

Helanna Marcano Pais - 24101010

31 / 3 / 25

The solid principle of OO Design - Resumo POO

→ O capítulo se inicia falando que o uso das classes e da polimorfia não seja a melhor forma. Por isso foca em reduzir a dependência entre as classes.

→ Diante dos problemas de código não reutilizável, TEMOS:

- Rigidez → quando você muda uma parte do código e ele quebra

- Fragilidade → coisas quebram em lugares não relacionados

- Inutilidade → o código não pode ser reutilizado fora do contexto original

→ SOLID - princípios na tentativa de tornar o código mais flexível, fácil de manutenção e evolução.

→ 5 princípios SOLID são:

1) SRP - Single Responsibility Principle

Cada classe tem que ter 1 objetivo bem definido

2) OCP - Open/Close Principles

Deve-se poder estender o comportamento de uma classe sem modificá-la.

3) LSP - Liskov Substitution Principles

Deve haver a possibilidade de substituir qualquer instância da classe mãe com uma instância da classe filha.

4) IPS - Interface Segregation Principle

É melhor ter várias pequenas interfaces do que poucas e grandes

Exemplo:

Fa preferível usar composição no lugar de herança

5) DIP - Dependency Inversion Principle

O código deve depender de abstrações:

- * Dependency inversion + princípio de inversar dependências
- * Dependency injection + ato de inverter dependências
- * Constructer injection + fazendo dependency injection via construtor
- * Parameter injection + " " " " via parâmetros de um método

O objetivo é reduzir o máximo possível a criação de um objeto concreto, como no main