

alpha 
<ed/tech>

Paradigma OO

<Módulo 05

/Aula 07>

Paradigma OO

Orientação a Objeto

A **orientação a objetos é um paradigma de programação** que organiza o código em torno de **"objetos", que representam instâncias de classes**.

Oposto ao paradigma funcional utilizando anteriormente. Com linguagens não orientadas a objeto com C.



Paradigmas principais da OO

1. Abstração:

- **Definição:** Abstração é o processo de **simplificar complexidades**, focando nos aspectos essenciais (relevantes para o projeto) de um objeto.
- **Exemplo:** Ao modelar um carro em um programa, você se concentra nos atributos e comportamentos importantes (cor, velocidade, aceleração) sem se preocupar com detalhes internos complexos.

2. Encapsulamento:

- **Definição:** Encapsulamento é o **empacotamento de dados (atributos) e métodos** que operam nesses dados em uma única unidade chamada classe.
- **Exemplo:** Uma classe "Conta Bancária" pode encapsular dados como saldo e métodos como "sacar" e "depositar". Outro exemplo, para exibir o saldo de uma conta bancária você pode ter um atributo que o saldo é atualizado a cada movimentação nessa conta. Mas ainda no exemplo do saldo, para não existir nenhuma "redundância" e evitar qualquer "inconsistência", é possível decidir recalculando o saldo no momento que precisar, varrendo todas as movimentações existentes nessa conta bancária, desde o depósito inicial, todos as retiradas dessa conta e todos os depósitos dessa mesma conta bancária. O desenvolvedor que vai utilizar a classe "Conta Bancária" não precisa, nem quer saber como foi feita a implementação do cálculo do saldo, mas precisa que o valor seja correto e parece que seria muito dispendioso computacionalmente recalculando o saldo toda vez.

3. Herança:

- **Definição:** Herança permite que **uma classe herde atributos e métodos de outra classe**, estabelecendo uma relação "é um". Esta técnica da Orientação a Objetos propicia a reusabilidade do código. O código ou os métodos da Classe Pai não precisam ser repetidos na Classe Filha. Mas a OO permite que o desenvolvedor inclua novos atributos para a Classe Filha ou novos métodos (comportamentos) nessa Classe Filha.
- **Exemplo:** Uma classe "Carro Elétrico" pode herdar da classe "Carro", aproveitando os atributos e métodos existentes. Implementando.

4. Polimorfismo:

- **Definição:** Polimorfismo permite que um objeto seja **tratado** como se fosse de um tipo diferente, **proporcionando flexibilidade e reutilização de código**.
- **Exemplo:** Diferentes classes podem ter métodos com o mesmo nome, mas comportamentos específicos para cada classe. Isso permite chamar o mesmo método em diferentes objetos, resultando em comportamentos específicos para cada tipo.

Esses paradigmas proporcionam uma maneira eficaz e modular de organizar o código, promovendo a reutilização, manutenção e compreensão do software. A orientação a objetos é amplamente utilizada em muitas linguagens de programação, como Python, Java e C++.