

## EXERCÍCIOS DA CADEIRA PP

### 1. CLASSE CONTA

```
public class Conta {  
  
    private int numero;  
    private String titular;  
    private double saldo;  
  
    public Conta(int numero, String titular, double saldo) {  
        this.numero = numero;  
        this.titular = titular;  
        this.saldo = saldo;  
    }  
  
    public int getNumero() {  
        return numero;  
    }  
  
    public void setNumero(int numero) {  
        this.numero = numero;  
    }  
  
    public String getTitular() {  
        return titular;  
    }  
  
    public void setTitular(String titular) {  
        this.titular = titular;  
    }  
  
    public double getSaldo() {  
        return saldo;  
    }  
  
    public void setSaldo(double saldo) {  
        this.saldo = saldo;  
    }  
  
    @Override  
    public String toString() {  
        return "Conta [numero=" + numero + ", titular=" + titular + ", saldo=" + saldo + "];"  
    }  
  
    public void depositar(double valor)  
    {  
        this.setSaldo(this.getSaldo()+valor);  
    }  
  
    public double verificarSaldoReal(double valorDeposito)  
    {  
        return this.getSaldo();  
    }  
  
}
```

---

## 2.CONTA ORDEM

---

```
public class ContaOrdem extends Conta {  
    private String RegimeTitularidade;  
  
    public ContaOrdem(int numero, String titular, double saldo,String rT) {  
        super(numero, titular, saldo);  
        this.RegimeTitularidade=rT;  
        // TODO Auto-generated constructor stub  
    }  
  
    @Override  
    public void depositar(double valor)  
    {  
        double comissao = 0.01*valor;  
        double valorDiminuido = valor-comissao;  
        this.setSaldo(this.getSaldo()+valorDiminuido);  
    }  
  
    @Override  
    public double verificarSaldoReal(double valorDeposito)  
    {  
        double valorOriginal= this.getSaldo()+(valorDeposito*1)/100;  
        return valorOriginal;  
    }  
}
```

### 3.CONTAPRAZO

```
public class ContaPrazo extends Conta {  
    ➤ public ContaPrazo(int numero, String titular, double saldo) {  
        super(numero, titular, saldo);  
        // TODO Auto-generated constructor stub  
    }  
  
    ➤ @Override  
    public void depositar(double valor)  
    {  
        double acrescimo=0.03*valor;  
        this.setSaldo(this.getSaldo()+valor+acrescimo);  
    }  
  
    ➤ @Override  
    public double verificarSaldoReal(double valor)  
    {  
        double valorOriginal= this.getSaldo()-(valor*3)/100;  
        return valorOriginal;  
    }  
}
```

#### 4. PRINCIPAL

```
public class Principal {  
    public static void main(String[] args) {  
        ContaOrdem co = new ContaOrdem(1234, "Tafari",0,"Solidaria");  
        ContaPrazo cp = new ContaPrazo(2181,"Stelia",0);  
  
        System.out.println("TAFARI VILANCULOS");  
        // 2. Alinea a)  
        co.depositar(1000);  
        cp.depositar(1000);  
  
        // 2. Alinea b)  
        System.out.println("-----");  
        System.out.println("Conta ordem:"+co.getSaldo());  
        System.out.println("Conta praz :"+cp.getSaldo());  
  
        System.out.println(" ");  
        // 2. Alinea c)  
        Conta c1 = co;  
        Conta c2= cp;  
  
        c1.depositar(1000);  
        c2.depositar(1000);  
  
        System.out.println(" ");  
        System.out.println("-----");  
        System.out.println("Conta ordem referenciada :"+c1.getSaldo());  
        System.out.println("Conta prazo referenciada :"+c2.getSaldo());  
  
        System.out.println(" ");  
        // EXERCICIO 3  
        System.out.println("-----");  
        System.out.println("Saldo Real Conta Ordem "+co.verificarSaldoReal(1000));  
        System.out.println("Saldo Real Conta Prazo "+cp.verificarSaldoReal(1000));  
  
    }  
}
```

---

## 4.BASEDADOS

```
import java.io.*;
public class BaseDados {

    public static void main(String[] args) throws IOException {

        ContaOrdem co = new ContaOrdem(1234, "Tafari", 0, "Solidaria");
        ContaPrazo cp = new ContaPrazo(2181, "Tembe", 0);

        co.depositar(1000);
        cp.depositar(1000);

        escrevaNoFicheiro(cp);
        escrevaNoFicheiro(co);

    }

    public static void escrevaNoFicheiro(Conta c) throws IOException
    {
        FileWriter fl = new FileWriter("contas.txt", true);
        BufferedWriter buffer = new BufferedWriter(fl);

        String linha = "Conta: " + c.getNumero() + " Saldo: " + c.getSaldo() + " Saldo real: " + c.verificarSaldoReal(1000);

        buffer.write(linha);
        buffer.newLine();
        buffer.close();
        fl.close();
    }

}
```