

PROJETO DE SOFTWARE ORIENTADO A OBJETO

PROGRAMA

1. Introdução

- 1.1 Análise e Projeto OO
- 1.2 O Processo de Desenvolvimento de Software
- 1.3 Modelos e Artefatos

2. Conceitos Básicos de Padrões de Projeto (Revisão de conceitos de Projeto)

- 2.1 Padrão para atribuir responsabilidades: Expert
- 2.2 Padrão para atribuir responsabilidades: Creator
- 2.3 Padrão para atribuir responsabilidades: Baixo Acoplamento
- 2.4 Padrão para atribuir responsabilidades: Alta Coesão
- 2.5 Padrão básico: Interfaces e Polimorfismo
- 2.6 Padrão básico: Herança versus Composição

3. Padrões de Projeto Arquitetural

- 3.1 Projeto de uma Arquitetura de Software
- 3.2 Padrão Layers (Camadas)
- 3.3 Estruturas de Controle
- 3.4 Padrão Observer
- 3.5 Padrão Model-View-Controller (MVC)
- 3.6 Opções de Persistência
- 3.7 Finalmentes
- 3.8 Resumo: Perguntas a fazer ao elaborar um projeto arquitetural

4. Padrões de Projeto Detalhado

- 4.1 Facetas da Reusabilidade de Software

- 4.2 Elementos essenciais de um Design Pattern
- 4.3 Design Pattern: Factory Method
- 4.4 Design Pattern: Singleton
- 4.5 Design Pattern: Abstract Factory
- 4.6 Design Pattern: Iterator
- 4.7 Design Pattern: Composite
- 4.8 Design Pattern: Visitor
- 4.9 Design Pattern: Strategy
- 4.10 Design Pattern: Command
- 4.11 Design Pattern: Decorator
- 4.12 Design Pattern: Template Method
- 4.13 Comentários Finais sobre Design Patterns

5. Software Baseado em Componentes

- 5.1 O que são componentes?
- 5.2 Aspectos técnicos do desenvolvimento baseado em componentes
- 5.3 Construção de componentes: JavaBeans
- 5.4 Server components: Java 2 Enterprise Edition
- 5.5 A evolução de componentes e comparação de arquiteturas

6. Frameworks

- 6.1 O que é um framework?
- 6.2 Vantagens e desvantagens no uso de frameworks
- 6.3 Tipos de frameworks
- 6.4 Uma lista de frameworks
- 6.5 Detalhamento de um framework horizontal: JUNIT
- 6.6 Detalhamento de um framework vertical: Accounts
- 6.7 Construção de frameworks: Um processo evolutivo

[map-2](#) [home](#)