

5 solid principles

first Single-responsibility

A class should have one and only one reason to change, meaning that a class should have only one job

second Open-Closed Principle

Objects or entities should be open for extension but closed for modification ,class should be extendable without modifying the class itself

third Liskov Substitution Principle

every subclass or derived class should be substitutable for their base or parent class

fourth Interface segregation principle

A client should never be forced to implement an interface that it doesn't use, or clients shouldn't be forced to depend on methods they do not use

fifth Dependency inversion principle states

High-level modules should not depend on low-level modules, both should depend on abstractions. Abstractions should not depend on details, details should depend on abstractions.