Arquitetura e Projeto de Sistemas

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Rio Grande
Divisão de Computação

Agenda

Introdução

Tecnologias/Ferramentas Utilizadas

- Linguagem de Programação: **JAVA**
- SGBD: **PostgreSQL**
- IDE: **NetBeans**
- Códigos e Exemplos: **Github**
- Slides e Materiais:
  - `Moodle/Github`
- Comunicação:
  - `Discord`

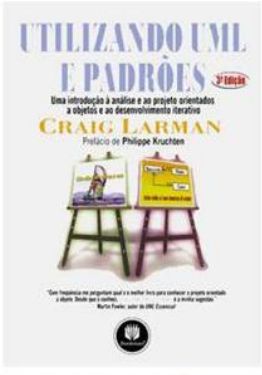
Programa

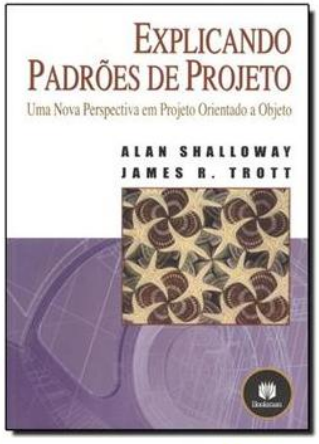
- Revisão O.O
- Introdução a Padrões de Projeto
- Conectividade JDBC
- Padrões Arquiteturas de Persistência
- Padrões de Projeto de Software: Criacional
- Padrões de Projeto de Software: Comportamental
- Padrões de Projeto de Software: Estrutural
- Testes Unitários
- Boas Práticas de Programação

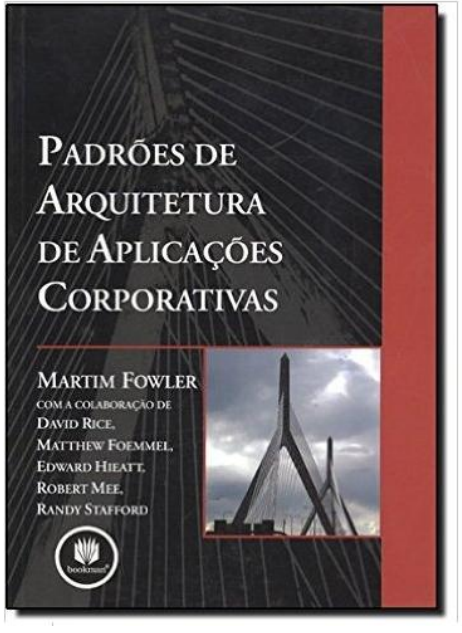
Atividades Avaliadas

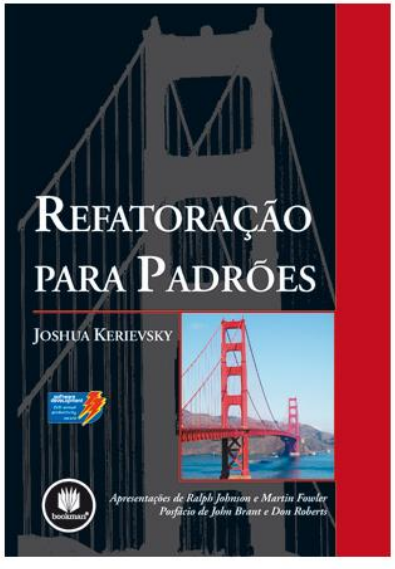
- Prazos estipulados (geralmente de 1 semana)
- Tira-dúvidas: atendimento e aulas tira-dúvidas
- Envio por e-mail

Referências Bibliográficas









Introdução a Padrão de Projetos

- **Padrões de projeto** são soluções típicas para problemas comuns em projeto de software.
  - **Entretanto:** Para ser um padrão, uma solução não basta ser recorrente, mas precisa ser uma boa solução (caso contrário é um anti-padrão!)
- O padrão não é um pedaço de código específico, mas um conceito geral para resolver um problema em particular.
- Você não pode apenas encontrar um padrão e copiá-lo para dentro do seu programa, como você faz com funções e bibliotecas que encontra por aí.
  - Os desenvolvedores devem avaliar o contexto e o problema que querem resolver e adaptar, e combinar padrões de forma a equilibrar as forças envolvidas.

Do que consiste um padrão?

- O **Propósito** do padrão descreve brevemente o problema e a solução.
- A **Motivação** explica a fundo o problema e a solução que o padrão torna possível.
- As **Estruturas** de classes mostram cada parte do padrão e como se relacionam.
- **Exemplos de código** em uma das linguagens de programação populares tornam mais fácil compreender a ideia por trás do padrão.

Alguns catálogos de padrões de projeto listam outros detalhes úteis, tais como a aplicabilidade do padrão, etapas de implementação, e relações com outros padrões.

Por que devo aprender padrões?

- Você pode conseguir trabalhar como um programador por muitos anos sem saber sobre um único padrão.
- Ainda assim, contudo, você estará implementando alguns padrões mesmo sem saber.
- **Então, por que gastar tempo para aprender sobre eles?**

Por que devo aprender padrões?

- Os **padrões de projeto** são um kit de ferramentas para soluções tentadas e testadas para problemas comuns em projeto de software
- Os padrões de projeto definem uma linguagem comum que você e seus colegas podem usar para se comunicar mais eficientemente.
  - Você pode dizer: "Oh, é só usar um *Singleton* para isso?" e todo mundo vai entender a ideia por trás da sua sugestão.

Características de um bom projeto

Antes de prosseguirmos para os próprios padrões, vamos discutir o processo de arquitetura do projeto de software: coisas que devemos almejar e coisas que devemos evitar.

- **Reutilização de código**
  - A reutilização de código é um dos modos mais comuns para se reduzir custos de desenvolvimento.
- **Extensibilidade**
  - **Mudança** é a única constante na vida de um programador
    - Você criou um framework de interface gráfica com botões quadrados, mas, meses mais tarde, botões redondos é que estão na moda.

Princípios de projeto

- **Encapsule o que varia:** Identifique os aspectos da sua aplicação que variam e separe-os dos que permanecem os mesmos.
  - O objetivo principal deste princípio é minimizar o efeito causado por mudanças.
- **Programe para uma interface, não uma implementação**
  - Programe para uma interface, não uma implementação.
  - Dependa de abstrações, não classes concretas.
  - **O código torna-se mais flexível**.

Classificação dos Padrões de Projeto

- A disseminação dos padrões na comunidade de desenvolvimento de software iniciou-se com o conhecido livro *Design Patterns: Elements of Reusable Object-Oriented Software*
- Esse livro também é conhecido como **GoF**, um acrônimo de *Gang of Four*, uma referência aos seus quatro autores.
- Segundo o livro, os padrões são divididos em três categorias:
  1. Criação,
  2. Estrutural
  3. Comportamental

Todos os padrões destas categorias tem um conjunto de características específicas que motivam a categorização deles.

Padrões de Criação

Os **padrões de criação** tem como intenção principal abstrair o processo de **criação de objetos, ou seja, a sua instanciação.**

Os **padrões criacionais** fornecem vários mecanismos de criação de objetos, que aumentam a flexibilidade e reutilização de código já existente.

Desta maneira o sistema não precisa se preocupar com questões sobre, como o objeto é criado, como é composto, qual a sua representação real.

Padrões Estruturais

- Os **padrões estruturais** vão se preocupar em **como as classes e objetos são compostos**, ou seja, como é **a sua estrutura**.
- O objetivo destes padrões e facilitar o design do sistema identificando maneiras de realizar o relacionamento entre as entidades, deixando o desenvolvedor livre desta preocupação.

Padrões Comportamentais

- **Padrões comportamentais** são voltados aos algoritmos e a designação de responsabilidades entre objetos.
- Os **padrões comportamentais** atuam sobre **como responsabilidades são atribuídas as entidades**, ou seja, qual o comportamento das entidades.
- Estes padrões facilitam a comunicação entre os objetos, distribuindo as responsabilidades e definindo a comunicação interna.

Classificação dos Padrões de Projeto GoF

A figura a seguir permite visualizar a classificação dos padrões de projeto, de acordo com os critérios falados:

Escopo	1. Criação	2. Estrutura	3. Comportamento
Classe	Factory Method	Class Adapter	Interpreter Template Method
Objeto	Abstract Factory Builder Prototype Singleton	Object Adapter Brigde Composite Decorator Façade Flyweight Proxy	Chain of Responsability Command Iterator Mediator Memento Observer State Strategy Visitor

Arquitetura e Projeto de Sistemas

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Rio Grande
Divisão de Computação