



D S T Q X S S

Memo No. _____
Date / /

John Borges (Engenharia de Sistemas)

The SOLID principles of

Objected - Oriented Design

O documento é basicamente um capítulo que explica os 5 princípios SOLID de design de software orientado a objetos, criados para gerenciar dependências e tornar o código mais flexível e de fácil manutenção.

Princípio da Responsabilidade Única (SPR): Uma classe deve ter apenas uma responsabilidade. Por exemplo, uma classe que calcula a área de formas não deve também ser responsável por exibir o resultado, pois existem motivos diferentes para mudar.

Princípio Aberto/Fechado (OCP): As classes devem ser "abertas para extensão, mas fechadas para modificação". Em vez de alterar uma classe existente para adicionar um novo tipo de forma, o design deve permitir que novas formas sejam adiciona-

das como novas classes, estendendo uma abstração comum.

Princípio da Substituição de Liskov (LSP)

Uma classe filha deve ser capaz de substituir sua classe mãe sem quebrar o programa. O texto exemplifica que uma classe Quadrado herdando de Retângulo pode violar este princípio, pois um quadrado não se comporta exatamente como um retângulo em todos os contextos de código, e por isso não é um substituto perfeito.

Princípio da Segregação de Interface (ISP)

É melhor ter interfaces menores e mais específicas de que uma única interface grande. Isso evita que uma classe seja forçada a implementar métodos que não utiliza, favorecendo um design mais desacoplado.

Princípio da Inversão de Dependência (DIP)

O código deve depender de abstrações (como interfaces), mas de implementações concretas. Em vez de uma classe criar suas próprias dependências, elas devem ser fornecidas de fora (injeção de dependência), o que aumenta a flexibilidade e o desacoplamento.