

Sólido

O capítulo explica que o 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.

Destacando assim sistemas mal projetados que tendem a se tornar:

- Bróquios
- Monólogos
- Sincrônios

Para lidar com esses problemas, surgem os principios SOLID, um guia de boas práticas para manter o software flexível, testável e de fácil manutenção.

Os 5 principios:

SRP

- uma classe deve ter apenas um motivo para mudar
- manter diferentes responsabilidades gera acoplamento desnecessário

OCP

As 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)

↳ Um subclasse deve poder substituir seu superclasse sem quebrar o programa.

(ISP)

↳ É melhor ter várias interfaces pequenas e específicas do que uma interface unica com métodos desnecessários.
↳ Além, 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 principios SOLID estão ligados a conceitos fundamentais da POO.