

Exercícios de Fixação – Design Patterns

clenio.silva@uniube.br

1. Qual das alternativas abaixo descreve corretamente o padrão de projeto Singleton?
 - a) O padrão Singleton permite que um objeto seja criado apenas uma vez e fornece um ponto de acesso global a esse objeto.
 - b) O padrão Singleton cria múltiplas instâncias de uma classe, mas mantém o controle sobre elas.
 - c) O padrão Singleton não permite que objetos sejam compartilhados entre diferentes classes.
 - d) O padrão Singleton é usado para criar instâncias de classes com parâmetros dinâmicos.
2. O padrão de projeto Service é utilizado para:
 - a) Implementar a lógica de negócios e fornecer uma camada de abstração entre a interface e a persistência de dados.
 - b) Criar objetos com a responsabilidade de gerenciar suas próprias instâncias.
 - c) Facilitar a implementação de interfaces gráficas em sistemas de software.
 - d) Encapsular objetos que fornecem funcionalidade comum a várias partes do sistema.
3. Qual é o objetivo principal do padrão DAO?
 - a) Criar uma camada intermediária entre a aplicação e a camada de dados, simplificando o acesso a um banco de dados.
 - b) Definir um conjunto de regras para objetos de dados, como objetos de valor.
 - c) Manter a interface entre os dados e a lógica de negócios.
 - d) Garantir que a comunicação entre classes siga as regras do princípio SOLID.
4. O padrão Abstract Factory é usado para:
 - a) Criar famílias de objetos relacionados sem especificar suas classes concretas.
 - b) Garantir a criação de objetos em um formato específico, com base em um protocolo.
 - c) Criar uma interface para a criação de objetos em um sistema, permitindo a troca de implementação.
 - d) Facilitar a reutilização de objetos através de um repositório centralizado.
5. O princípio da Responsabilidade Única (Single Responsibility Principle - SRP) afirma que:

- a) Uma classe deve ter uma única razão para mudar, ou seja, deve ter uma única responsabilidade.
 - b) Cada classe deve ser responsável por realizar um conjunto de tarefas de forma independente.
 - c) Uma classe deve poder mudar de forma independente das demais classes, sem depender de suas funcionalidades.
 - d) O código de cada classe deve ser completamente modular e independente.
6. O princípio Aberto/Fechado (Open/Closed Principle - OCP) preconiza que:
- a) Uma classe deve estar aberta para extensão, mas fechada para modificação.
 - b) As classes devem ser abertas para alterações constantes, mas fechadas para adições de novas funcionalidades.
 - c) A arquitetura de um sistema deve ser completamente aberta para extensões e modificações sem restrições.
 - d) As classes devem ser fechadas para mudanças, mas abertas para inclusão de novos componentes.
7. O padrão MVC (Model-View-Controller) organiza o código da aplicação em três componentes. Qual é a principal função do componente "Controller"?
- a) A View é responsável pela lógica de exibição dos dados, e o Controller manipula as ações do usuário e atualiza o Model.
 - b) O Controller armazena e manipula os dados da aplicação, enquanto a View exibe esses dados.
 - c) O Controller serve apenas para gerenciar o banco de dados e garantir a persistência das informações.
 - d) O Controller é responsável por exibir a interface gráfica para o usuário.
8. De acordo com o Princípio da Substituição de Liskov (Liskov Substitution Principle - LSP), qual das alternativas está correta?
- a) Objetos de uma classe derivada devem ser substituíveis por objetos da classe base, sem alterar o comportamento correto do sistema.
 - b) Classes base devem ser totalmente substituíveis por suas classes derivadas sem causar problemas de desempenho.
 - c) As classes derivadas podem adicionar novos métodos, mas não devem sobrescrever os

métodos da classe base.

d) Objetos derivados podem substituir objetos base, mas somente em contextos específicos.

9. O princípio da Segregação de Interface (Interface Segregation Principle - ISP) recomenda que:

a) Uma classe não deve ser forçada a implementar interfaces que não utilizam.

b) Uma classe deve ser capaz de implementar várias interfaces para garantir flexibilidade.

c) As interfaces devem ser únicas e não ter métodos duplicados em várias classes.

d) Uma interface deve ser projetada para atender a um único tipo de cliente.

10. Qual dos seguintes exemplos melhor caracteriza o uso do padrão Service?

a) Criar uma camada de serviço para processamento de pedidos de compra, separando a lógica de negócios da camada de apresentação.

b) Criar objetos que representam dados persistentes, sem lógica de negócios associada.

c) Definir um padrão de comunicação entre diferentes sistemas em uma arquitetura distribuída.

d) Organizar o código da interface gráfica de um sistema, permitindo a troca de elementos de forma dinâmica.