



链滴

设计模式之桥接模式

作者: [sologxl](#)

原文链接: <https://ld246.com/article/1543303532303>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

在网上，有很多桥接模式的说明文章

我这里也从我的角度说一下桥接模式的本质：

即，好好的去用好接口类。

哟西，是不是so easy，妈妈再也不用担心你堕落了。

这个其实很好理解，也很好应用的。

平时我们都是直接传具体类名称，但是如果类太多，类还相似，那就没必要定义那么多不同参数的函数了，接口的好处就显示出来了。我拿网上的一些代码给你们展示一下：

```
//抽象接口类
public abstract class HandsetSoft {
    public abstract void run();
}

//继承抽象类
public class HandsetGame extends HandsetSoft {

    @Override
    public void run(){
        println("游戏");
    }
}

//继承抽象类
public class HandetAddress extends HandsetSoft {

    @Override
    public void run(){
        println("通讯录");
    }
}
```

//这个类是手机品牌和手机软件的桥接主要类，此时已经实现抽象类传参，属于弱聚合。

```
public abstract class HandsetBrand {
    //弱聚合
    protected HandsetSoft soft;

    //设置手机软件
    public void setHandsetSoft(HandsetSoft soft){
        this.soft = soft;
    }

    //运行
    public abstract void run();
}

//品牌N
```

```

class HandsetBrandN extends HandsetBrand {
    @Override
    public void run(){
        soft.run();
    }
}

//品牌M
class HandsetBrandM extends HandsetBrand {
    @Override
    public void run(){
        soft.run();
    }
}

//测试
public class Main {
    HandsetBrand ab;
    //创建品牌N
    ab = new HandsetBrandN();

    //给品牌N安装游戏
    ab.setHandsetSoft(new HandsetGame());
    ab.run();

    //给品牌N安装通讯录
    ab.setHandsetSoft(new HandsetAddress());
    ab.run();
}

```

总结一下：桥接模式别看名称好像很高级，其实没那么多高大上的东西，只要大家记住一点，那就是发过程中，多使用抽象类和接口类，你会发现你会有很多收获的。