

Explorando os Pilares da Orientação a Objetos com Java

 **Venilton FalvoJr**

Tech Lead na DIO



DIGITAL
INNOVATION
ONE

Mais sobre mim

- Formação: Graduação, Mestrado, Doutorado...
- O Início: Estágio em 2008
- *“Alguém está sentado na sombra hoje porque alguém plantou uma árvore há muito tempo.” Warren Buffett*
- LinkedIn: <https://linkedin.com/in/falvojr>
- GitHub: <https://github.com/falvojr>

Percurso

Passo 1

Introdução

Passo 2

Pilares da Programação Orientada a Objetos

Passo 3

Desafio ;)

Domínio/Problema

Considerando nosso conhecimento no domínio bancário, iremos **abstrair** uma solução Orientada a Objetos em Java. Para isso, vamos interpretar o seguinte cenário:

“Um banco oferece aos seus clientes dois tipos de contas (corrente e poupança), as quais possuem as funcionalidades de depósito, saque e transferência (entre contas da própria instituição).”

Requisitos

- ✓ Noções básicas de Java e Orientação a Objetos (OO)
 - ✓ Noções básicas de Git/GitHub
 - ✓ Engajamento e vontade de aprender ;)
-

Percurso

Passo 1

~~Introdução~~

Passo 2

Pilares da Programação Orientada a Objetos

Passo 3

Desafio ;)

Abstração

Habilidade de concentrar-se nos aspectos essenciais de um domínio, ignorando características menos importantes ou acidentais.

Nesse contexto, objetos são abstrações de entidades existentes no domínio em questão.

Encapsulamento

Encapsular significa **esconder a implementação dos objetos, criando assim interfaces de uso mais concisas e fáceis de usar/entender**. O encapsulamento favorece principalmente dois aspectos de um sistema: a **manutenção e a evolução**.

Herança

Permite que você defina uma classe filha que reutiliza (herda), estende ou modifica o comportamento de uma classe pai. A classe cujos membros são herdados é chamada de classe base. A classe que herda os membros da classe base é chamada de classe derivada.

Polimorfismo

Capacidade de um objeto poder ser referenciado de várias formas, ou seja, é capacidade de tratar objetos criados a partir das classes específicas como objetos de uma classe genérica.

Cuidado, polimorfismo não quer dizer que o objeto fica se transformando, muito pelo contrário, um objeto nasce de um tipo e morre daquele tipo, o que pode mudar é a maneira como nos referimos a ele.

Percurso

Passo 1

~~Introdução~~

Passo 2

~~Pilares da Programação Orientada a Objetos~~

Passo 3

Desafio ;)

Desafio de Projeto

Agora é a sua hora de brilhar, use todo seu conhecimento em Java e Orientação a Objetos para evoluir o projeto criado nesta aula. Para isso, seguem algumas sugestões:

- Use todo o seu conhecimento para estender o domínio, incluindo novas classes, atributos e/ou métodos;
- Incluir o Lombok ao projeto para reduzir sua verbosidade de código.

Dúvidas?

- > Fórum
- > [Comunidade online \(discord\)](#)