

# 设计模式原则 - 接口隔离原则

---

寂然

## 官方定义

接口隔离原则 (Interface Segregation Principle) , 又称为ISP原则, 官方定义为:

1. Clients should not be forced to depend upon interfaces that they don't use.  
(客户端不应该依赖它不需要的接口)
2. The dependency of one class to another one should depend on the smallest possible interface. (类间的依赖关系应该建立在最小的接口上)

## 基本介绍

通俗的来讲, 不要在一个接口里面定义过多的方法, 接口应该尽量细化

## 案例演示 - 市场风云

假设有这样一个案例场景, 现在有一个接口knife, 给定他有三个能力, 可以切苹果, 切番茄, 切土豆, 两个类张

厨师, 李厨师分别具有这些能力, 有一个水果店类, 假设是需要张师傅来切苹果和切番茄, 而另一个蔬菜店类需要

李师傅来切番茄和切土豆

```
// 定义接口knife
interface Knife {
    //切苹果的能力
    void cutApple();

    //切番茄的能力
    void cutTomato();

    //切土豆的能力
    void cutPotato();
}
```

```

}

//张厨师类
class CookZhang implements Knife {

    @Override
    public void cutApple() {
        System.out.println("张厨师正在切苹果");
    }

    @Override
    public void cutTomato() {
        System.out.println("张厨师正在切番茄");
    }

    @Override
    public void cutPotato() {
        System.out.println("张厨师正在切土豆");
    }
}

//李厨师类
class CookLi implements Knife {

    @Override
    public void cutApple() {
        System.out.println("李厨师正在切苹果");
    }

    @Override
    public void cutTomato() {
        System.out.println("李厨师正在切番茄");
    }

    @Override
    public void cutPotato() {
        System.out.println("李厨师正在切土豆");
    }
}

```

## 案例分析

？出现了问题，上面的写法违反了接口隔离原则

## 解决方案

## 案例总结

接口隔离原则就是当我一个类通过接口依赖（使用）另一个类的时候，要保证依赖的该接口是最小的，接口里面有

方法用不到的，就进行隔离，而隔离的做法就是，就对原来接口进行拆分，拆分为最小粒度，来避免耦合

## 与单一职责原则对比

单一职责原则：合理的职责分解，一个类只负责一项职责

接口隔离原则：类间的依赖关系应该建立在最小的接口上

### 相同点

都要求对结构进行拆分，都要求更小的粒度，都希望减少耦合

### 不同点

#### 审视角度的不同

单一职责原则：类与接口职责单一，**注重的是职责**

接口隔离原则：要求我们尽量使用多个专门的接口，**注重的是接口的设计**

当然了，我们使用接口隔离原则进行接口拆分的时候，要遵循单一职责原则

## 下节预告

### 依赖倒置原则

