



Sejam bem-vindos!

PROGRAMAÇÃO ORIENTADA A OBJETOS

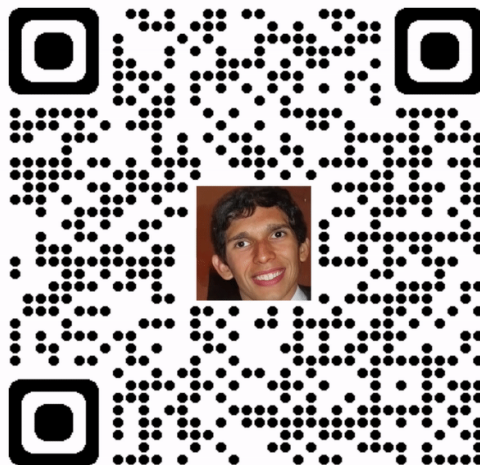


Diogenes Carvalho Matias

Formação:

- **Graduação: Sistemas de Informação;**
- **Especialista em: Engenharia e Arquitetura de Software;**
- **MBA EXECUTIVO EM BUSINESS INTELLIGENCE (em andamento);**
- **Mestrado Acadêmico em Engenharia de Computação (UPE em andamento);**

Maiores informações :



[Linkedin](#)

Programação Orientada a Objetos

Encapsulamento

“Objetos do mundo real encapsulam em si os próprios atributos, quer sejam descritivos, partes componentes ou funções.” (Meyer, 1997)

Programação Orientada a Objetos

Encapsulamento

É o mecanismo que une o código e os dados que ele manipula. Outra forma de pensar sobre o encapsulamento é que ele é um escudo protetor que evita que os dados sejam acessados pelo código fora desse escudo.

Programação Orientada a Objetos

Encapsulamento

Tecnicamente no encapsulamento, as variáveis ou dados de uma classe são ocultados de qualquer outra classe e podem ser acessados somente por meio de qualquer função de membro de sua própria classe na qual é declarada.

Programação Orientada a Objetos

Encapsulamento

Como no encapsulamento, os dados em uma classe são ocultados de outras classes usando o conceito de ocultação de dados que é obtido tornando os membros ou métodos de uma classe privados, e a classe é exposta ao usuário final ou ao mundo sem fornecer quaisquer detalhes por trás da implementação usando o conceito de abstração, por isso também é conhecido como uma **combinação de ocultação de dados e abstração** .

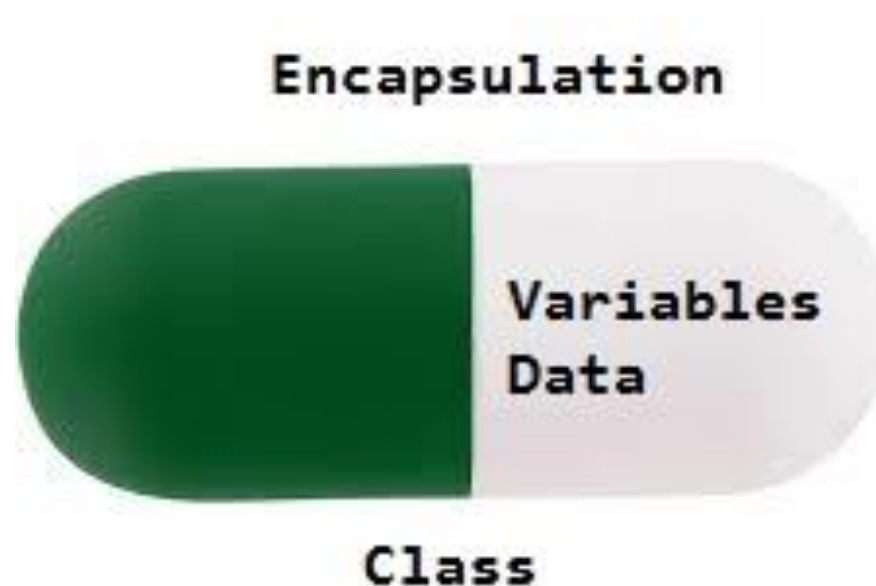
Programação Orientada a Objetos

Encapsulamento

- O encapsulamento pode ser alcançado declarando todas as variáveis na classe como privadas e escrevendo métodos públicos na classe para definir e obter os valores das variáveis.

Programação Orientada a Objetos

Encapsulamento



Programação Orientada a Objetos

Encapsulamento

```
public class Cliente {  
  
    private String nome;  
    private String endereco;  
    private String cpf;  
    private int idade;  
  
    public void mudaCPF(String cpf) {  
        if (this.idade <= 60) {  
            validaCPF(cpf);  
        }  
        this.cpf = cpf;  
    }  
    private void validaCPF(String cpf) {  
        //regras de negocio para podermos validar o CPF...  
    }  
}
```

Programação Orientada a Objetos

Encapsulamento

Níveis de Acesso:

- Privada: não visível a nenhuma classe externa; visível apenas dentro da classe.
- Pública: completamente visível internamente e para outras classes e elementos externos.
- Protegida: não visível a classes e elementos externos; visível apenas dentro da classe e para seus herdeiros.

Programação Orientada a Objetos

Encapsulamento

Níveis de Acesso:

Atributos Privados e Métodos Públicos:

- Na Programação orientada a objetos uma classe possui, em geral, os atributos privados e os métodos podem ser públicos, tornando o objeto como uma caixa preta onde só aparece o suficiente para que o programa possa utilizá-lo.

Programação Orientada a Objetos

Encapsulamento Níveis de Acesso

Atributos Privados e Métodos Públicos:

- Privada (private): visível exclusivamente dentro da classe.
- Pública (public): completamente visível dentro e fora da classe.
- Protegida (protected): visível apenas dentro da classe, pelos seus herdeiros e pelas classes no mesmo pacote.
- Pacote (padrão): visível apenas dentro da classe e pelas classes que estão no mesmo pacote.

Programação Orientada a Objetos

Encapsulamento Níveis de Acesso

Atributo final :

- Atributo que recebe o prefixo final;
- Só pode receber o valor uma vez na declaração ou no construtor;
- Depois que recebe o valor não pode ser modificado;
- Usualmente associado ao ***static*** para a criação de uma constante;

Programação Orientada a Objetos

Encapsulamento Níveis de Acesso

GETTERS E SETTERS

- Precisamos então arranjar uma maneira de fazer esse acesso. Sempre que precisamos arrumar uma maneira de fazer alguma coisa com um objeto, utilizamos de métodos!

Programação Orientada a Objetos

Encapsulamento Níveis de Acesso

GETTERS E SETTERS

```
public class Conta2 {  
  
    private double saldo;  
  
    public double pegaSaldo() {  
        return this.saldo;  
    }  
  
}
```

Programação Orientada a Objetos

Encapsulamento Níveis de Acesso

GETTERS E SETTERS

```
public class Conta2 {  
    private String titular;  
    private double saldo;  
    private double limite;  
    public double getSaldo() {  
        return this.saldo + this.limite;  
    }  
    public String getTitular() {  
        return this.titular;  
    }  
    public void setTitular(String titular) {  
        this.titular = titular;  
    }  
    void deposita(double quantidade){  
        this.saldo += quantidade;  
    }  
}
```


Programação Orientada a Objetos

Encapsulamento Níveis de Acesso

GETTERS E SETTERS

```
public class PrincipalConta {  
  
    public static void main(String[] args) {  
  
        Conta2 minhaConta = new Conta2();  
        minhaConta.deposita(1000);  
        System.out.println("Saldo:" + minhaConta.getSaldo());  
  
    }  
}
```

Programação Orientada a Objetos

Encapsulamento Níveis de Acesso

```
public class Pessoa {  
    private String nome;  
    private int idade;  
    private String endereco;  
    public String getNome() {  
        return nome;  
    }  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
    public int getIdade() {  
        return idade;  
    }  
    public void setIdade(int idade) {  
        this.idade = idade;  
    }  
    public String getEndereco() {  
        return endereco;  
    }  
    public void setEndereco(String endereco) {  
        this.endereco = endereco;  
    }  
}
```

GETTERS E SETTERS

Programação Orientada a Objetos

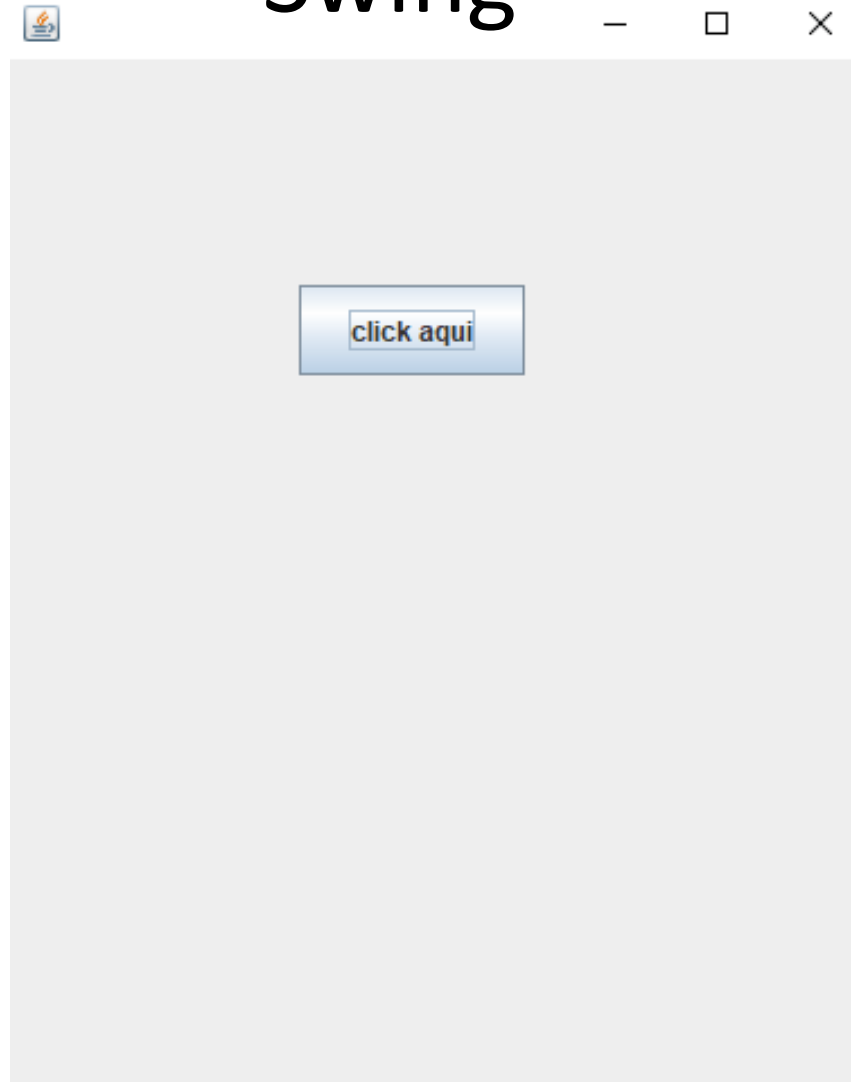
Encapsulamento Níveis de Acesso

GETTERS E SETTERS

```
public class PrincipalConta {  
  
    public static void main(String[] args) {  
  
        Pessoa objPessoa = new Pessoa();  
        objPessoa.setNome("Diogenes Carvalho Matias");  
        objPessoa.setIdade(35);  
        objPessoa.setEndereco("Avenida Gal Manoela Rabelo");  
  
        System.out.println("Meu nome é"+objPessoa.getNome());  
        System.out.println("Minha idade é :"+objPessoa.getIdade());  
        System.out.println("Meu endereço é"+objPessoa.getEndereco());  
  
    }  
}
```

Programação Orientada a Objetos

Swing



Programação Orientada a Objetos

Swing

```
import javax.swing.*;
public class EstudoSwing {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        JFrame janela=new JFrame();//criamos nosso objeto JFrame

        JButton botao01=new JButton("click aqui");//criamos nosso objeto JButton
        botao01.setBounds(130,100,100, 40);//posição :x , posição :y, Largura: width,Altura: height

        janela.add(botao01);//Adicionando nosso botao em nossa janela

        janela.setSize(400,500);//Largura 400 por 500 de altura
        janela.setLayout(null);//sem usar gerenciadores de layout
        janela.setVisible(true);//tornando nossa janela visível

    }
}
```

Programação Orientada a Objetos

Swing

```
import javax.swing.*;

public class Janela {

    JFrame janela;
    Janela(){
        janela=new JFrame();

        JButton botao=new JButton("click aqui");
        botao.setBounds(130,100,100, 40);

        janela.add(botao);

        janela.setSize(400,500);
        janela.setLayout(null);
        janela.setVisible(true);

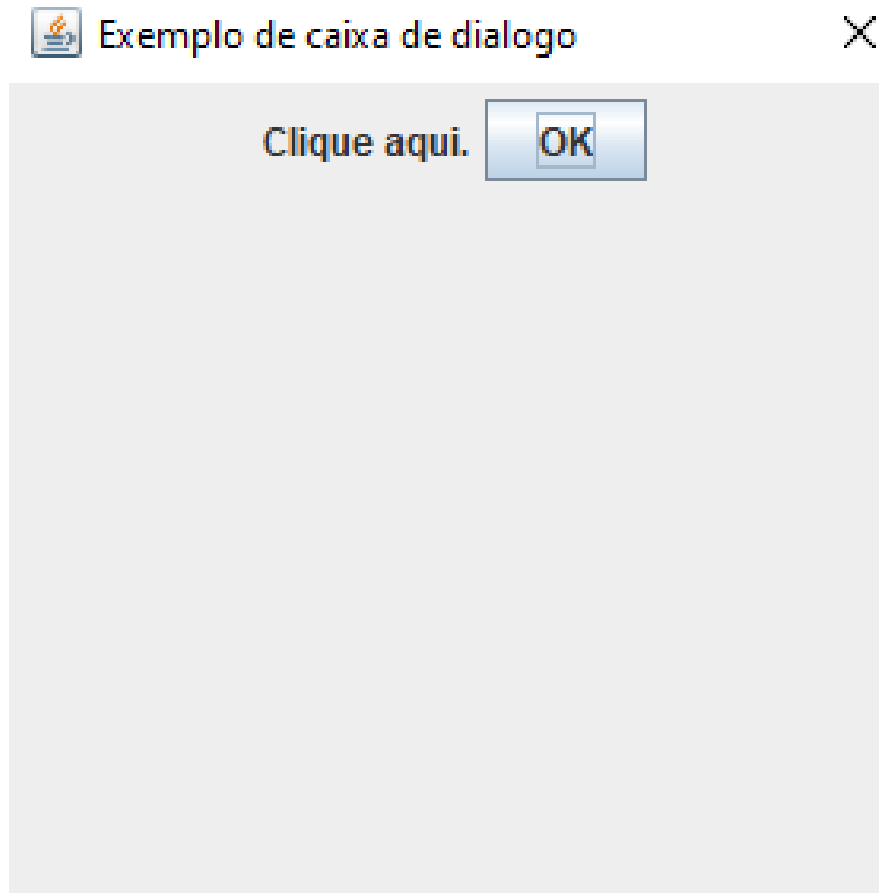
    }
}
```

Programação Orientada a Objetos

Swing

```
public static void main(String[] args) {  
  
    new Janela();  
  
}
```

Programação Orientada a Objetos Swing



Programação Orientada a Objetos Swing

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class CaixaDialogo {
    private static JDialog dialogo;
    CaixaDialogo() {
        JFrame janela= new JFrame();
        dialogo = new JDialog(janela , "Exemplo de caixa de dialogo", true);
        dialogo.setLayout( new FlowLayout() );
        JButton botao = new JButton ("OK");
        botao.addActionListener ( new ActionListener()
        {
            public void actionPerformed((ActionEvent e )
            {
                JOptionPane.showMessageDialog(janela,"Obrigado por continuar estudando.");
                CaixaDialogo.dialogo.setVisible(false);
            }
        });
        dialogo.add( new JLabel ("Clique aqui."));
        dialogo.add(botao);
        dialogo.setSize(300,300);
        dialogo.setVisible(true);
    }
}
```

Programação Orientada a Objetos Swing

```
public static void main(String[] args) {  
    new CaixaDialogo();  
  
}
```

Programação Orientada a Objetos

Exercício para ser feliz 😊

- Faça uma pesquisa sobre os demais componentes da biblioteca swing como: JButton, JLabel, JTextField, JTextArea, JCheckBox, JComboBox, JPanel

E seus exemplos práticos de uso, enviar para : prof.dcm.web@gmail.com em grupo de no máximo de 5 pessoas.