

# <大话设计模式>

## 本教程说明及版权声明

- 该文档参考和使用了网络上的免费开放图片和内容，并以免费开放的方式发布, 希望为移动互联网和智能手机时代贡献绵薄之力！可以随意转载，但不得使用该文档谋利。
- 如果对该文档有任何疑问或者建议，请进入官方微博客  
<http://www.cnblogs.com/guoshiandroid/>留言或者直接与国土工作室联系（后附联系方式），我们会慎重参考您的建议并根据需要对本文档进行修改，以造福更多开发者！
- 《大话设计模式》的最新及完整内容会在国土工作室官方微博客定期更新，请访问国土工作室博客  
<http://www.cnblogs.com/guoshiandroid/>获取更多更新内容。

## 针对接口编程-问世间情为何物 直教人生死相许

应用场景举例：



“十六年后 在此重会；夫妻情深 勿失信约”，悲痛欲绝的杨过跑到断肠崖，看到小龙女亲手留在石壁上的文字，即惊喜又痛苦不欲生：“十六年！为什么要等到十六年？！”。

但是信约已定，痴情的杨过也只能等十六年了。

离开断肠崖后，杨过一边开始了自己的苦苦的等待与思恋，一边寄情练功，当然开始时候也忘不了吃那疗伤的草药。后来杨过巧遇了千年神雕，和神雕一见如故，从此便开始修炼独孤求败的武功。无事可做，寄情练剑倒也不失为人生的一大快事。“相思无用,唯别而已。别期若有定,千般煎熬又何如?莫道黯然销魂,何处柳暗花明? !”，惊天地泣鬼神的黯然销魂掌就这样诞生了。时光飞逝，恍惚间快过了十六年。此时，杨过的神功已成，想象着十六年约期就将来临，心中想象着自己一生的挚爱，不免感慨和激动万分！在祭拜过求败他老人家之后，杨过和神雕一起开始去赴那场长达十六年之久的约会。令众生激动和艳羡。

再出江湖的杨过惩奸除恶、帮扶弱小，很快就侠名远播，被人尊称为“神雕侠”。自己心中想象着小龙女过往的一笑一颦，想象着她是怎么度过这十六年的，不禁催生了更加浓烈的相思和相见之情。

千呼万唤，终于，这一天来到！

断肠崖边，佳人芳踪迹未现，过儿万念俱灰，纵身跳下悬崖...

幸好悬崖下面是深渊，杨过并没有死，被水冲到了岸边的杨过苏醒过来后，看到了很多小蜜蜂，他一眼就认出了这是小龙女样的蜜蜂，莫非龙儿就在附近？最后在深潭水下，杨过找到了自己苦苦等待了十六年的挚爱。原来小龙女得知自己无药可救，也纵身跳下断肠崖，

十六年之期只不过是为了让杨过不要轻生。但是跳崖后的小龙女并没有死掉，接着就在谷底一个世外桃源的地方慢慢的疗伤，竟然完全康复了。真是有情人终成眷属，有情人终成名人。

定义：

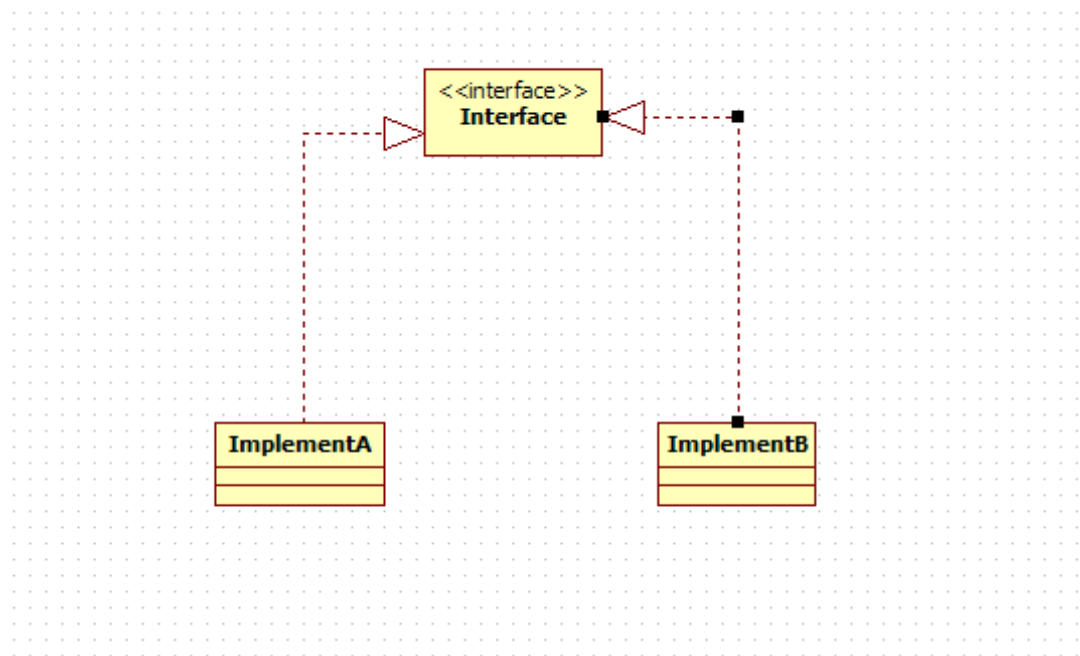
接口是一系列方法的声明，是一些方法特征的集合，一个接口只有方法的特征没有方法的实现，因此这些方法可以在不同的地方被不同的类实现，而这些实现可以具有不同的行为(功能)。

接口是对抽象的抽象。

接口就是标准，就是承诺。

针对接口编程，不要针对具体编程是依赖倒转原则的另外一种表述。

针对接口编程又称为面向接口编程，针对接口编程就是要先设计一系列的接口，把设计和实现分离开，使用时只需引用接口即可，也由于系统各部分的解耦合。如下图所示：

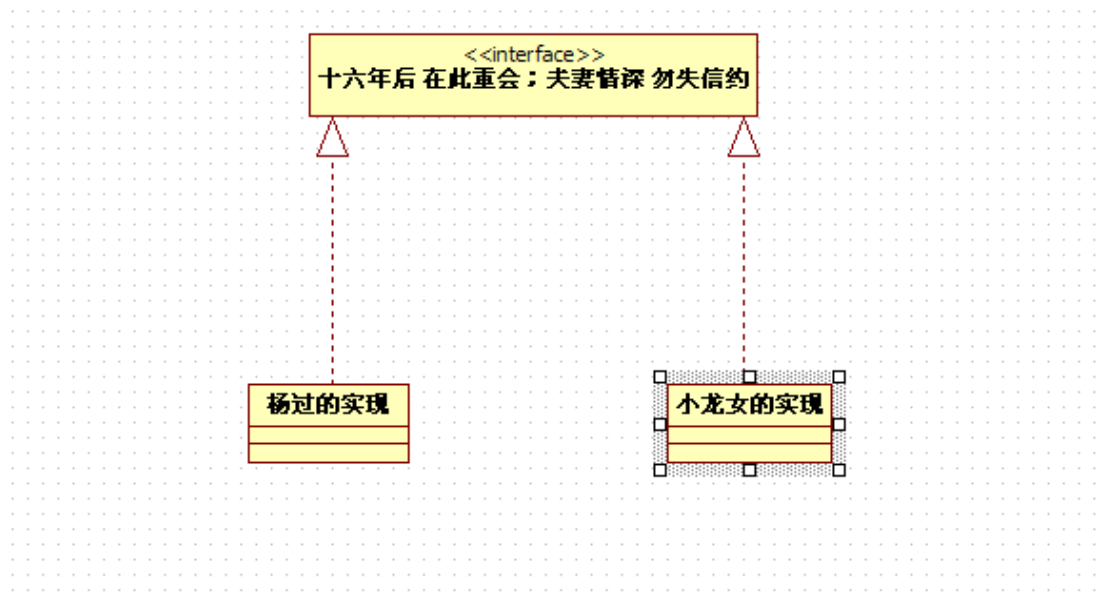


故事分析：

“十六年后 在此重会；夫妻情深 勿失信约”就是针对接口编程的一个绝妙的例子。而且最后还加了“信约”一次。言外之意就是说我们说好了要十六年在此地重逢，我们俩都要遵照此约定。根据上面的故事，小龙女和杨过制定好接口后，就纵身跳下了悬崖，不管自己的过儿了，因为她此时已经不用关心过儿怎么去再十六年后河自己相见，只要十六年后在此地相见即可，也就是说小龙女针对和使用都是接口，至于杨过怎么实现，她此时已经身患绝症而无法顾及了。而杨过看过此约定后，虽然无奈，但也只好照办。具体实现如下：回去吃断肠草调理自己，然后以神雕为伴练习武功，知道练成了黯然销魂掌而神功大成，然后就是在江湖上做侠义之事，然后就是按照信约与十六年后来到了断肠崖边；而小龙女就在谷底慢慢的调养，十六年后身体早已康复，而且越发迷人了。当杨过没有见到小龙女时纵身跳崖，遵照了“问世间情为何物，直教人生死相许”标准。

总结一下:小龙女和杨过定下接口,然后各自针对接口各自独立的做事,最终得以相见。

针对接口编程是未来提高程序的可维护性、可伸缩性和可复用性。如果你在一个类中直接使用另外的一个,这样就把两个类紧密的联系在了一起,以后如果想做出改变就很难了。如果针对接口编程,当业务变化时我们只需要用一个新的类实现接口即可,而客户端依旧可以使用接口引用新的类的,同时也保证了客户端的不变性。这样客户端和实现端互不影响,保持了各自的相对独立性。正如小龙女和杨过的,他们树立了十六年制约后,就不用关心彼此的如何去赴这场约定,只需要按照约定做事就 OK 了。互不影响,自由自在。如下图所示:



## Java 代码实现:

新建一个“信约”的接口,这个接口是杨过和小龙女都必须通过自己的方式实现的。代码如下:

```
package com.diermeng.designPattern.dating;
/*
 * 杨过和小龙女定下的约定接口
 */
public interface Dating {
    /*
     * 约定的接口
     */
    public void dating();
}
```

然后分别建立杨过和小龙女的实现类,分别实现上面的接口。代码依次如下:

```
package com.diermeng.designPattern.dating.impl;

import com.diermeng.designPattern.dating.Dating;
/*
```

```

    * 杨过对接口的实现
    */
    public class Yangguo implements Dating {
        /*
         * 姓名
         */
        String name;

        /*
         * 默认空构造方法
         */
        public Yangguo() {}

        /*
         * 传入name参数的构造方法
         */
        public Yangguo(String name) {
            this.name = name;
        }

        public String getName() {
            return name;
        }

        public void setName(String name) {
            this.name = name;
        }

        /*
         * (non-Javadoc)
         * @see com.diermeng.dating.inter.Dating#dating()
         * 杨过对约定的实现
         */
        public void dating() {
            if (this.getName() != null) {
                System.out.println(this.getName() + " : " + "十六年后 在此重会；夫妻情深 勿失信约");
            }
            else {
                System.out.println("十六年后 在此重会；夫妻情深 勿失信约");
            }
        }
    }
}

```

```
package com.diermeng.designPattern.dating.impl;

import com.diermeng.designPattern.dating.Dating;
/*
 * 小龙女对接口的实现
 */
public class XiaoLongnv implements Dating{
    /*
     * 姓名
     */
    String name;

    /*
     * 默认空构造方法
     */
    public XiaoLongnv() {}

    /*
     * 传入name参数的构造方法
     */
    public XiaoLongnv(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    /*
     * (non-Javadoc)
     * @see com.diermeng.dating.inter.Dating#dating()
     * 小龙女对约定的实现
     */
    public void dating() {
        if(this.getName() != null) {
            System.out.println(this.getName() + " : "+"十六年后 在此重会; 夫妻情深 勿失信约");
        }
        else{
            System.out.println("十六年后 在此重会; 夫妻情深 勿失信约");
        }
    }
}
```

```
    }  
}  
  
}
```

建立一个测试类，代码如下：

```
package com.diermeng.designPattern.dating.client;  
import com.diermeng.designPattern.dating.Dating;  
import com.diermeng.designPattern.dating.impl.XiaoLongnv;  
import com.diermeng.designPattern.dating.impl.Yangguo;  
/*  
 * 对杨过和小龙女约定进行测试的客户端  
 */  
public class DatingTest {  
  
    public static void main(String[] args)  
    {  
        //分别实例化实例化  
        Dating yangguo = new Yangguo("过儿");  
        Dating xiaoLongnv = new XiaoLongnv("龙儿");  
  
        //调用各自的方法  
        yangguo.dating();  
        xiaoLongnv.dating();  
    }  
}
```

程序运行结果如下：

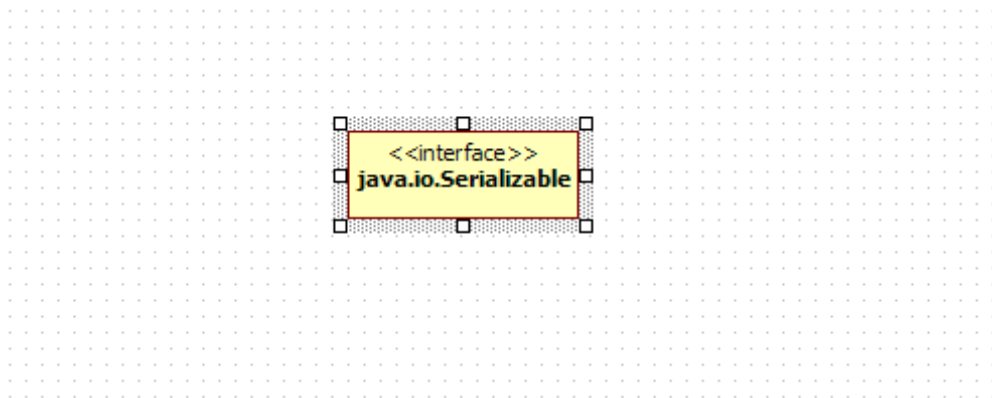
```
过儿 ： 十六年后 在此重会；夫妻情深 勿失信约  
龙儿 ： 十六年后 在此重会；夫妻情深 勿失信约
```

## 已有应用简介：

Java 是相面向对象编程的语言，而面向对象编程的核心之一就是要针对针对接口编程、不要针对实现编程，在 Java API 中的标志接口 `java.io.Serializable` 和 `java.rmi.Remote` 等就是我们经常遇到的，下面以 `java.io.Serializable` 为例说明一下，源代码如下：

```
package java.io;  
public interface Serializable{}
```

UML 图形如下图所示：



当然在 J2EE 框架的使用中到处都是针对接口编程的身影。例如在 [www.babasport.com](http://www.babasport.com) 中几乎每一处都是针对接口编程的，令人印象非常深刻的一点就是巴巴运动网把对数据库的 CRUD 等基本操作封装在了一个统一接口中，这给以后的代码的编写和数据库的操作带来了极大的方便，当然这里也使用了 Java 5 的泛形技术。有兴趣的读者可以去学习巴巴运动网的源代码。

### 温馨提示：

许下的承诺就是欠下的债。所以不要轻易做出承诺。

杨过和小龙女为了承诺而付出了十六年的努力。

在软件设计和编码中，如果确立了接口，也就对客户做出了承诺，这种承诺几乎没有改变的机会，时间越长越是如此，因为那是别人对你接口的使用已经遍布世界各地，当然前提是你的借口很出色，这样才能取得很多人的信赖和消费。

不要轻易说：“我爱你”，因为这是一生的承诺。