



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br



Introdução aos Padrões de Projetos



Design Patterns



Introdução aos Padrões de Projetos

Os padrões de projetos foram inicialmente introduzidos pela GoF (Gang of Four) – Gangue dos Quatro, como solução para determinados problemas da orientação a objetos.



Este livro discute soluções de engenharia de software para problemas que ocorrem comumente em design de software.

Inicialmente foram identificados 23 padrões de projetos e a primeira implementação foi feita com base na linguagem Java. Por fim, vale destacar que os padrões de projeto não são invenções mas sim descobertas.



Introdução aos Padrões de Projetos

Principais características dos padrões de projeto:

- São independentes de linguagem e podem ser implementados em linguagens diferentes;
- São dinâmicos, e novos padrões são introduzidos ou removidos ocasionalmente;
- São passíveis de personalização e, portanto, são úteis aos desenvolvedores;

O interessante é que os padrões de projeto são soluções para problemas conhecidos. Deste modo, eles podem ser muito utilizados na análise ou no design e, conforme esperado, na fase de desenvolvimento, por causa da relação direta com o código da aplicação.



Introdução aos Padrões de Projetos

Taxonomia dos padrões de projeto:

Nem todo código ou design pode ser classificado como um padrão de projeto.

Por exemplo, um trecho de código ou estrutura de dados de programação que resolva um determinado problema não pode ser chamada de padrão.

Por isso devemos entender, mesmo de forma simplista, alguns termos:

- **Trecho de código**: É um código em alguma linguagem de programação que serve a determinado propósito, por exemplo, uma conexão com um banco de dados e etc;
- **Design**: É uma melhor solução para resolver um problema em particular;
- **Convenção**: É uma maneira de resolver algum tipo de problema, e pode ser bem genérica e aplicável a uma situação que se tenha;
- **Padrão**: É uma solução eficiente e escalável, resistente ao teste do tempo, que resolverá toda uma classe de problemas conhecidos.



Introdução aos Padrões de Projetos

Contexto de aplicabilidade dos padrões de projeto:

Para que o desenvolvedor tenha sucesso no uso dos padrões de projeto deve-se ter conhecimento do contexto em que os padrões se aplicam.

Podemos classificar o contexto conforme:

- **Participantes**: São as classes usadas nos padrões de projeto. As classes desempenham diferentes papéis para atingir vários objetivos no padrão;
- **Requisitos Não Funcionais**: Requisitos como otimização de memória, usabilidade e desempenho se enquadram nesta categoria. Esses fatores exercem impacto na solução de software e portando são essenciais;
- **Negociações**: Nem todos os padrões de projeto se enquadram no desenvolvimento de aplicações exatamente como estão, portando negociações se fazem necessárias. São decisões que você deve tomar quando usar um padrão de projeto em sua aplicação;
- **Resultados**: Os padrões de projeto podem ter um impacto negativo em outras partes do código caso o contexto não seja apropriado. Os desenvolvedores devem compreender as consequências do uso dos padrões de projeto.



Introdução aos Padrões de Projetos

Padrões para linguagens dinâmicas:

Python é uma linguagem dinâmica e sua natureza pode ser descrita da seguinte forma:

- Tipos ou classes são objetos em tempo de execução;
- As variáveis podem ter um tipo a partir de um valor e podem ser modificadas em tempo de execução;
- Linguagens dinâmicas têm mais flexibilidade no que diz respeito a restrições de classe;
- O polimorfismo está embutido na linguagem;
- Não há palavras reservadas como `private` e `protected`, ou seja, tudo é público por padrão;
- A flexibilidade da linguagem Python faz com que os padrões de projeto possam ser implementados facilmente;



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br