

SOLID:

•S - Принцип единственной ответственности (Single Responsibility Principle):

Принцип единой ответственности гласит, что у каждого класса должна быть только одна «ответственность» и он не должен брать на себя другие обязанности. Роберт К. Мартин объяснял его так: «У класса должна быть лишь одна причина для изменения».

•O - Принцип открытости/закрытости (Open-Closed Principle):

«Сущности программы (классы, модули, функции и т.п.) должны быть открыты для расширения, но закрыты для изменений»

Добавление любых дополнительных функций может быть реализовано путем создания новых сущностей, которые расширяют возможности существующего класса и добавляют дополнительные функции самим себе. Таким образом можно предотвратить частые и тривиальные изменения в хорошо зарекомендовавшем себя классе низкого уровня.

•L - Принцип подстановки Барбары Лисков (Liskov Substitution Principle):

Принцип подстановки Лисков гласит: «Объекты в программе должны быть заменяемы экземплярами их подтипов без ущерба корректности работы программы».

•I - Принцип разделения интерфейсов (Interface Segregation Principle):

Принцип разделения интерфейсов гласит, что «Ни один клиент не должен зависеть от методов, которые он не использует».

Принцип разделения интерфейсов предполагает создание небольших интерфейсов, известных как «ролевые интерфейсы», вместо большого интерфейса, состоящего из нескольких методов. Разделяя методы по ролям на более мелкие интерфейсы, клиенты будут зависеть только от методов, которые имеют к ним отношение.

•D - Принцип инверсии зависимостей (Dependency Inversion Principle):

1. Модуль высокого уровня не должен зависеть от модулей низкого уровня. И то, и другое должно зависеть от абстракций.

2. Абстракции не должны зависеть от деталей реализации. Детали реализации должны зависеть от абстракций. Если ваш код уже реализует принципы открытости/закрытости и подстановки Лисков, он уже будет неявно согласован с принципом инверсии зависимостей.