

Software Reuse Lecture 13

## Review to 1<sup>st</sup> Exam

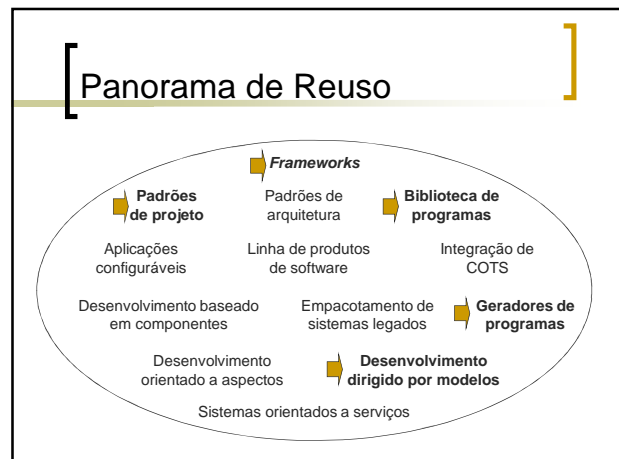
Eduardo Figueiredo  
<http://www.dcc.ufmg.br/~figueiredo>  
[reuso.software@gmail.com](mailto:reuso.software@gmail.com)  
 15 September 2014

## Aula 01: Apresentação

- Bibliografia
- Método de avaliação
  - Provas
  - Exercícios
  - Trabalho Prático
  - Monografia
- Site e email da disciplina
- Agenda do curso

## Aula 02: Técnicas de Reuso

- Introdução a Reutilização de Software
- Técnicas de Reutilização
  - Bibliotecas
  - Frameworks
  - Padrões de Projeto
  - Reuso de Modelos



## Bibliografia da Aula 02

- Ian Sommerville. **Engenharia de Software**, 9<sup>a</sup> Edição. Pearson Education, 2011.
  - Cap. 16 Reuso de Software
  - Seção 5.5 Engenharia Dirigida por Modelos

## Aula 03: POO e Padrões

- Reutilização em Programação Orientada a Objetos
  - Classes
  - Bibliotecas
  - Frameworks
- Padrões de Projeto
  - Padrões de Criação
  - Padrões Estruturais
  - Padrões Comportamentais

## Os 23 Padrões de Projeto

- Os 23 padrões de projeto mais conhecidos foram popularizados pelo livro de E. Gamma, R. Helm, R. Johnson e J. Vlissides
  - Conhecido como Gang-of-Four (GoF)



## Padrões de Criação

- Abstract Factory
- Builder
- **Factory Method**
- Prototype
- **Singleton**

## Padrões Estruturais

- **Adapter**
- Bridge
- **Composite**
- **Decorator**
- Facade
- Flyweight
- Proxy

## Padrões Comportamentais

- **Chain of Responsibility (CoR)**
- Command
- Interpreter
- Iterator
- **Mediator**
- Memento
- **Observer**
- **State**
- Strategy
- Template Method
- Visitor

## Bibliografia da Aula 03

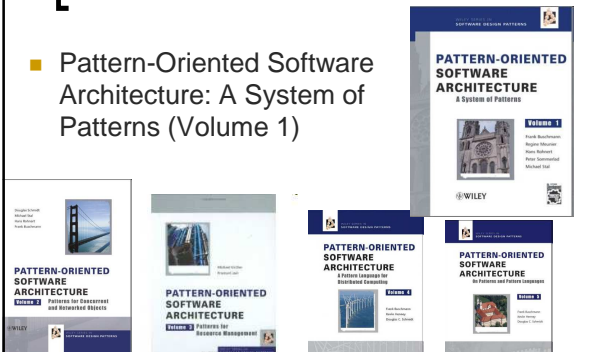
- E. Gamma, R. Helm, R. Johnson, J. Vlissides. **Padrões de Projeto**, 1a. Edição. Bookman, 2000.
  - Capítulo 1
  - Padrões: Factory Method e Singleton; Adapter, Composite e Decorator; Chain of Responsibility, Mediator, Observer e State

## Aula 04: Arquitetura e Padrões

- Arquitetura de Software
- Padrões arquiteturais

## Padrões Arquiteturais: Livros

- Pattern-Oriented Software Architecture: A System of Patterns (Volume 1)



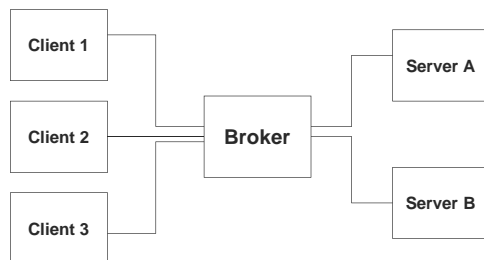
## Da Desordem a Estrutura

- Layered Architecture
  - Arquitetura em Camadas
- Blackboard
  - Arquitetura de Repositório
- Pipes and Filters
  - Dutos e Filtros

## Padrões de Arquitetura

- Sistemas Distribuídos
  - Client-Server (Cliente-Servidor)
  - **Broker**
- Sistemas Interativos
  - Model-View-Controller (MVC)
  - **Presentation-Abstraction-Control**
- Sistemas Adaptáveis
  - **Microkernel**
  - Reflection

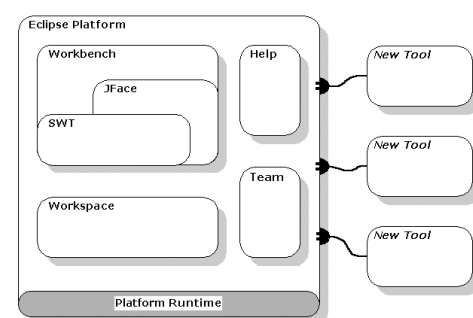
## Broker Structure



## PAC Agents

- Every agent consists of three parts: Presentation, Abstraction and Control
- Presentation
  - Interface of an agent (visible behavior)
- Abstraction
  - Data model of an agent
- Control
  - Connects presentation and abstraction
  - Allows agents communicate to each other

## Example of Microkernel



## Bibliografia da Aula 04

- Ian Sommerville. **Engenharia de Software**, 9a. Edição. 2011.
  - Cap. 6 Projeto de Arquitetura
- F. Buschmann et al. **Pattern-Oriented Software Architecture: A System of Patterns**. John Wiley & Sons, 1996.
  - Cap. 2 Architectural Patterns

## Aula 05: Idiomas

- Idiomas de Programação Java
  - 23 idiomas de programação Java
- Group Work
- Final Project
- Paper Presentation

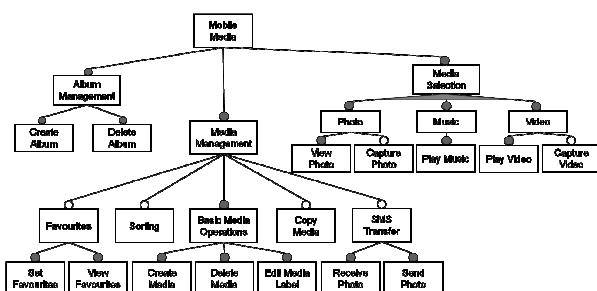
## Bibliografia da Aula 05

- F. Buschmann et al. **Pattern-Oriented Software Architecture: A System of Patterns**. John Wiley & Sons, 1996.
  - Cap. 4 Idioms
- A. von Staa. **Programação Modular**. Elsevier, 2000.
  - Apêndices 3, 4 e 5

## Aula 06: Linha de Produtos

- Linha de produtos de software
  - Motivação e conceitos principais
  - Extração de uma linha de produtos
  - Modelo de características
- Técnicas para linha de produtos
  - Ferramentas para variabilidade
  - Técnicas de implementação

## Um Modelo de Características



## Ferramentas

- Feature Modeling Plug-in (FMP)
- XFeature
- Pure::Variants
- SPLOT
- FeatureIDE

## [ Técnicas de Implementação ]

- Compilação condicional
  - Antenna
- Programação orientada a aspectos
  - AspectJ
- Programação orientada a características
  - AHEAD

## [ Bibliografia da Aula 06 ]

- I. Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
  - Cap. 16 Reuso de Software (Seção 16.3)
- K. Pohl, G. Bockle, F. Linden. **Software Product Line Engineering: Foundations, Principles and Techniques**, 1st edition, Springer, 2005.
  - Capítulos 1 e 2
- Páginas das ferramentas e artigos

## [ Aula 07: Ferramenta FeatureIDE ]

- Exercício prático
- Criar modelo de característica usando a ferramenta
- Avaliar a ferramenta

## [ Aula 08: LPS e Jogos ]

- Jogos para ensino e treinamento em Engenharia de Software
  - Problems and Programmers
  - SimulES

## [ Bibliografia da Aula 08 ]

- A. Baker, E. Navarro, and A. van der Hoek. **An Experimental Card Game for Teaching Software Engineering Processes**. Journal of Systems of Software (JSS), 2004.
- Figueiredo, E.; Lobato, C.; Dias, K.; Leite, J. e Lucena, C. **“Um Jogo para o Ensino de Engenharia de Software Centrado na Perspectiva de Evolução”**. Workshop sobre Educação em Computação (WEI), Rio de Janeiro, pp. 37-46, 2007.

## [ Aula 09: Jogar SimulES e PnP ]

- Estudo de caso do trabalho prático
  - Jogar os jogos educacionais
- Exercício sobre modelo de características dos jogos

## Aula 10: Aspectos

- Separação de Interesses
- Desenvolvimento de Software Orientado a Aspectos (DSOA)
- A Linguagem AspectJ

## Construções de AspectJ

- Pontos de Junção
- Pontos de Corte
- Adendos
- Declarações Intertipo

## Bibliografia da Aula 10

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
  - Capítulo 21
- R. Laddad. **AspectJ in Action**, 2ª Ed. 2010.
  - Part 1 Understanding AOP and AspectJ
- Sergio Soares. **Programação Orientada a Aspectos com AspectJ**. Minicurso CBSoft 2010.

## Aula 11: Artigos

1. M. Eaddy, T. Zimmermann, K. D. Sherwood, V. Garg, G. C. Murphy, N. Nagappan, A. V. Aho. *Do Crosscutting Concerns Cause Defects?* IEEE Transactions on Software Engineering, 34(4), 2008. [Danilo]
2. J. Hannemann and G. Kiczales. *Design Pattern Implementation in Java and AspectJ*. Conference on OO Programming, Systems, Languages, and Applications (OOPSLA), 2002. [Tamara]

## Aula 12: Implementação

- Implementar o interesse do padrão **Observer** em AspectJ
  - Usar a implementação equivalente em Java (atual) como referência
  - Consultar a solução do Hannemann e Kiczales para o padrão *Observer*
- Pode usar uma solução diferente da solução do Hannemann e Kiczales
  - Mas, a mudança deve ser justificada

## Próxima Aula

- Prova 1
- A prova começa as **9:25 hrs** e termina as **11:05 hrs**.