|  |  |  |  |
| --- | --- | --- | --- |
| Reuso | Generalização | Delegação | Realização |
| Vantagens | * Fácil de implementar, utilizando conceito de herança simples. * Superclasse consegue esconder seus dados, não permitindo que os mesmos sejam alterados pelas classes derivadas. | * Permite que um objeto reutilize o comportamento de outro sem necessariamente ter que ser subtipo deste. * Pode ser realizado em tempo de execução. | * Melhor coesão e menor acoplamento da classe. * Captura semelhanças entre classes sem forçar relação entre elas. |
| Desvantagens | * Expõe detalhes da superclasse as subclasses, violando princípio do encapsulamento. * Possível violação do Princípio de Liskov, ou Regra da Substituição. | * Diminui desempenho, pois implica cruzar a fronteira de outro objeto para realização de uma função. * Não pode ser utilizado em situação em que uma classe parcialmente abstrata está envolvida. | * Não reutiliza código. * Interfaces devem ser utilizadas várias vezes, caso contrário não existe motivo delas serem utilizadas. |