**设计模式笔记**

**单一职责原则：Single Responsibility Principle**

一个类里面应该是一组相关性很高的函数、数据的封装，完全不一样的功能不应该放在一个类中。我们可以根据具体的业务、功能对类进行相应的拆分，优化我们的代码，提高灵活性和可扩展性、降低耦合性。

**开闭原则：Open Close Principle**

软件中的对象（类、模块、函数等）应该对于扩展是开放的，对于修改是封闭的。

当软件需要变化时，我们应该尽量通过扩展的方式来实现变化，而不是修改已有的代码来实现，这样可以避免把错误引入已经经过测试的旧代码中，破坏原有的系统。不过实际上，修改原有代码与扩展代码往往是同时存在的。

遵循开闭原则最重要的手段是抽象。

里氏替换原则：Liskov Substitution Principle

所有引用基类的地方必须能透明地使用其子类的对象。换句话说，只要父类能出现的地方，子类就可以出现，把父类替换成子类也不会产生任何错误或异常。