

Sólido

O capítulo explica que a programação Orientada a Objetos não deve ser visto apenas como uma forma de "imitar o mundo real" mas sim como um meio de organizar e reduzir dependências no código.

Deslocando assim sistemas mal projetados que tendem a se tornar:

- Rígidos
- Groxos
- Inerentes

Para lidar com esses problemas, surgem os princípios **Solid**, um guia de boas práticas para manter o software flexível, sustentável e de fácil manutenção.

Os 5 princípios:

SRP

- Uma classe deve ter apenas um motivo para mudar
- ↳ misturar diferentes responsabilidades gera acoplamento desnecessário

OCP

- ↳ Classes devem estar abertas para extensão, mas fechadas para modificação.
- ↳ É melhor adicionar novas funcionalidades por meio de herança ou composição, em vez de alterar o código existente.

(LSP)

↳ Uma classe deve poder substituir sua superclasse sem quebrar o programa.

(ISP)

↳ É melhor ter várias interfaces pequenas e específicas do que uma interface única com métodos desnecessários.
↳ Assim, classes implementam apenas o que realmente precisam.

(DIP)

↳ o código deve depender de abstrações, não de implementações concretas.

No final o autor reforça que os princípios SOLID estão ligados a conceitos fundamentais da POO.