

# Abstract Factory

Fornece uma interface para criar famílias de objetos relacionados ou dependentes sem especificar suas classes concretas.

Frequência de uso: 

1	2	3	4	5

 Alto



# Factory Method

Defina uma interface para criar um objeto, mas deixe as subclasses decidirem qual classe instanciar. O Factory Method permite que uma classe adie a instanciação para as subclasses.

Frequência de uso: 

1	2	3	4	5

 Alto



# Singleton

Certifique-se de que uma classe tenha apenas uma instância e forneça um ponto global de acesso a ela.

Frequência de uso: 

1	2	3	4	5

 Médio Alto



# Adapter

Converte a interface de uma classe em outra interface esperada pelos clientes. O adaptador permite que as classes trabalhem juntas de outra forma, devido a interfaces incompatíveis.

Frequência de uso: 

1	2	3	4	5
■	■	■	■	□

 Médio Alto



# Facade

Fornece uma interface unificada para um conjunto de interfaces em um subsistema. Facade define uma interface de nível superior que torna o subsistema mais fácil de usar.

Frequência de uso: 

1	2	3	4	5

 Alto



# Composite

Componha objetos em estruturas de árvore para representar hierarquias parte-todo. O Composite permite que os clientes tratem objetos individuais e composições de objetos uniformemente.

Frequência de uso: 

1	2	3	4	5
■	■	■	■	□

 Médio Alto



# Command

Encapsula uma solicitação como um objeto, permitindo, assim, parametrizar clientes com diferentes solicitações, solicitações de fila ou log e oferecer suporte a operações que podem ser desfeitas.

Frequência de uso: 

1	2	3	4	5
■	■	■	■	□

 Médio Alto



# Strategy

Defina uma família de algoritmos, encapsule cada um e torne-os intercambiáveis. O strategy permite que o algoritmo varie independentemente dos clientes que o utilizam.

Frequência de uso: 

1	2	3	4	5

 Médio Alto



# Observer

Defina uma dependência um-para-muitos entre objetos para que, quando um objeto mudar de estado, todos os seus dependentes sejam notificados e atualizados automaticamente.

Frequência de uso: 

1	2	3	4	5

 Alto