

Projeto : Banco Simplificado

1. Introdução: Este projeto implementa um Sistema Bancário simplificado, aplicando os padrões de projeto Singleton e Abstract Factory. O sistema permite a criação e manipulação de contas correntes e poupanças em um banco fictício.

2. Classes:

- **ContaBancaria:** Uma classe abstrata que representa uma conta bancária genérica. Contém métodos para depósito, saque e obtenção do saldo da conta.
- **ContaCorrente:** Subclasse concreta de ContaBancaria que representa uma conta corrente.
- **ContaPoupanca:** Subclasse concreta de ContaBancaria que representa uma conta poupança.
- **ContaBancariaFactory:** Uma fábrica de objetos ContaBancaria que implementa os padrões Singleton e Abstract Factory. Responsável por criar instâncias de ContaCorrente e ContaPoupanca conforme necessário.
- **Banco:** A classe principal do sistema, que faz uso da ContaBancariaFactory para criar e gerenciar contas bancárias. Esta classe contém o método main onde o sistema é inicializado.

3. Padrões de Projeto:

- **Singleton (ContaBancariaFactory):** A classe ContaBancariaFactory garante que apenas uma instância seja criada durante a execução do programa. Isso é alcançado mantendo uma instância privada e estática da própria classe e fornecendo um método estático getInstance() para recuperar essa instância única.
- **Abstract Factory (ContaBancariaFactory):** A ContaBancariaFactory atua como uma fábrica abstrata para criar objetos ContaBancaria. Ela fornece um método abstrato criarConta(), que é implementado por subclasses concretas para criar instâncias específicas de ContaBancaria (ContaCorrente e ContaPoupanca).

4. Funcionalidades:

O sistema permite as seguintes funcionalidades básicas:

- Criar uma conta corrente ou poupança.
- Realizar depósitos em uma conta.
- Realizar saques de uma conta.
- Obter o saldo de uma conta.

5. Utilização:

Para utilizar o sistema, basta executar a classe Banco. Isso inicializará o sistema, que então permitirá a criação e manipulação de contas bancárias. Nesse caso, foi criado manipulações de testes para visualização da execução das operações.

6. Diagrama de Classes

