# S.O.L.I.D原则

S – Single responsibility principle

单一责任原则：一个类应该有且只有一个去改变他的理由，这意味着一个类应该只有一项工作。

O – Open close principle

开放封闭原则：对象或尸体应该对扩展开放，对修改封闭。

L – Liskov substitution principle

里氏替换原则：任何一个子类或派生类应该可以替换它们的基类或父类。

I – Interface segregation principle

接口分离原则：不应该强迫一个类实现它用不上的接口。

D – Dependency inversion principle

依赖反转原则：抽象不应该依赖于具体实现，具体实现应该依赖于抽象。

# 5C解题法

Clarif ：通过和面试官交流，去除题目中的歧义，确定答题范围。

Core objects ：确定题目所涉及的类，以及类之间的映射关系。

Cases ：确定题目中所需要实现的场景和功能。

Classes ：通过类图的方式，具体填充题目中涉及的类。

Correctness ：检查自己的设计，是否满足关键点。

# 设计模式

## 策略模式





