

SOLID Principles Handbook

Introduction to SOLID

SOLID principles are fundamental to writing good object-oriented code. These principles help create more maintainable, flexible, and scalable software.

Single Responsibility Principle

A class should have only one reason to change. This principle states that every class should have a single responsibility or job.

Open/Closed Principle

Software entities should be open for extension but closed for modification. This means you should be able to add new functionality without changing existing code.

Liskov Substitution Principle

Objects of a superclass should be replaceable with objects of its subclasses without affecting the correctness of the program.

Interface Segregation Principle

Clients should not be forced to depend on interfaces they do not use. Break down large interfaces into smaller and more specific ones.

Dependency Inversion Principle

SOLID Principles Handbook

High-level modules should not depend on low-level modules. Both should depend on abstractions.