

Módulo I

Princípios e Padrões de Projeto de SW em Java

Professores

Eduardo Bezerra – edubezerra@gmail.com

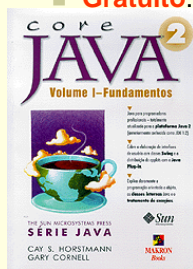
Ismael H F Santos – ismael@tecgraf.puc-rio.br

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*

Livros

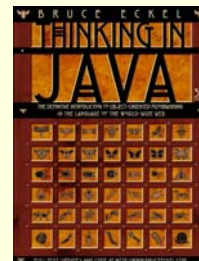
- **Core Java 2**, Cay S. Horstmann, Gary Cornell
 - Volume 1 (Fundamentos)
 - Volume 2 (Características Avançadas)
- **Java: Como Programar**, Deitel & Deitel
- **Thinking in Patterns with JAVA**, Bruce Eckel
 - **Gratuito.** <http://www.mindview.net/Books/TIJ/>



Julho 06



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



3

Ementa

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

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
Projeto de
Software*



Padrões de Projeto 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.

Design Patterns – Padrões de Projeto

- A expressão Design Pattern é usada para definir soluções de projeto que ocorrem com uma certa frequência em projetos de software.
- A idéia aplicada a software é inspirada em conceitos já estabelecidos na área de arquitetura.
- Os conceitos de na área de arquitetura foram estabelecidos por Christopher Alexander, através dos livros *A Pattern Language*(1977) e *A Timeless Way of Building*(1979).

Julho 06

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

11

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.
 - “Notes on the Synthesis of Form”(1964)
 - “The timeless way of building” (1977)
 - “A pattern language” (1979)



Christopher Alexander

Julho 06

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

12

Padrões de Projeto de SW

- Em 1987, a partir dos conceitos criados por Alexander, os programadores Kent Beck e Ward Cunningham propuseram os primeiros padrões de projeto para a área da ciência da computação. Em um trabalho para a conferência OOPSLA, eles apresentaram alguns padrões para a construção de janelas na linguagem Smalltalk.
- O movimento ao redor de padrões de projeto ganhou popularidade com o livro *Design Patterns: Elements of Reusable Object-Oriented Software*, publicado em 1995. Os autores desse livro são Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides, conhecidos como a "Gangue dos Quatro" (Gang of Four) ou simplesmente "GoF"

Julho 06

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

13

Padrões de Projeto de SW

- O trabalho da "gangue dos quatro" consistiu em estudar vários sistemas de software de grande porte, identificar um conjunto de *soluções de projeto* que ocorrem com frequência, classificá-las e catalogá-las num formato que facilite a sua compreensão e principalmente o seu *reuso*.
- Design Patterns tratam de soluções de projeto ao nível estrutural.
- Patterns são descobertos à medida em que se acumula experiência no projeto e desenvolvimento de software.

Julho 06

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

14

Padrões de Projeto de SW

- Em Programação Orientada a Objetos normalmente se enfatiza a **reutilização de código** como sendo uma das vantagens dessa técnica.
- O reuso de um Pattern implica no **reuso de uma solução**, que não significa necessariamente no reuso de código.
- Padrões são um **repertório de soluções** e princípios que ajudam os desenvolvedores a criar software e que são codificados em um formato estruturado consistindo de:
Nome, Problema que soluciona e Solução do problema
- O objetivo dos padrões é codificar conhecimento existente de uma forma que possa ser reaplicado em contextos diferentes

Julho 06

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

15

Característica de um padrão

- **Encapsulamento:** um padrão encapsula um problema/solução bem definido. Ele deve ser independente, específico e formulado de maneira a ficar claro onde ele se aplica.
- **Generalidade:** todo padrão deve permitir a construção de outras realizações a partir deste padrão.
- **Equilíbrio:** quando um padrão é utilizado em uma aplicação, o equilíbrio dá a razão, relacionada com cada uma das restrições envolvidas, para cada passo do projeto.

Julho 06

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

16

Característica de um padrão

- **Abstração:** os padrões representam abstrações da experiência empírica ou do conhecimento cotidiano.
- **Abertura:** um padrão deve permitir a sua extensão para níveis mais baixos de detalhe.
- **Combinatoriedade:** os padrões podem ser relacionados hierarquicamente. Padrões de alto nível podem ser compostos ou relacionados com padrões que endereçam problemas de nível mais baixo.

Formato de descrição de um padrão

- **Nome:** uma descrição da solução, mais do que do problema ou do contexto.
- **Exemplo:** uma ou mais figuras, diagramas ou descrições que ilustrem um protótipo de aplicação.
- **Contexto:** a descrição das situações sob as quais o padrão se aplica.
- **Problema:** uma descrição das forças e restrições envolvidos e como elas interagem.
- **Solução:** relacionamentos estáticos e regras dinâmicas descrevendo como construir artefatos de acordo com o padrão. Inclui referências a outras soluções e o relacionamento com outros padrões de nível mais baixo ou mais alto.

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 é 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**).