**Curso 3 – Abstraindo um Bootcamp usando Orientação a Objetos em Java**

**Desafio: Aprenda na Prática Programação Orientada a Objetos**

Sejam bem-vindos ao desafio: APRENDENDO NA PRÁTICA O PARADIGMA DE ORIENTAÇÃO A OBJETOS.

Desafio este que tive a honra de co-criar com a plataforma de cursos online DIGITAL INNOVATION ONE 💛🧡 e disponibilizado de forma gratuita para a comunidade dos desenvolvedores Java.

💎 O objetivo principal é colocar em prática umas das principais ferramentas da OO: ABSTRAÇÃO, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO, através de um projeto Java.

**🛑 Pré-Requisitos**

✅ Conhecer a sintaxe da Java

✅ Java JDK 11

✅ IDE para desenvolvimento Java (usarei IntelliJ)

✅ Git

✅ Conta no GitHub

👣 Passo-a-Passo

1. Vamos ABSTRAIR o DOMÍNIO Bootcamp e MODELAR seus ATRIBUTOS E MÉTODOS

2. Criaremos as CLASSES: Bootcamp, Cursos, Mentorias e Devs e vamos relaciona-las

3. As CLASSES Curso, Mentoria e Devs também serão MODELADOS, ou seja, criaremos seus ATRIBUTOS E MÉTODOS

4. Para que o código fique mais legível e de fácil manutenção, iremos utilizar de algumas das ferramentas que o PARADIGMA DE ORIENTAÇÃO A OBJETOS (POO) nos oferece: ABSTRAÇÃO, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO

5. E para representar CLASSES que foram criadas e relacionadas, iremos transforma-las em OBJETOS.

📚 Paradigma de Programação Orientado a Objetos (POO)

A visão de Orientação a Objetos (OO) é aquela de um mundo de objetos que interagem.

Este paradigma é um modelo de análise, projeto e programação baseado na aproximação entre o mundo real e o mundo virtual, através da criação e interação entre classes, atributos, métodos, objetos, entre outros.

São 4 os pilares principais do POO: ABSTRAÇÃO, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.

🔺 ABSTRAÇÃO:

🔺 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. 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. 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. A capacidade de tratar objetos criados a partir das classes específicas como objetos de uma classe genérica é chamada de polimorfismo.

‼️ CONCEITOS FUNDAMENTAIS POO ‼️

🔻 DOMÍNIO:

Domínio da aplicação, também conhecida como camada de negócio ou de objetos de negócio, é aquela onde estão localizadas as classes que fazem parte do domínio do problema, ou seja, classes correspondentes a objetos que fazem parte da descrição do problema.

🔻 CLASSE:

Um elemento do código que tem a função de representar objetos do mundo real. Dentro dela é comum declararmos atributos e métodos, que representam, respectivamente, as características e comportamentos desse objeto.

🔻 ATRIBUTO:

Atributos são, basicamente, a estrutura de dados que vai representar a classe. Os atributos também são conhecidos como VARIÁVEL DE CLASSE, e podem ser divididos em dois tipos básicos: atributos de instância e de classe.

🔻 VARIÁVEL:

Uma “região de memória (do computador) previamente identificada cuja finalidade é armazenar os dados ou informações de um programa por um determinado espaço de tempo”.

🔻 MÉTODO:

Os métodos representam os estados e ações dos objetos e classes.

🔻 OBJETO:

Em POO, objeto é um "molde" de uma determinada classe, que passa a existir a partir de uma instância da classe. A classe define o comportamento do objeto, usando atributos (propriedades) e métodos (ações). Objeto em ciência da computação, é uma referência a um local da memória que possui um valor. Um objeto pode ser uma variável, função, ou estrutura de dados.

🔻 INSTÂNCIA:

Uma instância de uma classe é um novo objeto criado dessa classe, com o operador new. Instanciar uma classe é criar um novo objeto do mesmo tipo dessa classe. Uma classe somente poderá ser utilizada após ser instanciada.

🧮 Linguagem de Programação vs Paradigma de Linguagem de Programação

✨ LINGUAGEM DE PROGRAMAÇÃO:

É uma linguagem formal que, através de uma série de instruções, permite que um programador escreva um conjunto de ordens, ações consecutivas, dados e algoritmos para criar programas que controlam o comportamento físico e lógico de uma máquina.

Seguem alguns exemplos de como as linguagens de programação podem ser classificadas:

🔺 Nível de abstração:

Baixo Nível: Assembly

Médio Nível: C, C++, D, Objective C, etc.

Alto Nível: Java, C#, PHP, Javascript, etc.

Altíssimo Nível: Python, Ruby, Elixir, etc.

🔺 Paradigma de programação:

Programação Estruturada: C, Pascal, Ada, etc.

Programação Orientada a Objetos: Java, C#, C++, Objective C, D, etc.

Programação Funcional: Lisp, Scheme, Erlang, Elixir, etc.

🔺 Linguagens classificadas pela arquitetura da aplicação:

Desktop: C, C++, Object Pascal, Java, etc.

Web: PHP, Ruby, Javascript, Java, etc.

🔺 Tipo de execução:

Linguagens compiladas: C, C++, Pascal, D, GO, etc.

Linguagens Interpretadas: Python, Ruby, PHP, Javascript, etc.

Linguagens Hibridas: Java, Erlang, Elixir, etc.

✨ PARADIGMA DE LINGUAGEM DE PROGRAMAÇÃO

É um conjunto de características que podem ser utilizados para categorizar determinado grupo de linguagens. Um paradigma pode oferecer técnicas apropriadas para uma aplicação específica.

PARADIGMAS PRINCIPAIS e SEUS SUBPARADIGMAS

🔸 1. Paradigma Imperativo

Neste paradigma, o programa descreve o processamento necessário para solucionar o problema. Assim, o paradigma imperativo é caracterizado por execução sequencial de instruções, pelo uso de variáveis que representam posições de memória e pelo uso de instruções de atribuição que alteram os valores dessas variáveis.

Vejamos alguns Subparadigmas do Paradigma Imperativo e exemplos linguagens de programação que adotam esses subparadigmas.

🔸 1.1 Paradigma estruturado: ALGOL 58 e ALGOL 60

🔸 1.2 Paradigma concorrente: Java e Ada

🔸 1.3 Paradigma Orientado a Objetos: Smalltalk e Java

🔹 2. Paradigma Declarativo

Este paradigma é o modelo no qual os resultados são descritos, mas os passos para chegar aos resultados não são estabelecidos.

Vejamos alguns Subparadigmas do Paradigma Declarativo e exemplos linguagens de programação que adotam esses subparadigmas:

🔹 2.1 Paradigma Funcional: Lisp e Haskell

🔹 2.2 Paradigma Lógico: Prolog

Links úteis:

<https://academiapme-my.sharepoint.com/:p:/g/personal/camila_cavalcante_dio_me/EaXyYOjBaFpZjkxhexMo5EcBKMEEAI5t5aHlsTjnBQJlUw?e=nxdB6C>

<https://github.com/cami-la/desafio-poo-dio/tree/master/out/production/desafio-poo-dio>