



CMP1550

Prof. Fabricio

Padrões GoF

O que é um padrão de projeto?

- Padrão de Projeto descreve uma solução comprovada para um problema recorrente e conhecido no desenvolvimento de software orientado a objetos.

Porque utilizar (Padrões)?

- A satisfação de se criar algo elegante, porém simples e de uma solução bastante reutilizável.
- Objetivo dos padrões é codificar conhecimento existente de uma forma que possa ser reaplicado em contextos diferentes.

Quando Documentar um padrão

- Quando programamos encontramos muitos problemas que ocorrem, ocorreram e **vão** ocorrer novamente.
- Documentar um padrão é uma maneira de você poder reusar e possivelmente compartilhar informação que você aprendeu sobre a melhor maneira de se resolver um problema de desenho de software.

Origem

- Design Patterns vem do trabalho de um arquiteto chamado Christopher Alexander, no final da década de 70.
- 2 livros:
 - A Pattern Language
 - A Timeless Way of Building

=> ele exemplificava o uso e descrevia seu raciocínio para documentar os padrões

Características – padrões

- Encapsulamento
- Generalidade
- Equilíbrio
- Abstração
- Abertura
- Combinatoriedade.



GoF - Gang of Four

A Gangue dos Quatro

- 1995, um grupo de quatro profissionais escreveu e lançou o livro "Design Patterns", um catálogo com 23 padrões de desenho.
 - Erich Gamma, Richard Helm, Ralph Johnson, e John Vlissides

“ Conhecer esses padrões é fundamental para entender os modernos frameworks e desenvolver software de qualidade que busca os melhores resultados da orientação a objetos ”

GoF padrão

- Repertório de soluções e princípios que ajudam os desenvolvedores a criar software, e que são codificados em um formato estruturado, consistindo de:
- Nome
- Problema que soluciona
- Solução do problema

GoF

- Nome: descreve a essência do padrão.
- Problema: descreve quando aplicar o padrão, e em que condições.
- Solução: descrição abstrata de como usar as classes e objetos para solucionar o problema

Podendo ainda conter:

- Exemplos: uma ou mais figuras, diagramas ou descrições que ilustrem um protótipo de aplicação.
- Consequência: custo, benefícios e impacto ao se aplicar o padrão no sistema.

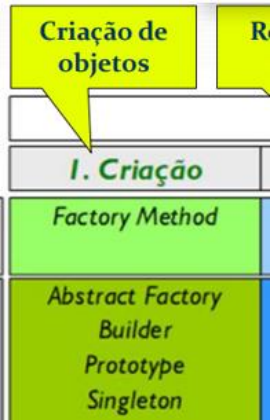
Catálogo de padrões GoF

3 abordagens

- Padrão de Criação
 - Relacionados à criação de classes e objetos. [?] Ligados ao processo de instanciação.
- Padrão Estrutural
 - Tratam da alteração da estrutura de um programa, e das associações entre classes e objetos.
- Padrão Comportamental
 - Observam a maneira com que classes e objetos podem interagir.

Classificação dos 23 padrões (com base no GoF)

		Criação de objetos	Relacionamento entre objetos	Comunicação entre objetos
		Propósito		
		1. Criação	2. Estrutura	3. Comportamento
Escopo	Classe	Factory Method	Class Adapter	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Object Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor



Padrões de Criação

- Factory
 - Permite a criação de famílias de objetos relacionados ou dependentes, através de uma única interface e sem que a classe concreta seja especificada.
 - Permite que uma classe delegue a responsabilidade de instanciamento às subclasses.
- Abstract Factory
 - O padrão é uma fábrica de objetos que retorna uma das várias fábricas.
 - Define e mantém relacionamentos entre objetos.
- Builder
 - Permite a separação da construção de um objeto complexo da sua representação.
- Prototype
 - Permite a criação de objetos a partir de um modelo original, ou protótipo.
- Singleton:
 - Usado para criar uma única instância de uma classe.

Classificação dos 23 padrões (com base no GoF)

		Criação de objetos	Relacionamento entre objetos	Comunicação entre objetos
		Propósito		
		1. Criação	2. Estrutura	3. Comportamento
Escopo	Classe	Factory Method	Class Adapter	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Object Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

Padrões Estruturais

- ADAPTER: Permite que classes com interfaces incompatíveis possam interagir.
- BRIDGE: Desacopla a interface da implementação. Ocultando detalhes dos clientes/aplicações
- COMPOSITE: Objeto que é constituído pela composição de objetos similares a ele.
- DECORATOR: Atribui responsabilidade adicionais a um objeto dinamicamente .
- FAÇADE: Interface unificada para um subsistema. Torna o subsistema mais fácil de usar.
- FLYWEIGHT: Vários objetos devem ser manipulados, mas não suportam dados adicionais.
- PROXY: Fornece objeto representante de outro objeto para controlar o acesso ao mesmo.

Padrões Comportamentais – 1/2

- Observer
 - Define uma relação de dependência 1:N de forma que quando um certo objeto (assunto) tem seu estado modificado os demais (observadores) são notificados.
- State
 - Permite objeto alterar seu comportamento quando estado interno muda.
- Strategy
 - Permite que uma família de algoritmos seja utilizada de modo independente e seletivo.
- Template
 - Define o esqueleto de um algoritmo em uma operação adiando a definição de alguns passos para a subclasse.
- Visitor
 - Define operações independentes a serem realizadas sobre elementos de uma estrutura

Classificação dos 23 padrões (com base no GoF)

		Criação de objetos	Relacionamento entre objetos	Comunicação entre objetos
		Propósito		
		1. Criação	2. Estrutura	3. Comportamento
Escopo	Classe	Factory Method	Class Adapter	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Object Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

Padrões Comportamentais – 2/2

- Chain of Responsibility
 - Evita dependência do remetente(cliente) de uma requisição ao seu destinatário.
- Command
 - Associa uma ação a diferentes objetos através de uma interface conhecida.
- Interpreter
 - Usado para ajudar uma aplicação a entender uma declaração de linguagem natural e executar a funcionalidade da declaração.
- Iterator
 - Provê forma de percorrermos elementos de uma coleção sem violar o seu encapsulamento.
- Mediator
 - É quem desacopla e gerencia as colaborações entre um grupo de objetos.
- Memento
 - Captura e externaliza o estado interno de um objeto

Padrões GoF: Formas de classificação

segundo Metsker

<i>Intenção</i>	<i>Padrões</i>
<i>1. Interfaces</i>	<i>Adapter, Facade, Composite, Bridge</i>
<i>2. Responsabilidade</i>	<i>Singleton, Observer, Mediator, Proxy, Chain of Responsibility, Flyweight</i>
<i>3. Construção</i>	<i>Builder, Factory Method, Abstract Factory, Prototype, Memento</i>
<i>4. Operações</i>	<i>Template Method, State, Strategy, Command, Interpreter</i>
<i>5. Extensões</i>	<i>Decorator, Iterator, Visitor</i>

Grupos de Pesquisa

- Apresente uma forma de desenvolver projeto de software que pode ser útil a uma equipe.