

16 de 09 de 2025

Cíthun da Silva

## The Solid Principles of object oriented design

O texto inicia desafiando a noção comum de que a programação orientada objeto que serve para modelar o mundo real. Em vez disso apresenta a visão de Robert Martin, de que o objetivo principal da POO é gerenciar dependências para evitar código rígido, frágil e não reutilizável.

O mau gerenciamento de dependências que leva problemas comuns de design de software que Martin identifica como rigidez.

Os princípios SOLID são o conjunto de cinco princípios de design introduzidos por Robert Martin.

Princípio da responsabilidade única, que estabelece que uma classe deve ter apenas um único motivo para mudar. Que seria uma classe CALCULAREAS que além de somar as áreas de várias formas também lida com a impressão dos resultados.

Princípio Aberto/Fechado, que dita que as entidades de software devem ser abertas para extensão mas fechadas para modificação, que seria uma classe que demonstra calcular a área dos retângulos, deixando clara a solução é usar classes abstratas.

ANOTAÇÕES

, de de

6.00 Princípio da substituição de letras, que afirma  
 7.00 que uma instância de uma classe pai deve poder ser  
 8.00 substituída por uma instância de uma das classes  
 filhas sem comprometer o funcionamento do programa.  
 9.00 Um exemplo tipo um: Square que herda de  
 Rectangle

10.00 Princípio da segregação de interfaces, defendem  
 11.00 que interfaces pequenas e específicas são bem melhor  
 12.00 do que poucas interfaces grandes e generosas

13.00 Por último o Princípio da Inversão de Dependência,  
 14.00 que estabelece que o código deve depender  
 15.00 de abstrações e não de abstrações concretas e  
 16.00 ilustram o seu princípio com três classes que são  
 17.00 cat que sua implementação de Makenoise, e a Mammal  
 18.00 generica que recebe um objeto de comportamento  
 19.00 abstrato  
 20.00

ANOTAÇÕES