

Padrões de Projeto

Aula 23 – Introdução – Padrão de Projetos

Prof. Nemésio Freitas Duarte Filho

Introdução

- ▶ Resumidamente, padrões são soluções de eficiência já comprovada e amplamente utilizadas para a resolução de problemas comuns em projeto de software.
- ▶ Padrão descreve um problema no nosso ambiente e o cerne da sua solução, de forma que essa solução possa ser usada varias vezes (Christopher Alexander)



O que é um padrão de projeto?

- ▶ Um padrão de projeto é uma estrutura recorrente no projeto de software orientado a objetos. Pelo fato de ser recorrente, vale a pena que seja documentada e estudada.



Padrões e reuso

- ▶ A aplicação de padrões é um exemplo de reuso de projeto, de idéias e soluções.
- ▶ Muitos dos padrões visam facilitar o reuso de módulos do software. O esforço extra gasto na fase de projeto é compensado pelos ganhos em flexibilidade e reuso (Gamma).



Por que usar padrões

- ▶ Aprender com a experiência de outros:
 - ▶ Ajudam a resolver os principais problemas dos iniciantes;
 - ▶ Permitem que façamos bons projetos rapidamente.
- ▶ Software de melhor qualidade:
 - ▶ Melhores práticas em OO;
 - ▶ Soluções bem testadas e documentadas



Componentes de um padrão

- Os principais atributos de uma boa descrição de um padrão de projeto são:
 - **Nome:** referência que descreve de forma bastante sucinta o padrão.
 - **Problema** (motivação, intenção e objetivos, aplicabilidade): apresenta o contexto do padrão e quando ele pode ser usado.
 - **Solução** (estrutura, participantes, exemplo de código): descreve a solução e os elementos que a compõem.
 - **Conseqüências e padrões relacionados:** analisa os resultados, vantagens e desvantagens obtidos com a aplicação do padrão.



Sistemas de padrões

- Requisitos de um bom sistema de padrões:
 - O sistema deve conter uma boa quantidade de padrões.
 - A descrição dos padrões deve seguir um formato padronizado.
 - O sistema deve ser estruturado, organizando os padrões seguindo critérios bem definidos.
 - O sistema deve mostrar o relacionamento entre os padrões.
 - O sistema deve estar estruturado de modo a poder evoluir.



Padrões reconhecidos

- ▶ GoF é a referência clássica;
- ▶ Novos problemas surgiram e incentivaram a descoberta de novos padrões de projeto;
- ▶ Muitos desses novos padrões são baseados nos padrões já descritos no catálogo do GoF.
- ▶ É importante conhecê-los porque se referem a domínios mais específicos:
 - ▶ Ex.: um desenvolvedor JEE deve conhecer os padrões de projeto JEE (antigo J2EE).



Projetando para mudanças

- Sistemas devem ser projetados para facilitarem a manutenção:
 - Riscos de custos imprevistos;
 - Retrabalho.
- Padrões de projeto podem ajudar:
 - Promover desacoplamento, permitindo maior liberdade dos objetos;
 - Código mais robusto, legível e de maior qualidade.



Classificação dos padrões

- ▶ Hoje existem padrões em grande número e com diferentes características. Por exemplo, temos padrões que se aplicam em diferentes áreas, em diferentes problemas, que possuem diferentes níveis de abstração, etc.
- ▶ Isso tornou necessária a classificação dos padrões, ou seja, a sua organização e divisão em grupos de acordo com as suas características.



Classificação dos padrões

- ▶ Recomendações para a escolha dos critérios de classificação:
 - ▶ Devem ser utilizados poucos critérios de classificação.
 - ▶ Os critérios devem ser baseados nas características principais dos padrões.



Classificação dos padrões

- ▶ Critérios utilizados:
 - ▶ Nível de abstração
 - ▶ Padrões de Arquitetura
 - ▶ Padrões de Projeto
 - ▶ Linguagem
 - ▶ Tipo de problema resolvido
 - ▶ Comunicação entre módulos
 - ▶ Sistemas distribuídos
 - ▶ etc



Classificação dos padrões

- Em geral os padrões de projeto podem ser classificados em três diferentes tipos (critérios utilizados em Gamma):
 - Padrões de criação: abstraem o processo de criação de objetos a partir da instanciação de classes.
 - Padrões estruturais: tratam da forma como classes e objetos estão organizados para a formação de estruturas maiores.
 - Padrões comportamentais: preocupam-se com algoritmos e a atribuição de responsabilidade entre objetos.
- Cada um desses tipos pode ser sub-classificado em (escopo):
 - Padrões de classes: em geral estáticos, definidos em tempo de compilação.
 - Padrões de objetos: em geral dinâmicos, definidos em tempo de execução.



Classificação dos padrões

► Classificação segundo o Gof (Gang of Four)

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

Classificação dos padrões

- Classificação dos Gof`s D.P. segundo Metsker

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



Finalidade dos Padrões

► Padrões de Criação

- Os padrões de criação mostram como tornar esse projeto mais flexível e não necessariamente menor.
- Tornam um sistema independente de como seus objetos são criados, compostos e representados



Finalidade dos Padrões

- ▶ Padrões de Criação:
 - ▶ Abstract Factory;
 - ▶ Builder;
 - ▶ Factory Method
 - ▶ Prototype;
 - ▶ Singleton



Finalidade dos Padrões

► Padrões Estruturais

- Os padrões se preocupam com a forma como classes e objetos são compostos para formar estruturas maiores.
- Os padrões estruturais de **classe** utilizam a herança para compor interfaces ou implementações.
- Os padrões estruturais de **objetos** descrevem maneiras de compor objetos para obter novas funcionalidades.



Finalidade dos Padrões

- ▶ **Padrões Estruturais:**

- ▶ Adapter;
- ▶ Bridge;
- ▶ Composite;
- ▶ Decorator;
- ▶ Façade;
- ▶ Flyweight;
- ▶ Proxy;



Finalidade dos Padrões

- ▶ Padrões Comportamentais
 - ▶ Tratam de algoritmos e como atribuir responsabilidades entre objetos
 - ▶ Associados à maneira que objetos e classes distribuem suas responsabilidades para realizar uma tarefa
 - ▶ Focam na abstração do comportamento



Finalidade dos Padrões

■ Padrões Comportamentais

- Chain of Responsibility
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor



Conclusão

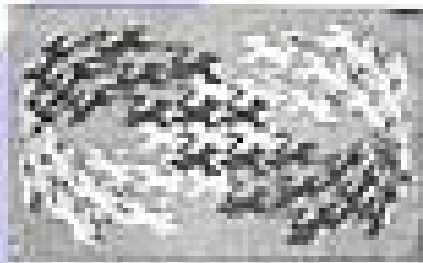
- ▶ Impossibilidade de um desenvolvedor se dedicar ao estudo de todas as categorias de padrões existentes.
- ▶ As atividades de análise e projeto devem ser bem desenvolvidas para permitir a aplicação de padrões.



Livro: Padrões de Projeto

Padrões de Projeto

Soluções reutilizáveis de software
orientadas a objetos



ERICH GAMMA
RICHARD HELM
RALPH JOHNSON
JOHN VLISSIDES

Design Patterns



Capa do livro "Padrões de Projeto...", de Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides. Bookman. 2000.

Conhecido por Padrões GoF = Gang of Four

Dúvidas

E-mail:

nemesiofreitas@gmail.com

Site:

<https://sites.google.com/site/nemesiofreitas>

