

Software Pattern (Padrões de Software)

SCE 186 – Engenharia de Software
Profa. Dra. Elisa Yumi Nakagawa
2o semestre de 2007

Material elaborado por pesquisadores e colaboradores do Labes.

1

Introdução

- Padrões de software: descrição de soluções para determinados problemas que ocorrem frequentemente durante o desenvolvimento de sistemas
- Vantagens:
 - Aumento de produtividade
 - Uniformidade na estrutura do software
 - Aplicação imediata por outros desenvolvedores
 - Redução da complexidade: blocos construtivos

2

Introdução

- É importante saber aplicar o padrão correto e de maneira correta.
- Os padrões devem ser lidos, entendidos e incorporados ao modelo mental de projeto de software. Só assim poderão ser utilizados no projeto de algum software.

Histórico

- Origem: Christopher Alexander (padrões de arquitetura para construção de casas, bairros, cidades, etc.)
- Em software:
 - Peter Coad: 1992- padrões de análise
 - James Coplien - 1992 - padrões de programação
 - Gamma et al - 1994 - padrões de projeto
 - Primeira conferência PLOP - 1994

4

Tipos de Padrão

- Existem padrões em diversos níveis de abstração:
 - Padrões de Análise
 - Padrões de Projeto
 - Padrões de Interface
 - Padrões Arquiteturais
 - Padrões de Programação
 - Padrões de Processo
 - Padrões de Padrão

5

Padrões de Análise

- Soluções para problemas de análise: esses padrões embutem conhecimento sobre o domínio de aplicação
- Exemplo:
 - Padrão: Time-Association
 - Problema: Duas entidades possuem uma associação temporal importante
 - Solução: crie uma classe para cada uma das entidades e uma classe representando a associação temporal

6

Padrões de Projeto

- Soluções para problemas de projeto de software
- Exemplo:
 - Padrão: Iterator
 - Problema: Um agregado precisa ser percorrido de diferentes maneiras
 - Solução: Defina uma classe abstrata Iterator e utilize instâncias desse iterator para fazer cada percurso desejado.

7

Padrões de Interface

- Soluções para problemas comuns no projeto da interface de sistemas
- Exemplo:
 - Padrão: Alternate subforms
 - Problema: como projetar uma janela na qual são necessários conjuntos diferentes de dispositivos de acordo com dados da aplicação.
 - Solução: Monte um formulário usando sub-formulários menores. Crie um subformulário para cada variação dos dispositivos ...

8

Padrões Arquiteturais

- Expressam o esquema ou organização estrutural fundamental de sistemas de software.
- Exemplo:
 - Padrão: Model-View-Controller
 - Problema: interfaces voláteis podem afetar a manutenibilidade do sistema como um todo
 - Solução: desvincular a interface do modelo e do controlador ...

9

Padrões de programação

- Soluções de programação particulares de uma determinada linguagem
- Exemplo: C++ Idioms (Coplien 92)
 - Idiom: "forma canônica ortodoxa"
 - Problema: funcionamento de um objeto apresenta problemas devido a não existência de um padrão de implementação
 - Solução: Uma classe contém sempre: um construtor *default*, um construtor *copy*, um operador de atribuição, e um destrutor

10

Padrões de processo

- Definem soluções para os problemas encontrados no processo de desenvolvimento de software
- Exemplo: Patterns for evolving frameworks
 - Padrão: Three examples
 - Problema: como começar a projetar um framework
 - Solução: desenvolva 3 aplicativos que você considera que o framework poderia ajudar a construir

11

Padrões de padrão

- Tentam uniformizar a estrutura básica de um padrão
- Exemplo: *Templates* fornecidos por diversos autores de padrão, entre eles:
 - Brad Appleton (reuniu diversos em apenas um)
 - Gang of Four (GoF): Gamma, Helm, Johnson e Vlissides, autores do primeiro livro sobre padrões
 - Pattern Language for writing patterns

12

Estrutura básica 1 - GoF

- **Pattern Name (Scope, Purpose)**
 - O nome do padrão capta sucintamente a essência do padrão. Um bom nome é vital por poder se tornar parte do vocabulário de projeto
- **Intent**
 - Uma descrição sucinta que responda: O que o padrão faz? Qual seu objetivo e razão de ser? Que problema ele enfoca?

13

Estrutura básica 1 - GoF

- **Also Known As**
 - Outros nomes conhecidos para o padrão, se existirem
- **Motivation**
 - Um cenário que ilustre o problema de projeto e como os elementos do padrão resolvem o problema. O cenário ajuda a entender a descrição mais abstrata do padrão que se segue.

14

Estrutura básica 1 - GoF

- **Applicability**
 - Quais as situações nas quais o padrão de projeto pode ser aplicado? Quais são os exemplos de projetos mal feitos que o padrão pode resolver? Como reconhecer tais situações ?
- **Structure: esquema do padrão**
- **Participants**
 - Classes e objetos que participam do padrão de projeto e suas responsabilidades

15

Estrutura básica 1 - GoF

- **Collaborations**
 - Como os participantes colaboram para satisfazer suas responsabilidades
- **Consequences**
 - Como o padrão alcança seus objetivos? Quais são os compromissos e resultados do uso do padrão? Quais aspectos da estrutura do sistema ele permite que sejam variados independentemente ?

16

Estrutura básica 1 - GoF

- **Implementation**
 - Quais as armadilhas, sugestões ou técnicas que devem ser conhecidas para implementar o padrão? Existem aspectos específicos da linguagem?
- **Sample Code and Usage**
 - Fragmentos de código que ilustram como o padrão pode ser implementado.

17

Estrutura básica 1 - GoF

- **Program Listing**
- **Known Uses**
 - Exemplos do padrão encontrados em sistemas reais.
- **Related Patterns**
 - Quais padrões de projeto são intimamente relacionados com este? Quais são as diferenças importantes ? Com quais outros padrões este deve ser usado?

18

Estrutura básica 2 - Brad Appleton

- Name: deve ser significativo
 - Pode haver sub-seções Aliases e Also known as
- Problem: qual é a intenção do padrão ?
- Context: aplicabilidade, pré-condições
- Forces: motivação para aplicação do padrão
- Solution: como solucionar o problema ?
 - Pode haver sub-seções Structure, participants
- Examples: ilustram a aplicação do padrão

19

Estrutura básica 2 - Brad Appleton

- Related patterns:
 - relacionamento com outros padrões da mesma linguagem
 - padrões alternativos que oferecem outra solução para o mesmo problema mas sob outras forças e contexto.
 - Padrões que podem ou devem ser aplicados simultaneamente com esse padrão
- Known uses: ajudam a validar que o padrão é uma solução provada p/ problema recorrente

20

Exercício

- Escolher um dos padrões de projeto do livro do GoF que achar mais interessante.
- Apresentar na próxima aula.
- Exercício em grupo.
- Livro:
Design Pattern
Gamma, Helm, Johnson e Vlissides (GoF)
Addison-Wesley, 1998.

21

Sites Sobre Padrões

- The Patterns Home Page
 - <http://hillside.net/>
- Ward Cunningham's wonderful WikiWikiWeb
 - <http://c2.com/cgi/wiki?WelcomeVisitors>
- Brad Appleton's Software Patterns Links
 - <http://www.cmcrossroads.com/bradapp/links/sw-pats.html>

22