

## DESCRIÇÃO DO DESAFIO

### DICAS EXTRAS

- ☐ Crie 1 interface 1 classe abstrata e 2 classes concretas
- ☐ Adicione um atributo **saldo** (double) e um construtor para inicializar o Adicione um atributo **saldo** (double) e um construtor para inicializar o saldo..
- ☐ Implemente o método `consultarSaldo()` para retornar o saldo atual.
- ☐ Deixe o método **`depositar(double valor)`** abstrato para que as subclasses possam implementar de forma específica.

### DESCRIÇÃO DO DESAFIO

Você vai criar um programa para gerenciar diferentes tipos de contas bancárias no Banco de Konoha. O objetivo é usar classes abstratas e interfaces para definir comportamentos comuns e específicos para diferentes tipos de contas.

1. Crie uma **interface** chamada **Conta** com os métodos **`consultarSaldo()`** e **`depositar(double valor)`**.
2. Crie uma **classe abstrata** chamada **ContaBancaria** que implementa a interface **Conta**.
3. Crie uma classe chamada **ContaCorrente** que estende a classe **ContaBancaria**.
4. Implemente o método **`depositar(double valor)`** para adicionar o valor ao saldo.
5. Crie uma **classe** chamada **ContaPoupanca** que também estende a classe **ContaBancaria**.
6. Implemente o método **`depositar(double valor)`** para adicionar o valor ao saldo com uma pequena taxa de depósito (por exemplo, **deduzir 1% do valor depositado**).

## QUER MAIS DIFICULDADE?

Como desafio adicional, adicione um **enum** chamado **TipoConta** para representar diferentes tipos de contas, como **CORRENTE** e **POUPANCA**. Modifique as classes `ContaCorrente` e `ContaPoupanca` para usar esse enum e adicione um método sobrecarregado na classe `BancoKonoha` para realizar transferências entre contas.