

Curso JAVA

Aula 13 – Métodos

(Eclipse IDE – aula7/pt6)

Código Métodos:

```
package contas;

public class Conta {
    public String cliente;
    public double saldo;

    public Conta() {
        System.out.println("Agência: 0261");
    }

    public void exibirSaldo() {
        System.out.println("Saldo: R$" + saldo);
    }

    void sacar(double valor) {
        saldo -= valor;
        System.out.println("Débito: R$" + valor);
    }

    void depositar(double valor) {
        saldo += valor;
        System.out.println("Crédito: R$" + valor);
    }

    void transferir(Conta destino, double valor) {
        this.sacar(valor);
        destino.depositar(valor);
        System.out.println("Transferencia: R$" + valor);
    }

    double soma(double cc1, double cc2) {
        double total = cc1 + cc2;
        return total;
    }
}

package contas;

public class PessoaFisica {
```

```

public static void main(String[] args) {
    Conta cc1 = new Conta();
    cc1.cliente = "Felps Rabudo";
    cc1.saldo = 10000;
    System.out.println("Cliente: " + cc1.cliente);
    cc1.exibirSaldo();
    cc1.sacar(1000);
    cc1.exibirSaldo();
    System.out.println("-----");

    Conta cc2 = new Conta();
    cc2.cliente = "Felps Gotico";
    cc2.saldo = 8500;
    System.out.println("Cliente: " + cc2.cliente);
    cc2.exibirSaldo();
    cc2.depositar(2000);
    cc2.exibirSaldo();
    System.out.println("-----");

    System.out.println("Transferencia");
    System.out.println("Cliente: " + cc1.cliente);
    System.out.println("Favorecido: " + cc2.cliente);
    cc1.transferir(cc2, 2000);
    System.out.println("");

    System.out.println("Cliente: " + cc1.cliente);
    cc1.exibirSaldo();
    System.out.println("Favorecido: " + cc2.cliente);
    cc2.exibirSaldo();
    System.out.println("-----");

    System.out.println("Relatório");
    Conta gerente = new Conta();
    double relatorio = gerente.soma(cc1.saldo,
cc2.saldo);
    System.out.println("Saldo total: R$" + relatorio);
}
}

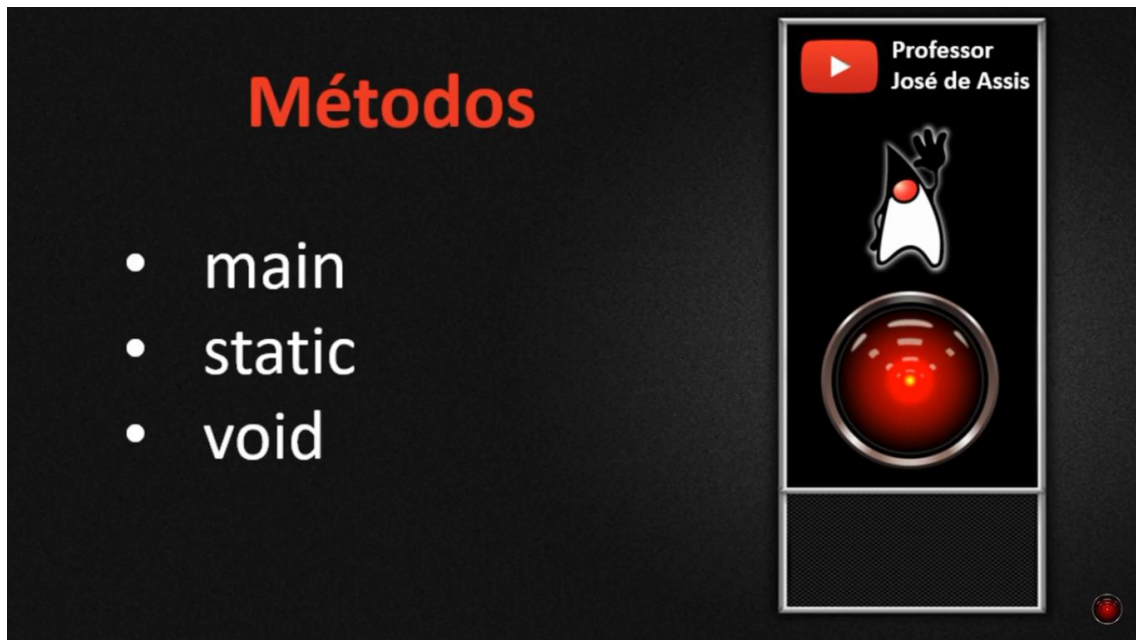
```

Criando métodos.

Comandos:

NULL

Comentários:



Static: Quando um método é definido com o static, ele pode ser invocado somente por outros métodos static.

Para criar um objeto dentro dos métodos, usa a classe modelo Conta e define o nome para o novo objeto:

Classe Conta:

```
void transferir(Conta destino, double valor) {  
    this.sacar(valor);  
    destino.depositar(valor);  
    System.out.println("Transferencia: R$" + valor);  
}
```

Classe PessoaFisica:

```
System.out.println("Transferencia");  
System.out.println("Cliente: " + cc1.cliente);  
System.out.println("Favorecido: " + cc2.cliente);  
cc1.transferir(cc2, 2000);  
System.out.println("");  
  
System.out.println("Cliente: " + cc1.cliente);  
cc1.exibirSaldo();  
System.out.println("Favorecido: " + cc2.cliente);  
cc2.exibirSaldo();
```

```
System.out.println("-----");
```

Para criar um método com retorno, basta definir o tipo deste retorno e o nome pro método.

```
double soma(double cc1, double cc2) {  
    double total = cc1 + cc2;  
    return total;  
}
```