

OO é sobre manejar dependências invertendo dependências para evitar um código rígido, frágil e não reutilizável.

is-a has-a

Rigidez ~ uma parte quebra a outra

fragilidade ~ quebra tudo

imobilidade ~ só serve pra uma coisa

SRP ~ (Single responsibility principle) - Cada classe só deve ter uma função

OCP (Open/Closed Principle) ~ Aberto para extensão, fechado para modificação

LSP (Liskov substitution Principle) - Subtipos devem ser substituídos pelos substituíveis por seus tipos base

ISP (Interface Segregation Principle) - Uma classe não deve ser forçada a depender de interfaces que não usa (muitas mini interfaces > 1 grande)

DIP (Dependency inversion Principle) - Alto nível não deve depender de baixo nível, ambos devem depender de abstrações