

```
package contas;

public abstract class Contas {

    private int numConta;
    private String titular;
    protected double saldo;

    public int getNumConta() {
        return numConta;
    }

    public void setNumConta(int numConta) {
        this.numConta = numConta;
    }

    public String getTitular() {
        return titular;
    }

    public void setTitular(String titular) {
        this.titular = titular;
    }

    public double getSaldo() {
        return saldo;
    }

    public abstract void depositar(double valor);

    public abstract void dadosDaConta();

    public abstract void mostrarSaldo();

}
```

```

package contas;

public class ContaCorrente extends Contas{

    public ContaCorrente() {
    }

    public ContaCorrente(String nome, int numero, double saldoInicial) {
        setTitular(nome);
        setNumConta(numero);
        depositar(saldoInicial);
    }

    public ContaCorrente(String nome, int numero) {
        setTitular(nome);
        setNumConta(numero);
    }

    public void sacar(double valor) {
        if (this.getSaldo() < valor) {
            System.out.println("Saque nao Realizado");
        } else {
            this.saldo = this.saldo - valor;
            System.out.println("Saque Realizado");
        }
    }

    @Override
    public void depositar(double valor) {
        this.saldo = this.getSaldo() + valor;
    }

    @Override
    public void dadosDaConta() {
        System.out.println("+-----+");
        System.out.println("|          DADOS DA CONTA          |");
        System.out.println("| Titular Conta Corrente: " + this.getTitular());
        System.out.println("| Numero da Conta: " + this.getNumConta());
        System.out.println("| Saldo atual: " + this.getSaldo());
        System.out.println("+-----+");
    }

    @Override
    public void mostrarSaldo() {
        System.out.printf("Saldo Atual Conta Corrente: %.2f\n", this.getSaldo());
    }

    public void aplicar(double valor, Poupanca destino) {
        if (this.getSaldo() >= valor) {
            this.saldo = this.getSaldo() - valor;
            destino.depositar(valor);
        }
    }

}

```

```
package contas;

public class Poupanca extends Contas {

    public Poupanca() {

    }

    public Poupanca(String nome, int numero) {
        setTitular(nome);
        setNumConta(numero);
    }

    public void resgatar(double valor, ContaCorrente destino) {
        if (getSaldo() >= valor) {
            this.saldo = getSaldo() - valor;
            destino.depositar(valor);
        } else {
            System.out.println("saldo insuficiente: Saldo atual: " + getSaldo());
        }
    }

    @Override
    public void depositar(double valor) {
        this.saldo = this.getSaldo() + valor;
    }

    @Override
    public void dadosDaConta() {
        System.out.println("+-----+");
        System.out.println("| Titular Poupança: " + getTitular());
        System.out.println("| Numero da Conta Poupança: " + getNumConta() + "-1");
        System.out.println("| Saldo atual: " + getSaldo());
        System.out.println("+-----+\n");
    }

    @Override
    public void mostrarSaldo() {
        System.out.println("Saldo atual da Poupança: " + getSaldo());
    }

}
```

```

package projetoBanco;

import java.util.Scanner;

import contas.ContaCorrente;
import contas.Poupanca;

public class Banco {

    // private static float valor;
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        Scanner inputString = new Scanner(System.in);
        ContaCorrente cc = new ContaCorrente();

        Poupanca cp = new Poupanca();

        String nome;
        int num, opcao;
        double valor, depositoInicial = 0;

        do {
            System.out.println("+=====+");
            System.out.println("|      *****Fuctura Bank*****      |");
            System.out.println("|          MENU DE OPÇÕES          |");
            System.out.println("|      1 - Criar Conta              |");
            System.out.println("|      2 - Dados da Conta          |");
            System.out.println("|      3 - Depositar valor         |");
            System.out.println("|      4 - Sacar valor              |");
            System.out.println("|      5 - Aplicar                  |");
            System.out.println("|      6 - Resgatar                 |");
            System.out.println("|      7 - Mostrar saldo           |");
            System.out.println("|      0 - Sair                     |");
            System.out.println("|      *****                     |");
            System.out.println("|          Escolha uma opção:      |");
            System.out.println("+=====+");

            opcao = input.nextInt();

            switch (opcao) {

                case 1: //Criar Conta
                    System.out.println("Nome titular da conta: ");
                    nome = inputString.nextLine();

                    System.out.println("Entre com o Numero da conta: ");
                    num = input.nextInt();

                    if (depositoInicial <= 0) {
                        System.out.println("Seu saldo é: " + depositoInicial);
                        System.out.println("Deseja realizar um depósito? 1-SIM / 2-NÃO");
                        int resposta = input.nextInt();

                        if (resposta == 1) {

```

```
        System.out.print("Digite um valor inicial: ");
        depositoInicial = input.nextDouble();
        cc = new ContaCorrente(nome, num, depositoInicial);
        cp = new Poupanca(nome, num);
    } else {
        cc = new ContaCorrente(nome, num);
        cp = new Poupanca(nome, num);
    }

    } else {
        System.out.println("Seu saldo é: " + depositoInicial + ", não precisa de
depósito inicial.");
    }

    break;

case 2:
    cc.dadosDaConta();
    cp.dadosDaConta();
    break;

case 3:
    System.out.println("Entre com o valor a ser depositado: ");
    valor = input.nextDouble();
    cc.depositar(valor);
    break;

case 4:
    cc.mostrarSaldo();
    cp.mostrarSaldo();
    System.out.println("Entre com o valor a ser sacado: ");
    valor = input.nextDouble();
    cc.sacar(valor);
    break;

case 5:
    System.out.println("Insira um valor para transferir para Poupança: ");
    valor = input.nextDouble();
    cc.aplicar(valor, cp);
    break;

case 6:
    System.out.println("Insira um valor para resgatar: ");
    valor = input.nextDouble();
    cp.resgatar(valor, cc);
    break;

case 7:
    cc.mostrarSaldo();
    cp.mostrarSaldo();
    break;

default:
    System.out.println("Fim");
```

```
    }  
  
    } while (opcao != 0);  
  
    input.close();  
    inputString.close();  
  
    }  
}
```