

Módulo III

Padrões GOF:

AbstractMethod

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

- Padrões GOF
 - Abstract Factory

Julho 06

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

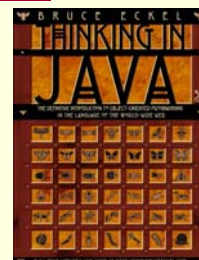
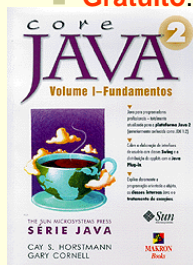
2

Bibliografia

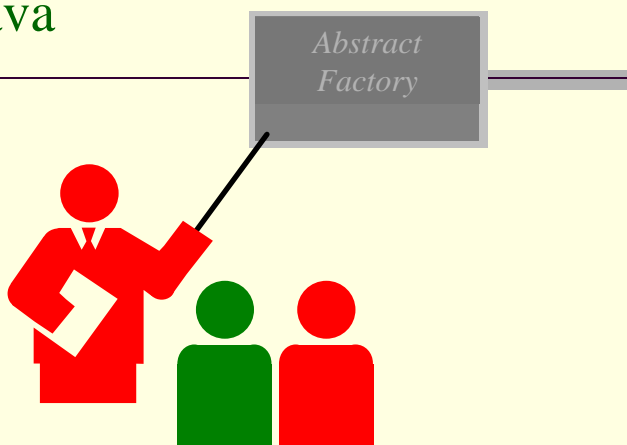
- *Eric Gamma, et ali, Padrões de Projeto, Ed Bookman*
- *Martin Fowler, Analysis Patterns - Reusable Object Models, Addison-Wesley, 1997*
- *Martin Fowler, Refatoração - Aperfeiçoando o projeto de código existente, Ed Bookman*

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/>



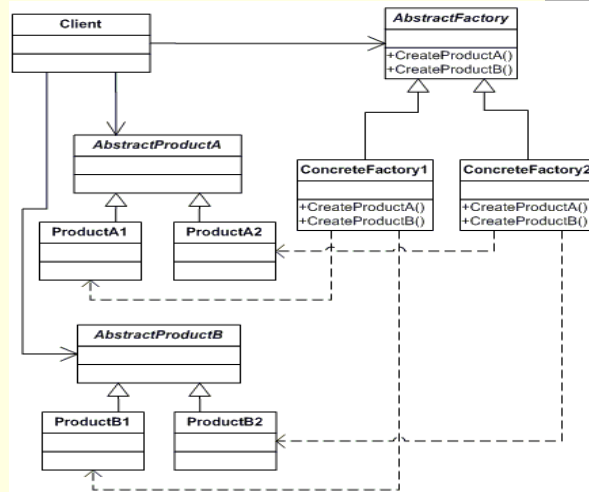
POO-Java



Abstract Factory

- **Intenção:** fornecer uma interface comum para a criação de famílias de objetos relacionados ou dependentes sem especificar suas classes concretas.
- **Solução:** crie uma interface para representar uma fábrica para cada família de objetos. As subclasses concretas instanciam cada família específica.

Abstract Factory (estrutura)



Julho 06

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

7

Abstract Factory (participantes)

- **Fábrica Abstrata**: declara uma interface para operações criam objetos-produto abstratos;
- **Fábrica Concreta**: implementa as operações para criar objetos-produto concretos;
- **Produto Abstrato**: declara uma interface para um tipo de objeto produto.
- **Produto Concreto**: implementa a interface abstrata de Produto Abstrato e define um objeto-produto a ser criado pela Fábrica Concreta correspondente.
- **Cliente**: utiliza apenas as interfaces declaradas por Fábrica Abstrata e Produto Abstrato.

Julho 06

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

8

Abstract Factory (conseqüências)

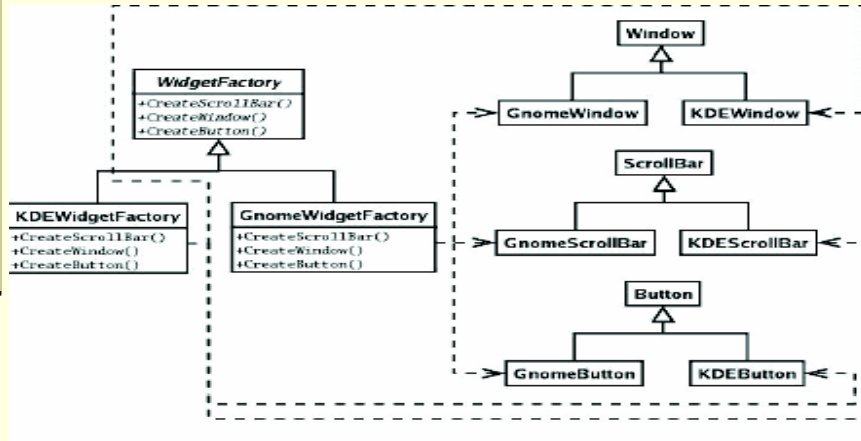
- 👉 **Isola classes concretas:** uma vez que uma fábrica encapsula a responsabilidade e o processo de criação de objetos-produto, ela isola clientes das classes de implementação.
- 👉 **Fica mais fácil a troca de uma família de produtos,** bastando trocar a fábrica concreta usada pela aplicação.
- 👉 **Promove consistência** entre produtos.
- 👉 **O suporte a novos tipos de produtos é dificultado,** já que a interface definida em AbstractFactory fixa o conjunto de produtos que podem ser criados.

Abstract Factory (aplicabilidade)

- Quando o sistema deve ser independente de como seus produtos são criados, compostos e representados.
- Quando o sistema deve ser configurado com uma dentre várias famílias de produtos.
 - É necessário fornecer uma biblioteca de classes, mas não é desejável revelar que produto particular está sendo usado.
- Quando uma família de produtos relacionados foi projetada para ser usada em conjunto, e esta restrição tem de ser garantida.

Abstract Factory (exemplo)

- Exemplo: portabilidade entre bibliotecas GUI (Gnome,KDE)



Julho 06

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

11