1.模拟鸭子项目
2.项目的新需求
3.用OO原则解决新需求的不足
4.用策略模式来新需求解决
5.重新设计模拟鸭子项目
6.总结策略模式定义

传统抽象类和子类去解决问题的弊端：

继承的问题：对类的局部改动，尤其是超类的局部改动，会影响其他部分。影响会有溢出效应。

超类中挖的一个坑，每个子类都要来填，增加工作量，复杂度O（N^2），不是好的设计方式。

需要新的设计方式，应对项目的扩展性，降低复杂性：

（1）分析项目变化与不变部分，提取变化部分，抽象成接口+实现 这样新增行为很简单，行为类更好的复用，组合更方便，没有挖坑

（2）鸭子的哪些功能时会根据新需求变化的？叫声、飞行...

这样就可以在鸭子的抽象类里去增加变化部分作为一个属性，在fly和quack方法中去调用其中的方法。而在子类中可以通过构造函数中·
去初始化变得那些接口。

策略模式：分别封装行为接口，实现算法族，超类里放行为接口对象，在子类里具体设定行为对象。原则就是：分离变化接口，封装接口，基于接口编程各种功能。从模式让行为算法的变化独立于算法的使用者。

注意点：
1.分析项目中变化部分和不变部分
2.多用组合少用继承；用行为类组合，而不是行为的继承。更有弹性