

Single Responsibility Principle (SRP)

Open-Closed Principle (OCP)

Liskov Substitution Principle (LSP)

Interface Segregation Principle (ISP)

Dependency Inversion Principle

★

① ~~S~~ (represent) quer dizer que cada classe/componente deve ter apenas uma responsabilidade. Ou seja, cada componente deve ter uma única responsabilidade.

② ~~O~~ significa que objetos devem poder ter a capacidade de se estender, mas não serem modificados. Ou seja, quando formos implementar uma nova classe por exemplo, toda a lógica referente a aquela classe, deve estar dentro dela, caso ~~isso~~ for necessário aplicar um método parecido em classes iguais, podemos usar uma classe abstrata.



O L diz que uma subclasse deve poder ser substituída pelo seu classe pai. Isso faz com que possamos chamar métodos de várias subclasses diferentes de uma vez com uma coleção. Mas não podemos violá-lo, que seria sobrescrever um método sem um ~~ou~~ lançando uma *Exception* inesperada e retornar valores diferentes do classe base.

O I (P) diz que uma classe não deve ser forçada a implementar métodos que não irá utilizar, no caso seria melhor criar uma interface mais específica, mesmo que seja uma (~~uma~~) interface de uma interface.

O D fala que devemos depender de abstrações e não de implementações, por exemplo para conectarmos a um banco de dados, não poder acoplar diretamente as instâncias no método que irá fazer o conexão, mas ~~sim~~ abstrairmos ele como uma dependência/parâmetro, para que assim possamos conectar vários tipos de bancos.