

Modulo I

Princípios e Padrões de Projeto de SW em Java

Professores

Eduardo Bezerra – edubezerra@gmail.com

Ismael H F Santos – ismael@tecgraf.puc-rio.br

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

1

Ementa

- **Princípios e Padrões de Projeto de SW**
 - Princípios de Projeto de SW
 - Padrões de SW

Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

2

Bibliografia

- *Craig Larman, Utilizando UML e Padrões, Ed Bookman*
- *Eric Gamma, et ali, Padrões de Projeto, Ed Bookman*
- *Martin Fowler, Analysis Patterns - Reusable Object Models, Addison-Wesley, 1997*
- *Martin Fowler, Refatoração - Aperfeiçoando o projeto de código existente, Ed Bookman*

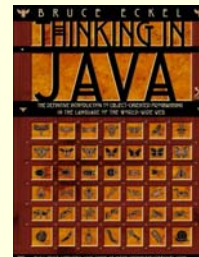
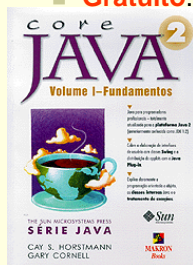
Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

3

Livros

- **Core Java 2**, Cay S. Horstmann, Gary Cornell
 - Volume 1 (Fundamentos)
 - Volume 2 (Características Avançadas)
- **Java: Como Programar**, Deitel & Deitel
- **Thinking in Patterns with JAVA**, Bruce Eckel
 - **Gratuito.** <http://www.mindview.net/Books/TIJ/>



Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

4

POO-Java

Princípios
De
Projeto de SW



Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

5

Reuso de Software

- **Motivações para reutilização de software**
 - Aspecto econômico
 - Produtividade
 - "Time to market"
 - Qualidade
 - Utilização de artefatos (código, decisões de projeto, bibliotecas de funções e classes, componentes etc.) já testados e validados
- **Formas de reutilização**
 - Anos 70: Subrotinas, Módulos
 - Anos 80: Biblioteca de classes, Geradores de aplicações
 - Anos 90: Componentes, Frameworks, **Padrões de software**

Julho 06

Prof(s). Eduardo Bezerra & Ismael H. F. Santos

6

POO-Java

*Padrões
de
Software*



Padrões de software

- Desde o início do desenvolvimento de sistemas de SW, diversos sistemas foram desenvolvidos.
 - Uma parte desses foi bem sucedida, outra parte não...
- À medida que esses projetos foram realizados, os desenvolvedores foram colecionando diversas soluções bem sucedidas para problemas recorrentes no desenvolvimento de software.
- Desta forma, quando o mesmo problema ocorria, a equipe de desenvolvimento já tinha uma solução genérica (re)aplicável a ele.

O que são padrões de software

- **Um padrão (pattern) é uma descrição das características de uma solução comprovada para um problema recorrente, onde os elementos essenciais são considerados e os detalhes irrelevantes são omitidos.** (Gamma et al, 1995)
 - Problema recorrente; descrição da solução; elementos essenciais
 - Permitem que os desenvolvedores **concentrem** seus **esforços** nos aspectos inéditos do problema.
 - Compõem um **vocabulário** de **alto nível** para discussão de questões relativas ao projeto de sistemas de software.

O que são padrões de software

- **Analogia: jogo de xadrez.**
 - Diversos mestres já elaboraram jogadas geniais.
 - Essas jogadas foram catalogadas e passaram a ser utilizadas por outros jogadores.
- **O mesmo acontece no desenvolvimento de software.**
 - Desenvolvedores experientes criaram soluções para diversos problemas relacionados ao desenvolvimento de software.
 - Essas “jogadas” (padrões) também foram catalogadas.
 - Esses padrões são agora utilizados por outros desenvolvedores quando estes se deparam com problemas semelhantes.

Histórico

- Historicamente, o conceito de padrões não foi concebido por profissionais da área de computação.
- Um arquiteto civil, Christopher Alexander escreveu (1970) dois livros sobre padrões de projeto para a arquitetura civil.
 - “The timeless way of building” (1977)
 - “A pattern language” (1979)



Christopher Alexander

Problema vs solução

- É importante notar que um padrão descreve as características essenciais de uma solução para um determinado problema.
 - Na maioria das vezes, a solução deve ser adaptada para a situação específica na qual o padrão está sendo aplicado.

“Cada padrão descreve um problema que ocorre frequentemente no nosso ambiente, e então descreve o núcleo de uma solução para tal problema. Esse núcleo pode ser utilizado um milhão de vezes, sem que haja duas formas de utilização iguais.”

Christopher Alexander (1977)

Níveis de abstração e padrões

- Existem padrões catalogados em diversos níveis de abstração
 - análise, projeto (design), arquitetura.
- **Padrões de análise** (Analysis Patterns)
 - Úteis na modelagem de domínio
 - Apóiam o reuso de idéias durante a fase de análise
 - Podem se aplicar a um único domínio ou a vários domínios
- **Padrões de projeto** (Design Patterns)
 - Úteis no desenho (solução)
- **Padrões arquiteturais** (Architectural Patterns)
 - Úteis na definição dos elementos arquiteturais de um sistemas (subsistemas, camadas, etc.)

Conclusões

- Desenvolver SW é difícil.
- Desenvolver SW reutilizável e ainda mais difícil.
- Padrões facilitam o reuso de projetos e arquiteturas de sucesso.
- OOP = ferramenta. “Ter um martelo não faz de ninguém um arquiteto.”
- Estudar padrões de software é uma necessidade para desenvolvedores de software que almejam qualidade em seu trabalho.

Conclusões

- Em última análise, padrões de software são manifestações e aplicações dos **fundamentos** da orientação a objetos.
 - Foco nas responsabilidades dos objetos, e não em como implementá-los.
 - **Identificação do que é variável** no desenho, e posterior encapsulamento dessa parte.
 - Adição de camadas entre coisas que podem mudar de forma independente uma da outra (**indireção**).