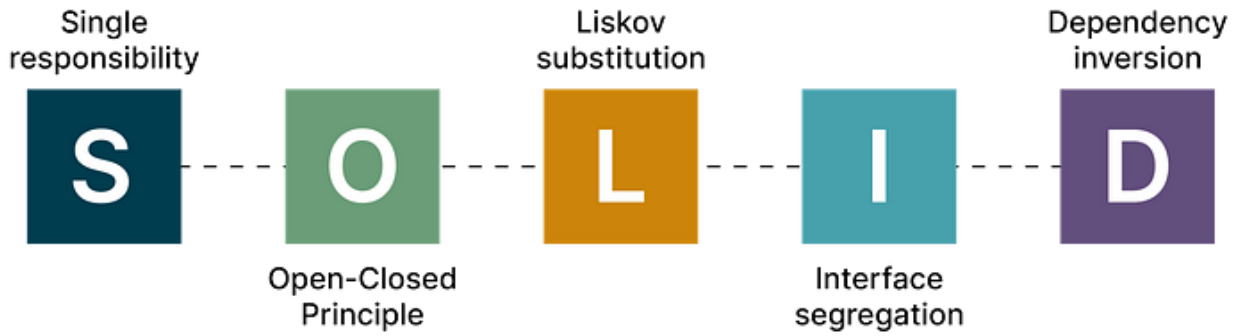
Applying SOLID Principles to Spring Boot Applications

https://medium.com/@saygiligozde/applying-solid-principles-to-spring-boot-applications-191d7e50e1b3



1. **Single Responsibility Principle (SRP)**

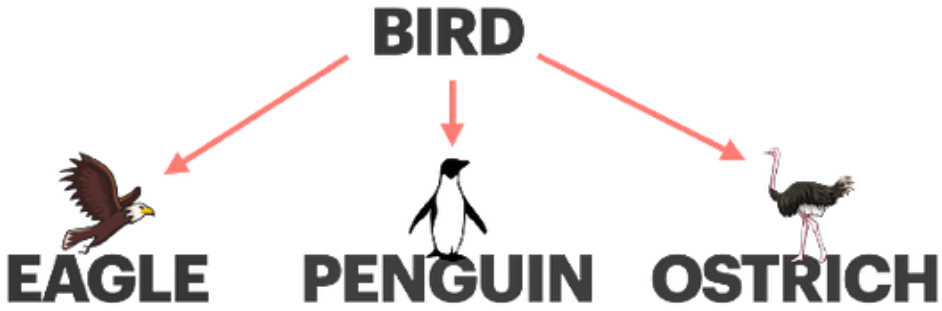
A class should have one, and only one, reason to change.

Open/Closed Principle (OCP)

class should be open for extension and closed to modification. This helps avoid introducing bugs to a working application. In simpler terms, this means that **you should be able to add new functionality to a class without changing its existing code**.

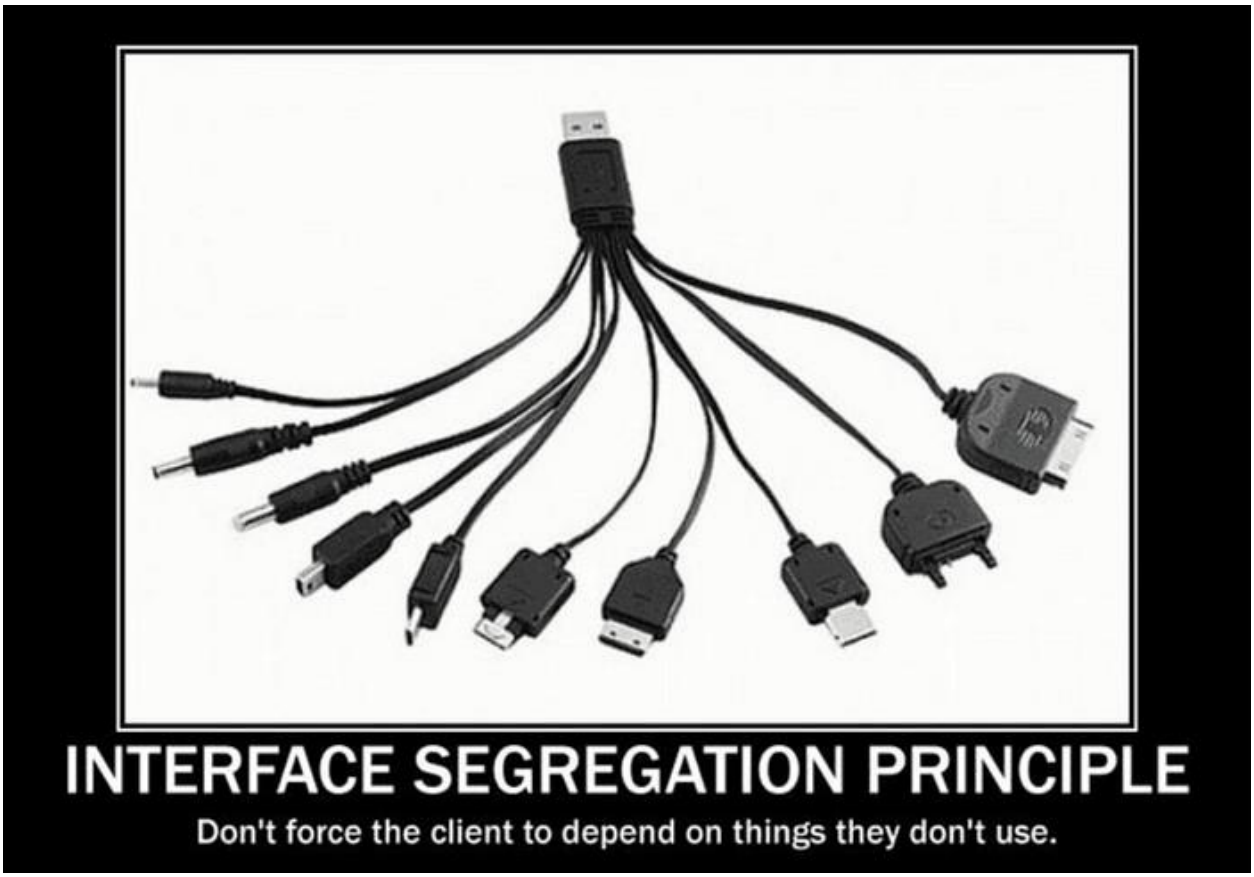
2. Liskov's Substitution Principle (LSP)

The Liskov Substitution Principle states that if you have a class, you **should be able to replace it with a subclass without causing any problems in your program**.



4. Interface Segregation Principle (ISP)

Interface Segregation Principle states that **larger interfaces should be split into smaller ones.**



Dependency Inversion Principle (DIP)

The Dependency Inversion Principle (DIP) states that **high level modules should not depend on low level modules**; both should depend on abstractions. Abstractions should not depend on details.