S – Single Responsibility Principle

O – Open Close Principle (Open for extension and closed for modification)

L – Liskov Substitution Principle

I – Interface Seggregation Principle

D – Dependency Injection Principle

Benefits

* Avoid Code duplication
* Easy to maintain
* Easy to understand
* Flexible
* Reduce Complexity

SRP – A class should have only 1 reason to change. A class should have only 1 responsibility.

OCP – A class should be open for extension but close for modification.

LSP - It states that objects of a superclass should be replaceable with objects of a subclass without affecting the correctness of the program.

ISP- It states that **clients should not be forced to depend on interfaces they do not use**. In other words, an interface should be designed in such a way that it only includes the methods that are relevant to the implementing class. Clients should not be forced to implement methods they don't need or use.

DIP – Class should depend on interfaces rather than on concrete classes.

Iterator Pattern – Cursor , Enumerator

* Access items in a collection one by one

Foreach in c# uses enumerator internally