

« . . »

Resumo 5 - Matheus Bojrim de Souza Bosco

RA: 24201178

The SOLID Principles of Object-Oriented Design

A leitura aprofunda-se nos cinco princípios definidos por Robert C. Martin como diretrizes para tornar o design de software mais claro, flexível e de fácil manutenção. O texto insiste explicando que a verdadeira essência da orientação a objetos não é apenas "modelar o mundo real", mas sim controlar dependências para evitar código rígido, frágil e pouco reutilizável.

O primeiro princípio, SRP (Single Responsibility Principle), afirma que uma classe deve ter apenas um motivo para mudar. O exemplo dado mostra como separar o cálculo de áreas da responsabilidade de exibir resultados, evitando que alterações de saída afetem a lógica de cálculo. O OCP (Open/Closed Principle) defende que classes devem estar abertas para extensão, mas fechadas para modificação, ilustrado pelo uso de uma classe abstrata Shape que permite criar novas figuras sem alterar o código já existente.

O LSP (Liskov Substitution Principle) destaca que subclasses devem poder substituir suas superclasses sem alterar o comportamento esperado, discutindo o clássico problema de modelar quadrados como retângulos. Já o ISP (Interface Segregation Principle) recomenda criar interfaces específicas e coesas em vez de interfaces grandes e genéricas, facilitando a composição de comportamentos independentes. Por fim, o DIP (Dependency Inversion Principle) enfatiza depender de abstrações e não de implementações concretas, explorando o uso de injeção de dependência via composição para reduzir acoplamento.



« . . . »

No conjunto, o texto mostra que os princípios SOLID reforçam e expandem os conceitos centrais da POO (encapsulamento, herança, polimorfismo e composição) ao mesmo tempo em que promovem boas práticas modernas, como favorecer composição sobre herança. A conclusão é que dominar esses princípios não só previne armadilhas comuns de design, como também cria bases sólidas para sistemas mais escaláveis e sustentáveis.