

## 简述

接口隔离原则是值得，用多个单一的接口，而不是使用一个多功能个的总接口。

- 一个类对另外一个接口的依赖应该建立在最小的接口上，不用引入不需要的接口
- 建立接口，不要建立臃肿庞大的接口
- 尽量细化接口，接口中的方法在保证适度的情况下，尽可能的少一些

## 举例说明

```
1 public interface IAnimal {
2     void fly();
3     void eat();
4     void swim();
5 }
6
7 public class Dog implements IAnimal {
8     public void fly() {}
9     public void eat() {}
10    public void swim() {}
11 }
12
13 public class Bird implements IAnimal {
14     public void fly() {}
15     public void eat() {}
16     public void swim() {}
17 }
```

上面代码中 定义了一个动物接口，有三个方法。 但是Dog很明显不会飞， Bird 不会swim，那么就会有方法空着。根据接口隔离原则应该

```
1 public interface IEat {
2     void eat();
3 }
4
5 public interface IFly {
6     void fly();
7 }
8
9 public interface ISwim {
10    void swim();
11 }
12
13 public class Bird implements IFly, IEat {
14     public void fly() {}
15     public void eat() {}
16 }
17
18 public class Dog implements IEat, ISwim {
19     public void eat() {}
20     public void swim() {}
21 }
```

