

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Prof. Felipe Scheidt – IFPR – Campus Foz do Iguaçu

Padrões de projetos

Apresentação

Inspiração

Cada padrão descreve um problema que ocorre repetidamente de em nosso ambiente, e descreve a parte central da solução para aquele problema de uma forma que seja possível usar esta solução um milhão de vezes, sem nunca implementa-la duas vezes da mesma forma.

Christopher Alexander, 1977

Problema

1. Código evolui: mudanças vão acontecer mais cedo ou mais tarde.
2. Exemplos:
 1. Pagamento
 2. Frete, formas de envio
3. Pesadelo da manutenção de código
4. Ocorre com certa frequência: sai mais "barato" refazer o sistema do que manter.

Estudo de padrões em software

- Identificação de códigos que se repetem (em diversos projetos)
- Boas soluções receberam um **nome**.
- Ao invés de discutir um sistema em termos de estrutura de dados (pilha, fila, árvore e listas ligadas), passamos a falar de padrões (Fábricas, Fachadas, Observador, ...)
- Foco no design da arquitetura, reaproveitamento e qualidade de código.

Gang of Four (GoF)

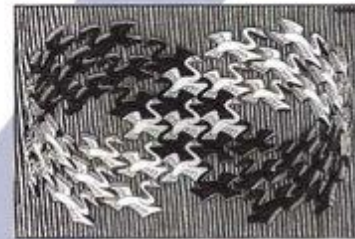
Primeira literatura abrangente e sistemática sobre Design Patterns.

E. Gamma and R. Helm and R. Johnson and J. Vlissides. **Design Patterns - Elements of Reusable Object-Oriented Software**. Addison-Wesley, 1995.

Design Patterns

Elements of Reusable
Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides



© 1995 W. H. Freeman & Co. All rights reserved.

Foreword by Grady Booch



ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

O Formato de um padrão

Todo padrão inclui:

1. Nome
2. Problema (explicação e contexto)
3. Solução
4. Consequências (resultados esperados)

Tipos de Padrões de Projetos

Criacional: relacionados à criação de objetos

Estrutural: tratam das associações entre classes e objetos.

Comportamentais: tratam das interações e divisões de responsabilidades entre as classes ou objetos.

Princípios de projetos

- Diferenciar aspectos que variam daqueles que se mantêm constantes
- Diferenciar interface e implementação

Princípios de projetos

- Identifique os aspectos da aplicação que variam e separe-os daqueles que se mantêm constante.
- Pegue o código que variar e encapsule para que isso não afete o resto do seu código.
- Tome a parte do código que varia e encapsule ela, de modo que mais tarde você pode alterar ou estender a parte que varia sem afetar a parte que não varia.
- Esses conceitos são a base dos padrões de projetos.
- Padrões fornecem uma maneira de deixar que alguma parte do sistema varie independentemente de todas as outras partes.

Princípios de projetos

- Programe uma interface não uma implementação
- Programar para uma interface significa na verdade programar para um supertipo.
- A ideia é explorar o polimorfismo programando um supertipo de modo que o objeto de tempo de execução real não fique preso no código.

Paradigma OO

Padrões de projetos são tipicamente definidos no paradigma orientado a objetos.

Conceitos fundamentais:

1. **Classe abstrata**
2. **Interface**
3. **Polimorfismo**
4. **Herança**