

Curso JAVA

Aula 14 – Encapsulamento

(Eclipse IDE – aula7/pt7)

Código Encapsulamento:

```
package contas;

public class Conta {
    private String cliente;
    //encapsulamento da variavel cliente
    public String getCliente() {
        return cliente;
    }

    public void setCliente(String cliente) {
        this.cliente = cliente;
    }
    private double saldo;
    //encapsulamento da variavel saldo
    public double getSaldo() {
        return saldo;
    }

    public void setSaldo(double saldo) {
        this.saldo = 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.setCliente("Felps Rabudo");
        cc1.setSaldo(10000); //usa o set para atribuir valor
        nessa variavel
        System.out.println("Cliente: " + cc1.getCliente());
        cc1.getSaldo();
        cc1.sacar(2000);
        cc1.exibirSaldo();
        System.out.println("-----");

        Conta cc2 = new Conta();
        cc2.setCliente("Felps Gotico");
        cc2.setSaldo(8500);
        System.out.println("Cliente: " + cc2.getCliente());
        cc2.getSaldo();
        cc2.depositar(2000);
        cc2.exibirSaldo();
        System.out.println("-----");

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

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

```

```

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

```

```
package seguros;
```

```
import contas.Conta;
```

```

public class SeguroPessoaFisica {

    public static void main(String[] args) {
        Conta cc3 = new Conta();
        cc3.setCliente("Felps Triste");
        cc3.setSaldo(9000);
        System.out.println("Cliente: " + cc3.getCliente());
        cc3.getSaldo();
        System.out.println("-----");
    }
}

```

Fazendo encapsulamento então... reforçando a segurança.

Comandos:

NULL

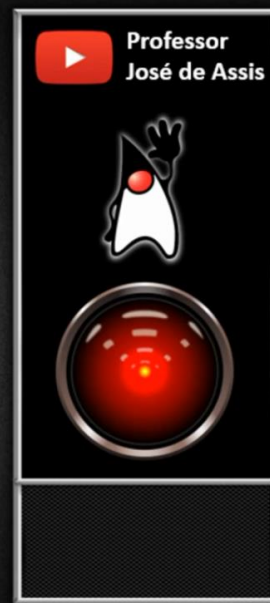
Comentários:

Encapsulamento

□ **private** String cliente;

▲ **getCliente**

▲ **setCliente**



A ideia principal é atribuir um modificador do tipo `private` a variável que você deseja proteger e criar 2 métodos públicos para que outras classes possam ter acesso a esta variável.

`getCliente`: Obtém conteúdo da variável.

```
public String getCliente() {  
    return cliente;  
}
```

`setCliente`: Atribui um novo valor a variável.

```
public void setCliente(String cliente) {  
    this.cliente = cliente;  
}
```