

Análise Orientada a Objetos

Outros tópicos de análise e design



cursos online




1

Tópicos Abordados


- *DDD: Domain-Driven Design*
- *TDD: Test-Driven Development*
- Refatoração
- *Design Patterns*
 - *GoF Design Patterns*



2

DDD

- **D**omain-**D**riven **D**esign
- É uma abordagem que permite criar projetos de software centrados no modelo do domínio
- As classes que fazem parte do domínio implementam a lógica de negócio
- Objetos do domínio colaboram para que a aplicação funcione



3

TDD



- **T**est-**D**riven **D**evelopment
- O desenvolvimento começa pelos testes
- Vantagens
 - Os testes são realmente escritos
 - Obriga o desenvolvedor a pensar em como a interface com a funcionalidade deverá funcionar mesmo antes de ela existir
 - Diminui a possibilidade de erro quando o código precisa ser alterado
- A IDE deve dar o suporte necessário à criação e execução dos testes de forma automatizada



4

Refatoração (*refactoring*)



- É uma forma de reescrever e reestruturar o código sem alterar o seu comportamento externo
- Exemplos
 - Extrair um trecho de código para um método
 - Renomear uma variável
 - Adicionar uma classe ou interface na hierarquia de outra classe
 - etc.
- Importância
 - Remover código duplicado
 - Reduzir métodos muito grandes
 - Remover constantes que são *hard-coded*
 - etc.



5


Refatoração (*refactoring*)



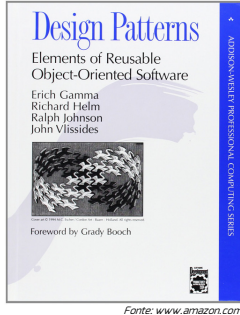
- Quando fazer o refactoring?
 - Quando você, com sua experiência, percebe que algo pode estar errado
 - Método muito grande
 - Grande dependência entre classes
 - Código duplicado
 - etc.
- A IDE deve dar suporte à execução de operações de refactoring de forma simplificada




6


Design Patterns


- O conceito de design pattern foi difundido em 1994 com o lançamento do livro *“Design Patterns”*
- Este livro tem grande influência na engenharia de software e é uma referência no aprendizado de orientação a objetos
- Os 4 autores são conhecidos com *GoF* (*Gang of Four*)







7

Design Patterns


- São modelos voltados à solução de um problema de desenvolvimento já existente
 - Não dependem de uma linguagem de programação específica
 - São modelos genéricos
- Voltados à programação orientada a objetos
- Vantagens
 - Auxiliam no desenvolvimento
 - Problema e solução já conhecidos
 - Estabelecem uma linguagem comum
 - Isto ajuda quando desenvolvedores precisam dar manutenção em softwares criados por outras pessoas




8

Definindo Design Patterns



- Um design pattern é definido através dos seguintes elementos

| | |
|---------|---------------|
| Nome | Problema |
| Solução | Consequências |



9

GoF Design Patterns



- O livro do GoF contém 23 design patterns

Creational Patterns

Factory Method

Abstract Factory

Builder

Prototype

Singleton

Adapter

Bridge

Composite

Structural Patterns

Decorator

Facade

Flyweight

Proxy

Interpreter

Template Method

Chain of Responsibility

Command

Behavioral Patterns

Iterator

Mediator


Memento

Observer

State

Strategy

Visitor



10



Softblue



11
