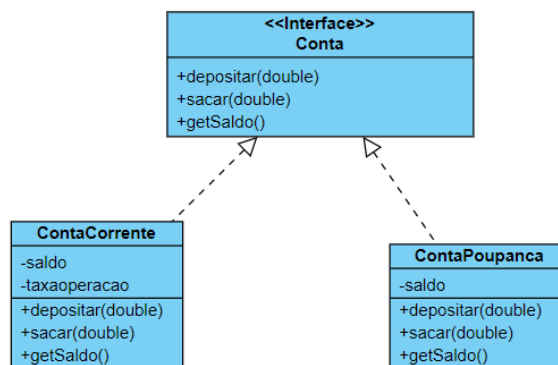
*Obs.: Em todos os programas, utilize o **Tratamento de Exceção** para a leitura de dados do teclado em qualquer outro trecho programa/operação que possa acontecer uma exceção.*

- 1) Utilizando o conceito de Classe abstrata, implemente um programa que leia um vetor de 5 valores que serão armazenados em uma conta poupança. Em seguida, imprima um relatório com os dados armazenados.



- 2) Utilizando o conceito de Interface, implemente um programa que leia um vetor de 5 saldos para armazenar em Conta Corrente e outro vetor para armazenar 5 saldos para armazenar em uma Conta Poupança.

O programa deve apresentar um menu com as operações disponíveis: Depositar | Sacar | Mostrar saldo. Para cada operação, o usuário deverá escolher o tipo de conta (Corrente ou Poupança) e escolher o número do cliente para o qual será realizada a operação. Obs.: o número do cliente deve ser a posição ocupada nos vetores Conta Corrente e Conta Poupança.

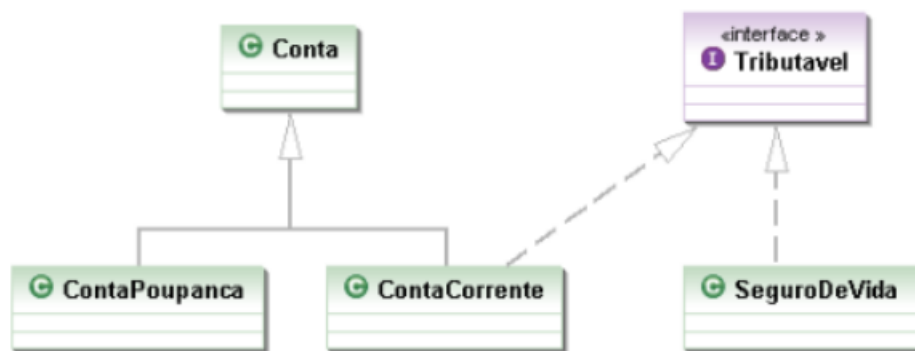


3) Considere a seguinte interface:

```
public interface IProduto {  
    public String getNome();  
    public float getCusto();  
}
```

O método `getNome` retorna o nome do produto e o método `getCusto` retorna o valor de compra do produto (este valor representa seu custo). Implemente uma classe `Produto` dessa interface para armazenar o nome e o valor de um produto. Em seguida, no programa principal, solicite que o usuário cadastre 5 produtos e apresente um relatório dos dados.

4) Nosso banco precisa tributar dinheiro de alguns bens que nossos clientes possuem. Para isso, vamos criar um sistema para isso.



- 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 Abstrata `Conta`. Essa classe `Conta` possui um saldo e os métodos `sacar(double)`, `depositar(double)` e `obterSaldo()` que retorna o saldo da conta.
- Crie uma classe principal para testar o funcionamento das classes.
- Obs.: definir os métodos e os atributos que devem ser implementados!**