

定义

里氏代换原则是对“开-闭”原则的补充，可以简单的理解为一个软件实体如果使用的是一个父类，那么一定适用于其子类，而且它察觉不出父类件里面，把父类都替换成它的子类，程序的行为没有变化

示例说明

定义一个长方形

```
1 @Setter
2 @Getter
3 @ToString
4 public class Rectangle {
5     private long length;
6     private long width;
7 }
```

定义一个正方形继承长方形，正方形其实是属于一个特殊的长方形

```
1 @Setter
2 @Getter
3 @ToString
4 public class Square extends Rectangle{
5     private long len;
6
7     @Override
8     public long getLength() {
9         return this.len;
10    }
11
12    @Override
13    public long getWidth() {
14        return this.len;
15    }
16
17    @Override
18    public void setLength(long length) {
19        super.setLength(length);
20    }
21
22    @Override
23    public void setWidth(long width) {
24        super.setWidth(width);
25    }
26 }
```

测试类，当长方形的宽度小于长度的时候让宽度 + 1，直到 长方形变成正方形

```
1 public class Client {
2
3     public static void main(String[] args) {
4         Rectangle rectangle = new Rectangle();
```

```

5     rectangle.setWidth(10);
6     rectangle.setLength(20);
7     resize(rectangle);
8 }
9
10    public static void resize(Rectangle rectangle){
11        while (rectangle.getWidth() <= rectangle.getLength()){
12            rectangle.setWidth(rectangle.getWidth() + 1);
13            System.out.println(rectangle.toString());
14        }
15    }
16 }
17
18
19 输出
20 Rectangle(length=20, width=11)
21 Rectangle(length=20, width=12)
22 Rectangle(length=20, width=13)
23 Rectangle(length=20, width=14)
24 Rectangle(length=20, width=15)
25 Rectangle(length=20, width=16)
26 Rectangle(length=20, width=17)
27 Rectangle(length=20, width=18)
28 Rectangle(length=20, width=19)
29 Rectangle(length=20, width=20)

```

如果将上面测试代码改成这样就会造成死循环，这样就违背了里氏替换原则

```

1 public static void main(String[] args) {
2     Square square = new Square();
3     square.setLen(10);
4     resize(square);
5 }

```