Single responsibility principle

Open/close principle

Liskov substitution principle

Interface segregation principle

Dependency inversion principle

Single responsibility principle

- Every class is responsible for a single part of the system.
- A class´s encapsulation by the class.
- A class´s narrowed alignment.

Open/close principle

- Class open if it is still available for extension.
- Class close for use by other classes.

Liskov substitution principle

- If S is a specialization of T, then an S object can be used wherever a t object is required.

Interface segregation principle

- No client is forced to depend on methods that it does not use.
- The public methods of a component can be grouped by purpose or responsibility as captured in a declaration in interfaces, or abstract classes.

Dependency inversion principle

- Organize the system, into layers.
- Components from the abstract layers should not depend on components from detail layers.
- Abstraction does not depend on details.
- Implementation details depend on abstractions.