里氏代换原则的内容只有一句话：子类的对象能够替换其基类的对象被使用。举个例子，任何使用Animal的地方，我们都可以放Cat进去，而完全不扰乱程序的逻辑。而且程序里面关于Animal的假设，Cat都不能打破。

这到底是什么意思呢？譬如说，凡是动物都会吃东西，这个没问题，猫也会吃。但是凡是动物都会走路，这个就不对了，因为鱼是动物，但是鱼不会走路。面对这个情况我们就有两种可能：

第一种是鱼根本就不应该继承自动物，那么我们就不要继承。

第二种是凡是动物都会走路的这个说法是不对的，所以不应该往动物类里添加走路的这个函数。

当然根据常识，我们有“一条鱼∈{动物}”，因此应该做的是在动物里去掉走路的这个函数。

另一个流行的错误是，因为正方形是长方形的一种，所以正方形应该继承自长方形。这当然也是不对的。因为对于任何长方形，我们都可以通过调用长方形的函数来修改长和宽，那我们给了不同的数字怎么办？有些人会说，那你不要使用这些函数，我们给正方形添加一个新的修改函数不行吗？当然不可以，因为这违反了里氏代换原则。如果你想让正方形继承自长方形，那么对正方形使用所有长方形的函数，都不应该出问题。

但是大家会想，这个做法其实是满足“一个正方形∈{长方形}”的说法，那问题到底出在哪里呢？这其实是因为，我们用数学语言去表达集合的时候，我们通常都认为这些东西是不可变的，譬如说一个长方形创建出来，他永远就是那个形状了。所以只读的正方形继承自只读的长方形，这是没有错的：

那么可以修改的长方形和正方形怎么办呢？正确的做法是，让他们分别继承自只读的长方形和正方形。所以可以修改的长方形和可以修改的正方形之间，其实没有任何继承关系