

10 Object Oriented Design Principles

What are the Object Oriented Principles?

Liskov Substitution Principle

Subtypes must be substitutable for their base types. Derived classes must be usable through the base class interface without the need for the user to know the difference

Open Closed Principle

Classes should be open for extension but closed for modification

Single Responsibility Principle

A class should have only one reason for change

Dependency Inversion Principle

Abstractions should not depend on details. Details should depend on abstractions

Common Reuse Principle

The classes in a package are reused together. If you reuse one of the classes in a package, you reuse them all. Classes that aren't reused together should not be grouped together.

Release-Reuse Equivalency Principle

The granule of reuse is the same as the granule of release. Only components that are released through a tracking system can be effectively reused

Interface Segregation Principle

Interfaces belong to clients, not to hierarchies as such client specific interfaces are better than one general purpose interface

Stable Abstraction Principle

The more stable a class category is, the more it must consist of abstract classes. A completely stable category should consist of nothing but abstract classes.

Stable Dependency Principle

Dependencies between released categories must run in the direction of stability. The dependee must be more stable than the dependor

Common Closure Principle

The classes in a package should be closed together against the same kinds of changes. A change that affects a closed package affects all the classes in that package and no other packages. Classes that change together, belong together.

Acyclic Dependencies Principle

The dependency structure for released components must be a directed acyclic graph. Allow no cycles in the dependency graph.