

S T Q Q S S D

16 de 09 de 2025

Círculo da Saber

The Solid Principles of object oriented design

O texto inicia desafiando a noção comum de que a programação orientada objeto que servem para modular o mundo real. Esse não é seu aparente a visão de Robert Martin, o que é importante no princípio 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 identificou como rígidos.

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

Princípio da responsabilidade única, que estabelece que uma classe deve ter apenas um único motivo para mudar. Que é com uma classe CALCULADORA que além de somar os outros de várias formas também lidar com as impressões dos resultados.

Princípio Aberto/Fechado, que dizia que as entidades de software devem ser abertas para extensões mas fechadas para modificações, que é uma classe que demonstra colunas e área das retângulos, deixando clara a solução é usar classes abstratas.

ANOTAÇÕES

, de _____ de _____

6.00 Princípio da substituição de bicho, que afirma que uma instância de uma classe por outra pode ser substituída por uma instância de uma das classes filhas sem comprometer o funcionamento do programa.
7.00 Existe exemplo tipo um: Só que aqui herda de Rectangle
8.00

9.00 Princípio da segregação de interface, defendem que interfaces pequenas e específicas são bem melhores do que poucas interfaces grandes e genéricas
10.00

11.00 O último é o Princípio da inversão de dependências, que estabelece que o código deve depender de abstrações e não de abstrações concretas e ilustram a esse princípio com três classes que só cat que sua implementação de Mammal, e a Mammal gerencia que recebe um objeto de comportamento abstrato

14.00

15.00

16.00

17.00

18.00

19.00

20.00

ANOTAÇÕES