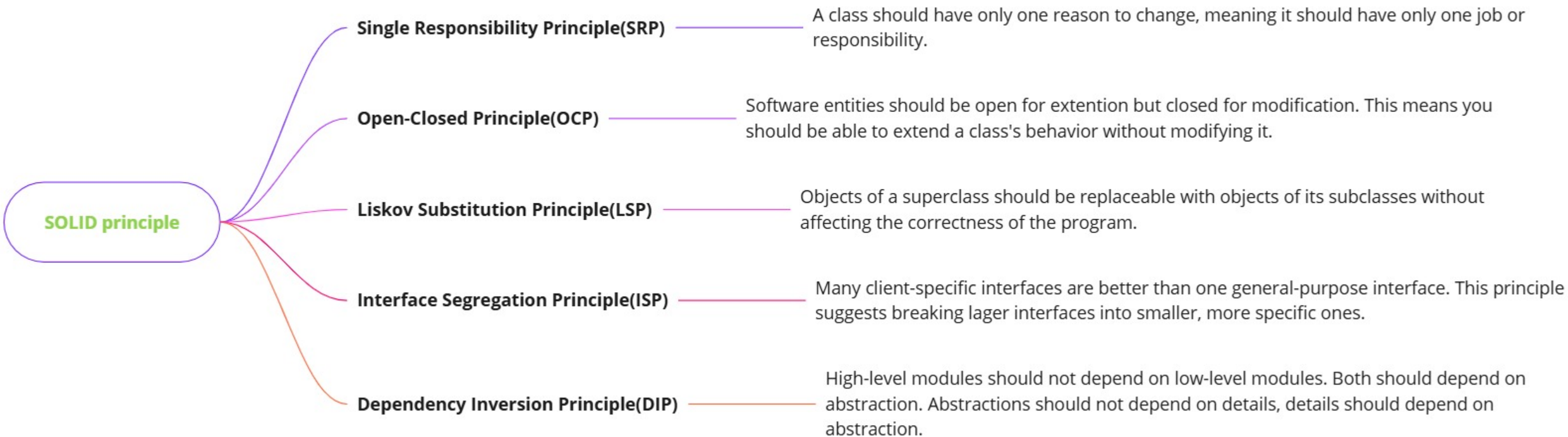
SOLID principle
Single Responsibility Principle(SRP)
A class should have only one reason to change, meaning it should have only one job or responsibility.
Open-Closed Principle(OCP)
Software entities should be open for extention but closed for modification. This means you should be able to extend a class's behavior without modifying it.
Liskov Substitution Principle(LSP)
Objects of a superclass should be replaceable with objects of its subclasses without affecting the correctness of the program.
Interface Segregation Principle(ISP)
Many client-specific interfaces are better than one general-purpose interface. This principle suggests breaking lager interfaces into smaller, more specific ones.
Dependency Inversion Principle(DIP)
High-level modules should not depend on low-level modules. Both should depend on abstraction. Abstractions should not depend on details, details should depend on abstraction.