Some Principles of Object Oriented Design

The Good, the Bad and the Ugly

Bad Design

- Rigide
- Fragile
- Immobile

Good Design

- Flexible
- Robust
- Reusable

Six Principles of Class Design

Single Responsibility

A class should have one, and only one, reason to change.

Open Close

You should be able to extend a classes behavior, without modifying it.

Liskov Substitution

Derived classes must be substitutable for their base classes.

Dependency Inversion

Depend on abstractions, not on concretions.

Interface Segregation

Make fine grained interfaces that are client specific.

Single Level of Abstraction

A method lives at a single level of abstraction.

Six Principles of Package Design

Release/Reuse Equivalence

Only components that are released through a tracking system can be effectively reused.

Common Reuse

The classes in a package are reused together. If you reuse one of the classes in a package, you reuse them all.

Common Closure

The classes in a package should be closed together against the same kinds of changes. A change that affects a package affects all the classes in it.

Acyclic Dependencies

There must be no cycles in the dependency structure between packages.

Stable Dependencies

A package should only depend upon packages that are more stable that it is.

Stable Abstractions

The abstraction of a package should be in proportion to its stability.

To Summarize

Two Key Ideas

- Low Coupling
- High Consistency

Two Key Mechanisms

- Abstraction
- Polymorphism

That's all folks!