

1 - Explique como a sobrescrita de métodos permite comportamentos diferentes para objetos da mesma hierarquia.

A sobrescrita de métodos permite que subclasses forneçam sua própria implementação de um método que foi definido na classe base. Isso permite comportamentos diferentes para objetos da mesma hierarquia, mesmo quando eles são tratados como objetos da classe base. Dessa forma, o comportamento do método pode ser ajustado para cada tipo específico de objeto, promovendo o polimorfismo em tempo de execução.

2 - Descreva o mecanismo de sobrecarga de métodos e como ele difere da sobrescrita em termos de polimorfismo.

A sobrecarga de métodos permite criar várias versões de um método com o mesmo nome, mas com diferentes assinaturas (quantidade ou tipo de parâmetros). Ela é resolvida em tempo de compilação e permite flexibilidade no uso de um método com diferentes tipos de dados. A sobrescrita, por outro lado, ocorre quando uma subclasse redefine o comportamento de um método da classe base, sendo resolvida em tempo de execução e é um exemplo de polimorfismo dinâmico.

3 - Discuta como a sobrecarga de construtores facilita a criação de objetos com diferentes conjuntos de informações, permitindo flexibilidade na instanciação de objetos.

A sobrecarga de construtores permite flexibilidade na criação de objetos, pois oferece diferentes formas de instanciar uma classe, de acordo com os dados disponíveis no momento. Isso facilita a criação de objetos com diferentes níveis de detalhe sem precisar fornecer todos os parâmetros obrigatoriamente.

4 - Como o polimorfismo por sobrescrita facilita a implementação de diferentes formas de pagamento dentro de um sistema sem alterar o código da classe base?

O polimorfismo por sobrescrita permite que subclasses personalizem o comportamento de um método sem modificar a classe base. Isso é útil em sistemas como o de pagamentos, onde diferentes formas de pagamento podem ser implementadas e expandidas sem necessidade de alterar o código da classe base. Cada subclasse pode sobrescrever o método conforme suas necessidades específicas, promovendo flexibilidade e manutenção simplificada.

5 - Explique como a sobrecarga de métodos permite implementar várias versões do método converter(), mesmo com diferentes tipos de dados e funcionalidades.

A sobrecarga de métodos permite criar várias versões de um mesmo método com assinaturas diferentes, de modo que o compilador possa escolher qual versão chamar com base nos tipos de dados fornecidos. Isso permite que um único método lide com diferentes tipos de entradas e forneça funcionalidades específicas para cada tipo de dado, aumentando a flexibilidade e a legibilidade do código.