# Exercício Teórico sobre Princípios SOLID

\*\*Instruções:\*\* Escolha a opção correta para cada pergunta.

1. Qual princípio SOLID se concentra na ideia de que as classes devem ter apenas um motivo para mudar?

a) Princípio da Substituição de Liskov (LSP)

b) Princípio da Responsabilidade Única (SRP)

c) Princípio do Aberto-Fechado (OCP)

d) Princípio da Inversão de Dependência (DIP)

2. Qual princípio SOLID diz que "as classes derivadas devem ser substituíveis por suas classes base"?

a) Princípio da Substituição de Liskov (LSP)

b) Princípio da Responsabilidade Única (SRP)

c) Princípio do Aberto-Fechado (OCP)

d) Princípio da Inversão de Dependência (DIP)

3. O Princípio da Responsabilidade Única (SRP) afirma que uma classe deve ter apenas uma razão para mudar. Qual é o problema de uma classe que viola esse princípio?

a) Baixa coesão

b) Alto acoplamento

c) Falta de encapsulamento

d) Violência no encapsulamento

4. Qual princípio SOLID sugere que as classes devem estar abertas para extensão, mas fechadas para modificação?

a) Princípio da Substituição de Liskov (LSP)

b) Princípio da Responsabilidade Única (SRP)

c) Princípio do Aberto-Fechado (OCP)

d) Princípio da Inversão de Dependência (DIP)

5. O Princípio do Aberto-Fechado (OCP) incentiva o uso de quais conceitos de programação orientada a objetos?

a) Herança e polimorfismo

b) Encapsulamento e composição

c) Acoplamento e coesão

d) Iteração e recursão

6. Qual princípio SOLID se concentra na ideia de que as classes de alto nível não devem depender de classes de baixo nível, mas sim de abstrações?

a) Princípio da Substituição de Liskov (LSP)

b) Princípio da Responsabilidade Única (SRP)

c) Princípio do Aberto-Fechado (OCP)

d) Princípio da Inversão de Dependência (DIP)

7. O Princípio da Inversão de Dependência (DIP) sugere o uso de quais conceitos para reduzir o acoplamento entre módulos?

a) Interfaces e classes abstratas

b) Herança e polimorfismo

c) Composição e injeção de dependência

d) Acoplamento forte e acoplamento fraco

8. Qual dos princípios SOLID enfatiza que as classes devem ter uma única responsabilidade, ou seja, fazer apenas uma coisa e fazê-la bem?

a) Princípio da Substituição de Liskov (LSP)

b) Princípio da Responsabilidade Única (SRP)

c) Princípio do Aberto-Fechado (OCP)

d) Princípio da Inversão de Dependência (DIP)

9. O Princípio da Substituição de Liskov (LSP) é um dos princípios SOLID. Qual é sua principal implicação para as classes derivadas?

a) Elas podem modificar livremente os métodos da classe base.

b) Elas devem herdar todos os atributos da classe base.

c) Elas devem poder substituir os objetos da classe base sem quebrar a corretude do programa.

d) Elas não podem herdar métodos da classe base.

10. Qual dos princípios SOLID é frequentemente aplicado em conjunto com o Padrão de Fábrica (Factory Pattern) para criar objetos sem conhecer as classes concretas?

a) Princípio da Substituição de Liskov (LSP)

b) Princípio da Responsabilidade Única (SRP)

c) Princípio do Aberto-Fechado (OCP)

d) Princípio da Inversão de Dependência (DIP)

11. Considere a seguinte afirmação: "O Princípio da Substituição de Liskov (LSP) é uma extensão do Princípio do Aberto-Fechado (OCP)". Essa afirmação é:

a) Verdadeira

b) Falsa

12. Qual dos seguintes cenários NÃO é uma violação do Princípio da Responsabilidade Única (SRP)?

a) Uma classe que lida com a validação de entrada do usuário e faz cálculos de negócios em um sistema de pedidos.

b) Uma classe que lê dados de um arquivo e os armazena em um banco de dados.

c) Uma classe que gera relatórios em formatos diferentes (PDF, Excel, CSV).

d) Uma classe que gerencia a autenticação do usuário e envia notificações por e-mail.