



链滴

设计模式学习笔记之桥接模式

作者: [hjljy](#)

原文链接: <https://ld246.com/article/1569146671021>

来源网站: [链滴](#)

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


```
}  
}
```

扩展抽象化角色

```
public class RefinedAbstractionA extends Abstraction{  
  
    public RefinedAbstractionA(Implementor impl) {  
        super(impl);  
    }  
    //扩展类需要实现的业务  
    public void otherOperation(){  
        System.out.println("小明吃完饭要干嘛? ");  
    }  
}
```

实现化角色

```
public interface Implementor {  
  
    /**  
     *实现抽象部分的具体业务操作  
     */  
    public abstract void operationImpl();  
}
```

具体实现角色

```
public class ConcretImplementorA implements Implementor {  
  
    @Override  
    public void operationImpl() {  
        operation();  
        System.out.println("实现类A的具体操作: 吃完饭, 去玩游戏");  
    }  
}  
  
public class ConcretImplementorB implements Implementor {  
  
    @Override  
    public void operationImpl() {  
        operation();  
        System.out.println("实现类B的具体操作: 吃完饭, 去打篮球");  
    }  
}
```

场景实例代码

资料里面写的很详细，我也是照着这个资料敲的一边代码然后慢慢理解的。

见资料：<https://www.cnblogs.com/betterboyz/p/9361784.html>

桥接模式总结

个人感觉桥接模式与其说是将抽象和实现分隔开来，还不如说是将不同的抽象和实现进行连接，产生相同的业务。

算是单一职责原则的一种体现吧。只负责抽象或者只负责实现。

1^