做不一样的码农

dboyaoao.blog.chinaunix.net

视野要开阔，创新灵感才能多

DBOYaoao

博客访问： 421004
博文数量： 83
博客积分： 821
博客等级： 军士长
技术积分： 1767
用 户 组： 普通用户
注册时间： 2011-10-23 16:17

个人简介

学校：上海交通大学软件工程 学历：硕士 行业：从事流媒体移动开发 QQ：412595942 邮箱：yiikai1987910@gmail.com

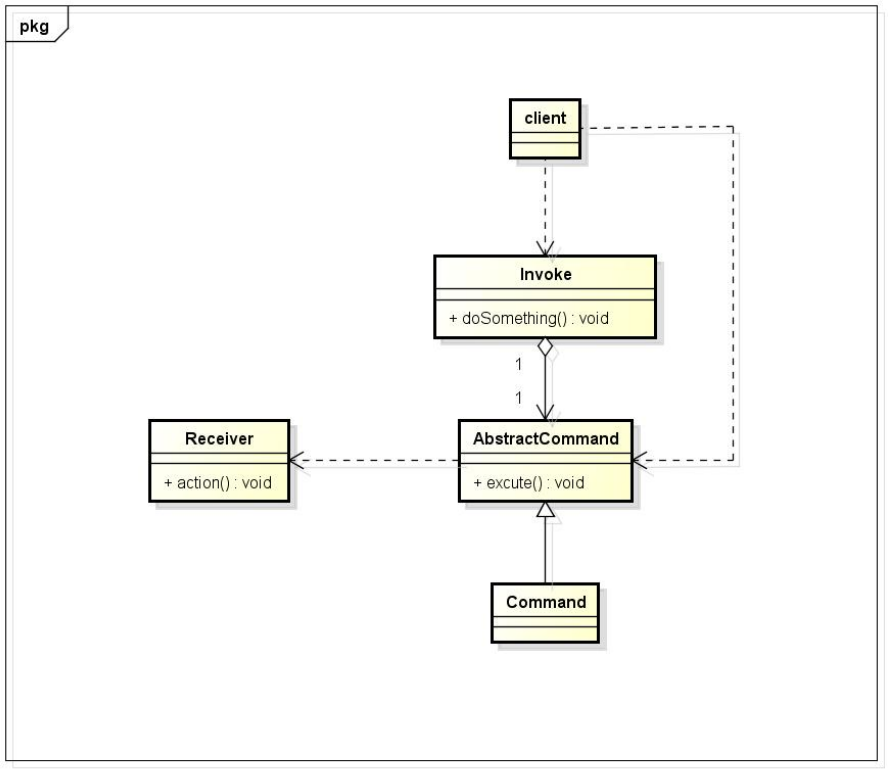
设计模式--命令模式

2014-02-17 18:34:44

分类： C/C++

命令模式其实和策略模式相似，都是将实现与调用解耦.

类图：



powered by Astah

而命令模式只是更形象的模拟了实现者和调用者之间的关系。来看下一般情况下：

如果我们需要实现一个订餐的程序，一般简单的想法就是一个顾客类，一个服务员，一个厨师类。调用的情况一般可以理解为由顾客类调用服务员的点餐方法，而服务员的点餐方法又调用了厨师的烧菜方法，你会发现，这就出现了耦合的问题，依赖于实现编程了。所以该如何去改呢？

其实现在想想设计模式的核心思想就是一个词----抽象！！！！

我们需要将这个三个类之间抽象出来，是他们之间不相互依赖于各自的实现，而是依赖于各自的抽象接口。从这点出发，就引出了commad模式，其实就是在三者之间分别建立命令抽象接口，以达到解耦的目的。

命令模式思想：

Command命令模式是一种对象行为型模式，它主要解决的问题是：在软件构建过程中，“行为请求者”与“行为实现者”通常呈现一种“紧耦合”的问题

是将实现者分装在一个类中，这个类一般就取名command类，随后通过这个command类进行调用者于实现者之间的联系。

现在我们就看看，在客户点餐的过程中如何使用命令模式

点击(此处)折叠或打开

```
class clientorder
{
public:
    clientorder();
    ~clientorder();

    void setorder(cookcommand* cook);
};
```

客户类，里面有一个订餐的接口，但是这个订餐的接口需要的是抽象的command类

点击(此处)折叠或打开

```
class cookcommand
{
public:
    cookcommand(){}
    ~cookcommand(){}

    virtual void excute() = 0;
};
```

commad类就是所谓的抽象层，用于调用实现者的功能。
接下来看下实现者是如何被封装的

点击(此处)折叠或打开

```
class chinesefood : public cookcommand
{
public:
    chinesefood(){}
    ~chinesefood(){}

    void excute()
    {
        printf("cook chinese food\n");
    }
};
```

其实实现者有可能是别的独有的类，这时就要用到组合的模式将实现者对象包含在分装类中（继承当然也可以，但是设计模式中推荐少用继承，多用组合：P）

这样就会发现，调用者和实现这分离了，点餐的顾客不需要知道厨师是怎么做菜的，只要知道自己想吃什么菜，命令厨师去做就可以了

点击(此处)折叠或打开

```
int _tmain(int argc, _TCHAR* argv[])
{
    clientorder client;
    client.setorder(new westfood());
    client.setorder(new chinesefood());
    system("pause");
    return 0;
}
```

看到没有，命令模式的实现方法和策略模式是很相似的，只不过策略模式是偏好于方法的抽象封装，而命令模式是对于实现类的封装，不过我感觉大体上还是差不多的

上一篇：设计模式--观察者模式

• linux下安装mysql
• CentOS 7:如何安装Gnome GUI ...
• Bash内置命令exec和重定向...

下一篇：ffmpeg 最简单的水印功能

相关热门文章

test123
linux dhcp peizhi roc
编写安全代码——小心有符号数...
关于Unix文件的软链接
使用openssl api进行加密解密...
求教这个命令什么意思，我是新...
一段自己打印自己的c程序...
sed -e "/grep/d" 是什么意思...
sql relay的c++接口
谁能够帮我解决LINUX 2.6 10...

给主人留下些什么吧！~~

评论热议

请登录后评论。
登录 注册