

O polimorfismo de sobrescrita pode ser aplicado tanto diante da herança de classes concretas quanto de classes abstratas ou interfaces. Portanto, já que é possível sobrescrever comportamentos em todos os casos, quando é necessário tornar um método concreto em um método abstrato ou criar uma interface?

Apesar de ser possível sobrescrever comportamentos, há casos em que isso está em desacordo com a programação orientada a objetos por dificultar o entendimento e adaptação, atrapalhando o desenvolvimento, reuso e manutenção. Assim, se houver classes com um método obrigatório mas com comportamentos divergentes, por exemplo, não se deve utilizar um método concreto. Se há uma relação de hierarquia e todas as classes em questão estão relacionadas, o correto é usar uma classe abstrata.

A classe abstrata garante a obrigatoriedade do método abstrato sem definir sua implementação, forçando cada subclasse a fazê-lo. Por este meio, ela é capaz de preservar o comportamento da classe e possibilitar a abstração de detalhes não relevantes (como se dará a implementação do método). As classes necessitam ter uma relação de hierarquia, uma vez que os métodos concretos serão implementados nas classes que forem herdadas da abstrata.

Já quando as classes não têm relação, deve-se usar interface. A interface pode ser aplicada por várias, diferentes classes e uma classe pode implementar diversas interfaces. Esse é o curso de ação para quando se tem um comportamento específico, apesar de não ter uma relação de herança entre as classes. Isso possibilita a implementação de um comportamento definido de diferentes maneiras sem ser necessário alterar o código da base. Contudo, a interface só pode ter métodos abstratos e constantes, diferentemente da classe que tem variáveis e construtores.