



Padrões de projeto



Padrões de projeto

- Repetição de problemas no desenvolvimento de software
 - Acesso a dados
 - Concorrência
 - Criação de objetos
 - Estruturação em composição
 - Adaptação de objetos
 - **E muitos outros**
- Com experiência, bons desenvolvedores aplicam padrões empiricamente
- Mas porque não reutilizar boas soluções ?

Padrões de projetos podem ser informalmente considerados como aplicação de boas práticas e/ou reuso de repertório de soluções

Padrões de projeto

- Repertório de soluções para resolver problemas recorrentes
 - Descrição detalhada do problema
 - Base de solução para ser aplicada e adaptada
- Reuso de ideias, não código
- Elementos possíveis na descrição de um padrão
 - Nome : reflete o padrão, facilmente lembrado (ex.: **Singleton, Visitor, Composite**)
 - Facilita documentação, troca de informações e aprendizagem
 - Problema : onde será aplicado
 - Solução : como será aplicado
 - Consequências : resultados esperados, impactos
 - Outros
 - Exemplos, justificativa, padrões relacionados, código

Muitos padrões/princípios

- Fonte informal (Wikipedia)
https://en.wikipedia.org/wiki/Software_design_pattern
- Mais de 50 padrões de projeto
 - Criação, Estrutura, Comportamentais e Concorrência
- E muitas soluções/siglas para criação de bons projetos
 - **GoF (Livro padrões de projeto - pioneiro)**
 - GRASP
 - SOLID
 - KISS, DRY, etc.
- **Objetivo central:** REUSO (sempre)

Livro “Design Patterns”

- Design Patterns: Elements of Reusable Object-Oriented Software (1994)

(Padrões de projeto: soluções reutilizáveis de software orientado a objetos)

- *Gang of Four:*
Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides
- 23 padrões
 - Não são invenções dos autores: documentação de soluções obtidas por experiência

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

