SIMULANDO OPERAÇÕES BANCÁRIAS ORIENTADAS A OBJETO

Passo 1: Criar uma classe abstrata chamada Operacao com o atributo valor do tipo double e um método abstrato chamado operar() que retora um valor do tipo double.

Passo 2: Crie uma classe Debito e outra Credito que herda as caraterísticas de Operacao. O construtor de Debito e Credito deve receber o valor da operação e atribuir este valor a variável definida em Operacao (superclasse).
Estas classes (Debito e Credito) devem ter um método operar() que deve ser programado de acordo com sua finalidade: operar() da classe Debito retorna o valor negativo pois a operação deve ser um debito enquanto a o método operar() de Credito retorna o valor positivo.

Passo 3: Criar a classe ContaCorrente com o atributo valor do tipo double que inicia com 0. Esta classe possui um método executarOperacao(Operacao opr) que recebe um parâmetro do tipo Operacao que vai operar o valor da conta correte (se for debito diminui, se for credito soma). Esta classe também possui o método getSaldo() que retorna o valor do saldo atual.

Passo 4: Crie a classe Correntista com os seguintes atributos: nome (do tipo String) e conta (do tipo ContaCorrente). O construtor de Correntista deve receber seu nome. A classe deve ter 2 métodos: public String getNome() e public ContaCorrente getContacorrente(). Estes métodos retornam o nome e a conta respectivamente.

Passo 5: Crie a classe Banco como descrito no código abaixo:

```
public class Banco {
    Correntista c1,c2,c3;


public Banco(String correntista1, ContaCorrente co1,
String correntista2, ContaCorrente co2, String
correntista3, ContaCorrente co3) {
        c1 = new Correntista(correntista1,co1);
        c2 = new Correntista(correntista2,co2);
        c3 = new Correntista(correntista3,co3);
}


public Correntista getCorrentista(String nome){
        if(c1.getNome().equals(nome)){
            return c1;
        }
        if(c2.getNome().equals(nome)){
```

```
            return c2;
        }
        if(c3.getNome().equals(nome)){
            return c3;
        }
        return null;
}


public void debitar(String nome_correntista, double
valor){
        Debito d = new Debito(valor);

getCorrentista(nome_correntista).getContaCorrente().execu
tarOperacao(d);
    }


public void creditar(String nome_correntista, double
valor){
        Credito c = new Credito(valor);

getCorrentista(nome_correntista).getContaCorrente().execu
tarOperacao(c);
}


    public double getSaldo(String nome_correntista){
        return
getCorrentista(nome_correntista).getContaCorrente().getSa
ldoAtual();
    }


    public void transferir(String
nome_correntista_origem, String nome_correntista_destino,
double valor){
        debitar(nome_correntista_origem, valor);
        creditar(nome_correntista_destino, valor);
    }

}
```

Passo 6: Crie uma classe APP que faça uso das operações definidas na classe Banco.