

Poo404:

Domine a Matrix do Backend com Java



Aprenda sobre os pilares da Programação Orientada a Objetos em Java.

Mauricio Rodrigues

Programação Orientada a Objeto

Despertando na Matrix da Programação

Bem-vindo à POO404. Aqui, você aprenderá a manipular o código da Matrix do backend usando Java e os poderes da Programação Orientada a Objetos. POO é a base da arquitetura moderna: mantém seu código organizado, reutilizável e preparado para o futuro.

O que é POO?

POO (Programação Orientada a Objetos) é um paradigma que organiza o software em torno de objetos, que representam coisas do mundo real.

Em Java, POO é o coração do sistema — cada variável, método ou classe existe para dar vida a um objeto.



01

Encapsulamento

Proteja o Código como um Firewall:

Encapsulamento:

Encapsular é proteger os dados dentro da classe, expondo apenas o necessário.

Isso evita acessos indevidos e garante controle total sobre as alterações.

```
Encapsulamento.Java

public class Usuario {
    private String senha;

    public void setSenha(String novaSenha) {
        if (novaSenha.length() ≥ 8) {
            this.senha = novaSenha;
        } else {
            System.out.println("A senha deve ter no mínimo 8 caracteres.");
        }
    }

    public String getSenha() {
        return "*oculto*";
    }
}
```

Contexto real: imagine um sistema de login. O encapsulamento garante que a senha do usuário nunca seja exposta diretamente.



02

Herança

Reutilize Código como um hacker inteligente.

HERANÇA:

Herança permite que uma classe “filha”(subclasse) herde atributos e métodos de uma classe “pai”(superclasse), ampliando ou modificando seu comportamento.

```
Encapsulamento.Java

public class Funcionario {
    double salario;

    public void trabalhar() {
        System.out.println("Trabalhando...");
    }
}

public class Programador extends Funcionario {
    public void programar() {
        System.out.println("Codando em Java...");
    }
}
```

Contexto real: uma empresa tem vários tipos de funcionários. Em vez de repetir código, você cria uma estrutura hierárquica com herança, usando “extends”.



03

Polimorfismo

Um método, Múltiplas formas.

Polimorfismo:

Polimorfismo permite que métodos com o mesmo nome se comportem de formas diferentes dependendo do objeto.

```
Encapsulamento.Java

public class Banco {
    public void processarPagamento(Funcionario f) {
        System.out.println("Pagamento realizado para funcionário padrão.");
    }

    public void processarPagamento(Programador p) {
        System.out.println("Pagamento realizado para programador especialista.");
    }
}
```

Contexto real: ao processar pagamentos, o comportamento do método muda conforme o tipo de funcionário, sem precisar novos nomes de função.



04

Abstração

Simplifique o caos.

Abstração:

Abstrair é focar apenas no essencial.
Em vez de lidar com cada detalhe interno, você cria classes e métodos genéricos que representam a ideia principal.

```
Encapsulamento.Java

public abstract class Conexao {
    public abstract void conectar();
}

public class ConexaoMySQL extends Conexao {
    public void conectar() {
        System.out.println("Conectado ao banco MySQL.");
    }
}
```

Contexto real: cada banco de dados tem sua forma de conexão, mas a aplicação só precisa saber que existe um método “conectar”



06

Conclusão

Dominando a Matrix do Backend

Conclusão

Com os quatro pilares da POO, você tem o poder de criar sistemas robustos, modulares e fáceis de manter.

Na Matrix do backend, Java é o seu código de libertação e você acabou de receber as chaves para dominá-lo.



AGRADECIMENTOS

OBRIGADO POR LER ATÉ AQUI

Esse Ebook foi gerado por IA, e diagramado por humano.
O passo a passo se encontra no meu Github

•

Esse conteúdo foi gerado com fins didáticos de construção,
não foi realizado uma validação cuidadosa humana no
conteúdo e pode conter erros gerados por uma IA.



[mauricioRodriguesDev/prompts-recipe-to-create-a-ebook: A recipes with IA tools to create a ebook](https://github.com/mauricioRodriguesDev/prompts-recipe-to-create-a-ebook)