

设计模式六大原则——里氏替换原则（LSP）

2014-05-25 21:35 1386人阅读 评论(11) 收藏^[1] 举报

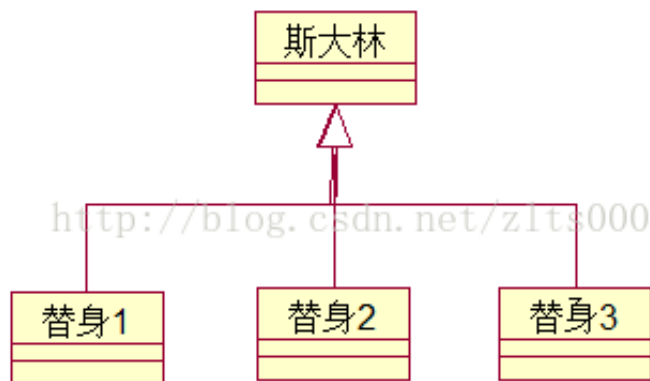
版权声明：本文为博主原创文章，未经博主允许不得转载。

里氏替换原则（LSP，Liskov Substitution Principle）是关于继承机制的原则，是实现开放封闭原则的具体规范，违反了里氏替换原则必然违反了开放封闭原则。



约瑟夫.斯大林，苏联时期苏联共产党的最高领导人，对于斯大林有没有替身？有几个替身？有一种说法：斯大林有好几个替身，最著名的当属“第一替身”叶夫谢伊.卢比茨基——他“扮演”领袖斯大林长达15年之久，很多高官都被蒙在鼓里。

斯大林作为一国的领导人，肯定要不停的参加各种的聚会，进行各中慰问。。。找几个替身，忙里偷闲或者外出打掩护.....



★子类必须完全实现父类的方法

替身代替斯大林去参加各种活动时，他的相貌和一举一动都必须和真的斯大林一模一样，毕竟斯大林代表的是一个国家，如果让别人发现来见自己的是个替身，那不就让人寒心了嘛... 看一下代码：

```
1. //斯大林
2. public class SiDaLin
3.     //斯大林讲话行为
4.     public virtual SpeakWay()
5.         Console.WriteLine("右手举过头顶、伸出两个手指、不断前后摆动，以表示强调或愤慨。")
6. public class TiShen : SiDaLin
7.     //替身的讲话行为
8.     public override SpeakWay()
9.         Console.WriteLine("右手举过头顶、伸出两个手指、不断前后摆动，以表示强调或愤慨。")
```

```
//斯大林
public class SiDaLin
{
    //斯大林讲话行为
    public virtual void SpeakWay()
    {
        Console.WriteLine("右手举过头顶、伸出两个手指、不断前后摆动，以表示强调或愤慨。")
    }
}

//替身
public class TiShen : SiDaLin
{
    //替身的讲话行为
    public override void SpeakWay()
    {
        Console.WriteLine("右手举过头顶、伸出两个手指、不断前后摆动，以表示强调或愤慨。")
    }
}
```

不知道大家有没有注意到上面代码中父类中的virtual和子类中的override,为什么要用到它们？

在继承关系中，子类对父类的继承除了字段、属性，还有方法，而使同一方法在子类中表现出不同的行为是通过多态表现的，具体在语言上的操作上表现为父类提供虚函数，而在子类中覆写该虚函数，这是抽象机制的重要基础。

1. `public static DoSomething(Fatherclass f)`
2. `f.Method();`

```
public static void DoSomething(Fatherclass f)
{
    f.Method();
}
```

如果Method被实现为虚函数，并且在子类中被覆写，那么传入DoSomething中的实参既可以是父类，也可以是子类，子类完全可以替代父类在此调用自己的方法，这正是里氏替换原则强调的继承关系，代码如下：

1. `class Fatherclass`
2. `//父类行为`
3. `public virtual Method()`
4. `class Sonclass :Fatherclass`
5. `//子类重写父类方法`
6. `public override Method()`

```
class Fatherclass
{
    //父类行为
    public virtual void Method()
    {
    }
}
class Sonclass :Fatherclass
{
    //子类重写父类方法
    public override void Method()
    {
    }
}
```

（上面这个问题的回答参考《你必须知道的.NET》）

★子类可以有自己的“个性”

采用里氏替换原则时，尽量避免子类的“个性”，一旦子类有“个性”，这个子类和父类之间的关系就很难调和了，把子类当做父类使用，子类的“个性”被抹杀——委屈了点；把子类单独作为一个业务来使用，则会让代码间的耦合关系变得扑朔迷离——缺乏类替换的标准。

就拿斯大林的替身来说，每一个替身都是一个人，都有自己的“个性”，但是他们是斯大林的替身，那么他们平时的表现就必须按照斯大林的表现来，如果他们加入自己的“个性”，那么他们替身的身份就会漏出破绽。

实现开闭原则的关键步骤是抽象化，父类与子类之间的继承关系就是抽象化的体现，因此里氏替换原则是实现开闭原则的具体步骤规范，违反里氏替换原则一定违反开闭原则，反之未必。

Links

1. `javascript:void(0);`