

Guia - Padrões de Projeto

SUMÁRIO

1. INTRODUÇÃO	2
2. GOF	2
3. GRASP	3
4. REFERÊNCIAS	3

1. INTRODUÇÃO

Em engenharia de software, um padrão de projeto ou padrão de desenho (do inglês design pattern) é uma solução geral reutilizável para um problema que ocorre com frequência dentro de um determinado contexto no projeto de software. Um padrão de projeto não é um projeto finalizado que pode ser diretamente transformado em código fonte ou de máquina, ele é uma descrição ou modelo (template) de como resolver um problema que pode ser usado em muitas situações diferentes. Padrões são melhores práticas formalizadas que o programador pode usar para resolver problemas comuns quando projetar uma aplicação ou sistema. Padrões de projeto orientados a objeto normalmente mostram relacionamentos e interações entre classes ou objetos, sem especificar as classes ou objetos da aplicação final que estão envolvidas. Padrões que implicam orientação a objetos ou estado mutável mais geral, não são tão aplicáveis em linguagens de programação funcional.

2. GOF

Os padrões "GoF" são organizados em 3 famílias :

- **Padrões de Criação:** relacionados à criação de objetos
- **Padrões Estruturais:** tratam das associações entre classes e objetos.
- **Padrões Comportamentais:** tratam das interações e divisões de responsabilidades entre as classes ou objetos.

Padrões de Criação

- Abstract Factory
- Builder
- Factory Method
- Prototype
- Singleton

Padrões Estruturais

- Adapter
- Bridge
- Composite
- Decorator
- Façade (ou Facade)
- Flyweight
- Proxy

Padrões Comportamentais

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

3. GRASP

Os padrões GRASP, sigla para **General Responsibility Assignment Software Patterns**, consistem de um conjunto de práticas para atribuição de responsabilidades a classes e objetos em projetos orientados a objeto.

Os padrões utilizados pelo GRASP são:

- Controlador (Controller),
- Criador (Creator),
- Indireção (Indirection),
- Especialista na informação (Information expert),
- Alta coesão (High Cohesion),
- Baixo acoplamento (Loose coupling),
- Polimorfismo (Polymorphism),
- Variações protegidas (Protected variations), e
- Invenção pura (Pure fabrication).

Todos esses padrões servem para a resolução de problemas comuns e bastante típicos de desenvolvimento de software orientado a objeto. Portanto, tais técnicas apenas documentam e normatizam as práticas já consolidadas, testadas e conhecidas no mercado.

Os padrões GRASP estão mais como uma ferramenta mental ou uma filosofia de design, mas que ainda assim são úteis para o aprendizado e desenvolvimento de um bom design de software. Note que alguns padrões GoF implementam soluções correspondentes com padrões GRASP.

4. REFERÊNCIAS

Anexo - Padroes GoF-UFPE.pdf

Anexo - Padroes GRASP-UFPE.pdf

CRAIG LARMAN. Utilizando UML e Padrões. Bookman. 2007.