

Polimorfismo

Polimorfismo em POO significa que **o mesmo método**, quando usado em **diferentes classes** e **referindo diferentes objetos**, pode ter comportamentos **diferentes**. Isso acontece porque o comportamento do método é definido pela **classe específica** do objeto em questão.


1. Duas formas principais de polimorfismo em POO:

- **Polimorfismo de substituição** (*Overriding*): Quando um método de uma superclasse é sobrescrito numa subclasse, permitindo comportamentos diferentes.
- **Polimorfismo de sobrecarga** (*Overloading*): Quando um método tem o **mesmo nome**, mas diferentes assinaturas (número ou tipo de parâmetros) dentro de uma **mesma classe**.

O

`override` do método `equals` é um exemplo de polimorfismo. Quando alguém sobrescreve o método `equals` de uma classe (como `Object`), está a alterar o seu comportamento para comparar objetos de uma forma específica para a sua classe. Isso é um exemplo clássico de polimorfismo, onde a implementação do método depende do tipo do objeto no momento da execução, e o comportamento pode variar.

Resumo: Polimorfismo POO vs. Polimorfismo GRASP

 Característica	Polimorfismo em POO	Polimorfismo no GRASP
O que é?	Capacidade de um método ter diferentes implementações em subclasses ou interfaces.	Princípio de design para evitar dependências diretas entre classes.
Foco	Execução de métodos de forma polimórfica.	Distribuição de responsabilidades no código.
Usa herança?	Sim (herança e interfaces).	Preferencialmente interfaces e composição.
Objetivo	Permitir que objetos de diferentes classes sejam tratados de forma uniforme.	Criar um código desacoplado e extensível.
Exemplo de uso	Sobrescrita de métodos em subclasses (<code>extends</code>).	Uso de abstrações (<code>interface</code> ou <code>abstract class</code>).