

Um padrão de design é uma solução repetível geral para um problema que ocorre comumente em design de software. Um padrão de design não é um projeto acabado que pode ser transformado diretamente em código. É uma descrição ou modelo de como resolver um problema que pode ser utilizada em muitas situações diferentes.

Usos de Design Patterns

Os padrões de design pode acelerar o processo de desenvolvimento, fornecendo testados, paradigmas de desenvolvimento comprovados. Design de software eficaz requer considerar as questões que podem não se tornam visíveis até mais tarde na implementação. Reutilizar padrões de design ajuda a evitar problemas sutis que podem causar grandes problemas e melhora a legibilidade do código para programadores e arquitetos familiarizadas com os padrões.

Muitas vezes, as pessoas só entender como aplicar certas técnicas de design de software para determinados problemas. Estas técnicas são de difícil aplicação a uma ampla gama de problemas. Os padrões de design fornecer soluções gerais, documentados em um formato que não requer especificidades ligadas a um problema particular.

Além disso, os padrões permitem que os desenvolvedores para se comunicar usando bem conhecidos, nomes bem compreendidas para interações de software. Padrões de design comum pode ser melhorada ao longo do tempo, tornando-os mais robustos do que os projetos ad-hoc

Padrões de projeto criacionais

Estes padrões de design são todos sobre a instanciação de classe. Esse padrão pode ainda ser dividido em padrões de classe de criação e padrões de objeto-criacional. Embora os padrões de classe de criação de usar a herança de forma eficaz no processo de instanciação, os padrões de criação de objeto usar a delegação eficaz para começar o trabalho feito.

Abstract Factory

Cria uma instância de várias famílias de aulas

Builder

Separa construção do objeto de sua representação

Factory Method

Cria uma instância de várias classes derivadas

Objeto pool

Evite aquisição cara e liberação de recursos através da reciclagem de objetos que não estão mais em uso

Protótipo

A instância totalmente inicializado a ser copiado ou clonado

Singleton

Uma classe da qual apenas uma única instância pode existir

Padrões de projeto estruturais

Estes padrões de design são todos sobre a classe e composição de objetos. Padrões de classe de criação estruturais usar a herança para compor interfaces. -objetos padrões estruturais definir formas de compor objetos para obter novas funcionalidades.

Adaptador de

interfaces de fósforo de diferentes classes

Ponte

Separa a interface de um objeto a partir de sua implementação

Composite

A estrutura da árvore de objetos simples e compostas

Decorator

Adicionar responsabilidades aos objetos dinamicamente

Fachada

A única classe que representa um subsistema inteiro

Flyweight

Um exemplo de grão fino usado para compartilhamento eficiente

Dados classe privada

Restringe o acesso acessor / modificador

Proxy

Um objeto que representa um outro objeto

Padrões de projeto comportamentais

Estes padrões de design são todos sobre a comunicação objetos de classe. Padrões de comportamento são os padrões que são mais especificamente preocupados com a comunicação entre objetos.

Cadeia de responsabilidade

Uma maneira de passar um pedido entre uma cadeia de objetos

Comando

encapsular um pedido de comando como um objeto

Intérprete

Uma maneira de incluir elementos de linguagem em um programa

Iterator

Sequencialmente acessar os elementos de uma coleção

Mediador

define a comunicação simplificada entre as classes

[Memento](#)

Capturar e restaurar o estado interno de um objeto

[Objeto nulo](#)

concebido para funcionar como um valor padrão de um objeto

[Observador](#)

Uma maneira de notificar a mudança para um número de classes

[Estado](#)

Alter comportamento de um objeto quando suas alterações de estado

[Estratégia](#)

Encapsula um algoritmo dentro de uma classe

[Template Method](#)

Adiar as etapas exatas de um algoritmo para uma subclasse

[Visitor](#)

Define uma nova operação a uma classe sem alterações