1. 开闭原则，对扩展开放，对修改关闭，典型的比如抽象类和接口
2. 里氏替换原则：子类必须可以实现父类所有功能，比如企鹅不能继承鸟类，因为他不会飞，这个原则一定要遵守，否则会出现很多bug
3. 依赖倒置原则: 高层模块不应该依赖低层模块，两者都应该依赖其抽象；抽象不应该依赖细节，细节应该依赖抽象, 其核心思想是：要面向接口编程，不要面向实现编程。

依赖倒置原则的主要作用如下:

* 1. 每个类尽量提供接口或抽象类，或者两者都具备。
  2. 变量的**声明**类型尽量是**接口或者是抽象类**。
  3. 任何类都不应该从具体类派生。
  4. 使用继承时尽量遵循里氏替换原则。

1. 单一职责原则的核心就是控制类的粒度大小、将对象解耦、提高其内聚性。如果遵循单一职责原则将有以下优点:
   1. 降低类的复杂度。一个类只负责一项职责，其逻辑肯定要比负责多项职责简单得多。
   2. 提高类的可读性。复杂性降低，自然其可读性会提高。
   3. 提高系统的可维护性。可读性提高，那自然更容易维护了。
   4. 变更引起的风险降低。变更是必然的，如果单一职责原则遵守得好，当修改一个功能时，可以显著降低对其他功能的影响。

## 合成复用原则：当实现新功能时，不继承之前的已有类，而是新建类，然后把之前的类的对象放入其中，组合

1. 建造者模式，感觉可以用，<http://c.biancheng.net/view/1354.html>

