

设计模式之命令模式

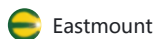
原创 Eastmount 2013-05-12 01:06:55 2723 收藏 2

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法、



¥9.90

订阅

在日常生活中,我们常常会遇到这样一些问题: 需要向某些对象发送请求, 但是并不知道请求的接收者是谁, 也不知道被请求的操作是哪个, 我们只需在程序运行时指定具体的请求接收者即可, 此时, 可以使用**命令模式**来进行设计, 使得**请求发送者与请求接收者消除彼此之间的耦合**, 让对象之间的调用关系更加灵活。

命令模式是非常实用的一个模式,比如最常用的就是我们编写项目中用到的**撤销\恢复(Undo\Redo)**就是通过**命令模式(Command)**来实现的;在如我们的计算器、小孩通过遥控器控制赛车运动, 这些都是通过命令模式实现的,下面就详细讲解命令模式.

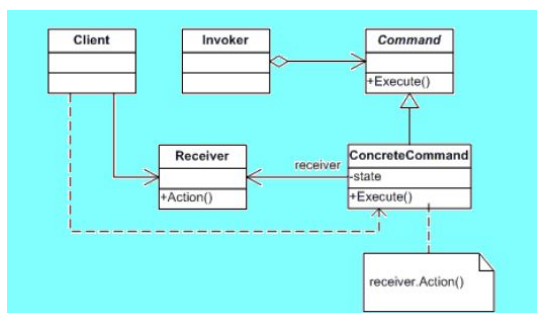
一.模式定义

命令模式(Command Pattern):**将一个请求封装为一个对象, 从而使我们可用不同的请求对客户进行参数化, 对请求排队或者记录请求日志, 以及支持可撤销的操作. 命令模式是一种对象行为型模式.**

在命令模式中发送者与接收者没有直接引用关系,发送请求的对象至需要知道如何发送请求,而不必知道如何完成请求,这就是命令模式的模式动机.

二.模式结构及组成

命令模式的结构图如下图所示:



其中命令模式的角色如下所示:

客户端 (Client) :创建一个具体的命令对象, 并确定其接收者.

请求者 (Invoker) :负责发出命令请求.

接收者 (Receiver) :具体实施或执行一个请求.

抽象命令 (Command) :给出所有具体命令的抽象接口.

具体命令 (ConcreteCommand) :实现抽象命令的方法,负责调用接收者.

三.命令模式实例

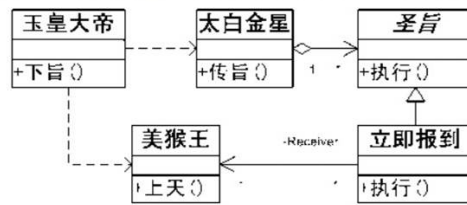
先举一个简单的例子:

1.玉帝下旨让太白金星去招悟空上天庭

在这个例子中共有五个角色:命令角色:圣旨,具体命令角色:具体圣旨, 立即上天庭报道,请求者角色:太白金星,接受者角色:美猴王,客户角色:Client玉帝.

它的模式图如下,这就是一个简单的命令模式的例子.

玉帝让太白金星招悟空上天庭：



下面这个例子是根据清华大学设计模式PPT中摘抄过来的，是一个比较经典的命令模式的例子，在使用遥控器中我们用到的就是命令模式。

2.电视机遥控器

电视机是请求的接收者，遥控器是请求的发送者，遥控器上有一些按钮，不同的按钮对应电视机的不同操作。抽象命令角色由一个命令接口来扮演，有三个具体的命令类实现了抽象命令接口，这三个具体命令类分别代表三种操作：打开电视机、关闭电视机和切换频道。显然，电视机遥控器就是一个典型的命令模式应用实例。

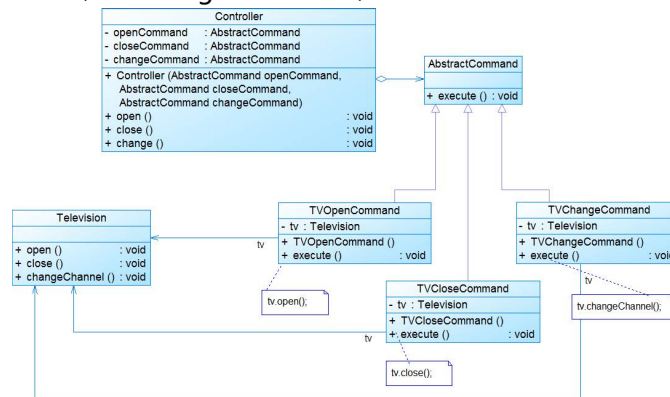
其中：

接受者--Television

请求者--Controller

抽象命令--AbstractCommand

具体命令--TCOpenCommand、TVChangeCommand、TVCloseCommand

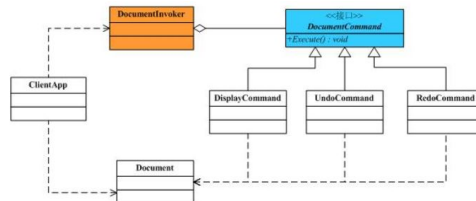


其中,在项目中的撤销\恢复(Undo\Redo)也是使用经典的命令模式, 这里就简单的给出它的类图,详细的Undo\Redo可见TerryLee的博客,他进行了详细的讲解并有源代码注释:

http://terrylee.cnblogs.com/archive/2006/07/17/Command_Pattern.html

3.Undo\Redo

请求者可以请求命令执行Display、Undo、Redo的操作队接受者(文章或工程)进行修改。

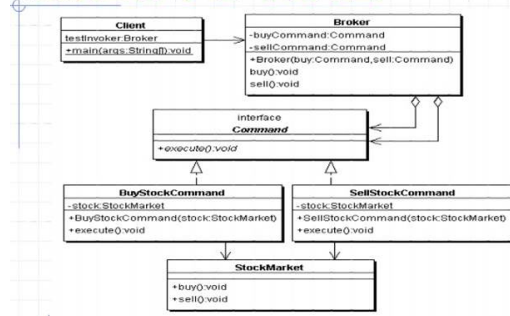


四.实例代码详解模式

下面通过股票交易系统的例子和详细代码分析命令模式

(由于忘记该代码是借鉴了某人的博客,在此处向他表示了对不起,作者只想借鉴你优秀的代码供大家学习,而当初没有记录下你的网址,在此说声抱歉.)

股票交易系统的设计图



具体的代码实现如下:

定义客户端代码,联系请求者、接受者和具体的买卖命令

客户端源代码

```

public class Client
{
    private Broker testInvoker; 定义请求者
    public static void main(String[] args)
    {
        定义接收者
        StockMarket stockMarket = new StockMarket();
        BuyStockCommand buy = new BuyStockCommand(stockMarket);
        SellStockCommand sell = new SellStockCommand(stockMarket);
        定义买和卖的命令类
        Broker myBroker = new Broker(buy, sell);
        myBroker.buy(); // Buy Shares
        myBroker.sell(); // Sell Shares
        赋予请求者买和卖的功能
    }
}

```

定义股票交易员, 请求者

股票交易员

```

public class Broker
{
    private Command buyCommand;
    private Command sellCommand; 定义买和卖的命令属性
    public Broker(Command aBuyCommand, Command aSellCommand)
    {
        buyCommand = aBuyCommand;
        sellCommand = aSellCommand;
    }
    void buy() {
        buyCommand.execute();
    }
    void sell() {
        sellCommand.execute(); 执行买与卖的命令
    }
}

```

定义抽象命令接口

抽象命令

```

public interface Command
{
    //correction abstraction
    void execute();
}
定义一个执行函数

```

定义股票市场, 是具体的接受者, 执行股票交易操作命令

接收者角色 (股票市场)

```

public class StockMarket
{
    public void buy() 接收者显示买入多少
    {
        System.out.println("You have bought in some stocks");
    }
    public void sell()
    {
        System.out.println("You have sold out some stocks");
    }
}
接收者显示卖出多少

```

定义具体的命令,它继承抽象命令接口,在具体类中定义了具体要执行的函数.

具体命令（买股票）

```
class BuyStockCommand implements Command
{
    private StockMarket stock; 定义市场属性
    public BuyStockCommand(StockMarket aStock)
    {
        stock = aStock;
    }
    public void execute() {
        stock.buy(); 执行买入操作
    }
}
```

具体命令（卖股票）

```
class SellStockCommand implements Command
{
    private StockMarket stock; 定义市场属性
    public SellStockCommand(StockMarket aStock)
    {
        stock = aStock;
    }
    public void execute() {
        stock.sell(); 执行卖出操作
    }
}
```

五.优缺点

最后,总结命令模式的优缺点。

优点:

- 1.降低对象之间的耦合度。
- 2.新的命令可以很容易地加入到系统中。
- 3.可以比较容易地设计一个组合命令。
- 4.调用同一方法实现不同的功能
- 5.可以用来做批处理操作
- 6.Undo\Redo操作

缺点:

因为每一个命令都需要设计一个具体命令类，所以可能会导致系统有过多的具体命令类，影响使用。

最后,我这篇文章主要讲述的是命令模式,并通过命令模式一些例子来叙述,同时该文章主要思想来自:作者Eastmount自己的思考,《大话设计模式》,《head first设计模式》等书,还有是清华大学的课件,以及一些其他人关于命令模式的博客文章,但由于作者的大意,找不到有些图来自哪?所以如果作者见到,请见谅,我主要是想分享自己的一些知识供大家学习和了解。如果有错误或不足之处,请读者原谅!!!

(By:Eastmount 2013-5-12-夜2点)



Eastmount  博客专家

原创文章 462 获赞 6725 访问量 525万+

关注

他的留言板