

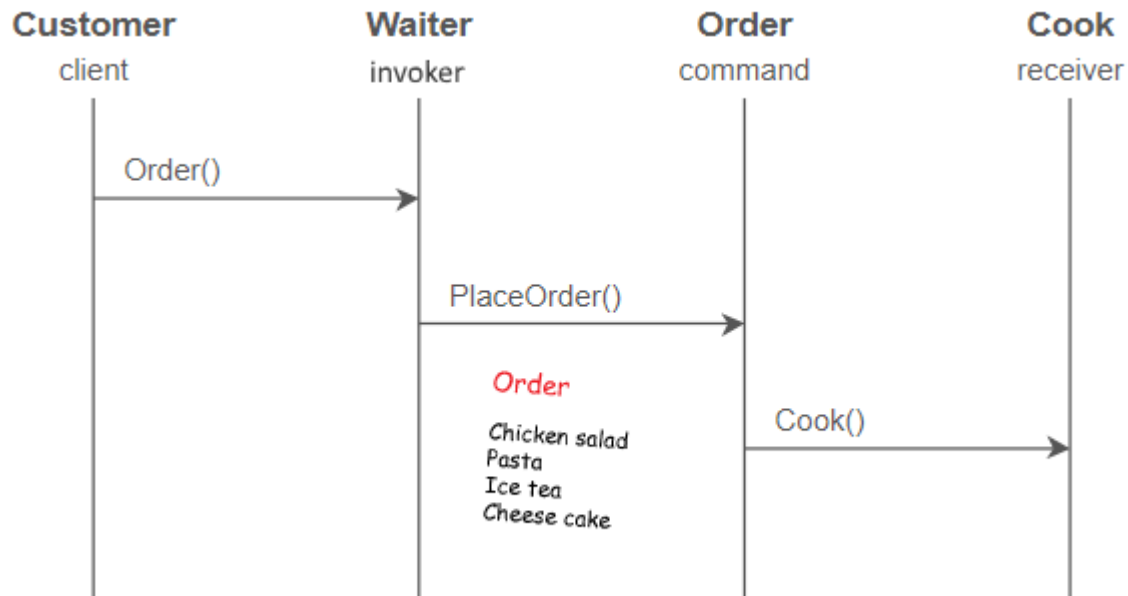
بررسی الگوهای طراحی - Command

توسط : محمد حسین زارع
@HoseinZare67: twitter
telegram
Join Us : @JavaLandCH

مفاهیم

- درخواست ها را در قالب اشیاء معرفی میکند
- **Object Oriented Callback**
- درخواست کننده را از پردازشگر جدا میکند
- اغلب برای مسأله ی undo به کار میرود
- نمونه ها
 - **java.lang.Runnable**
 - **javax.swing.Action**

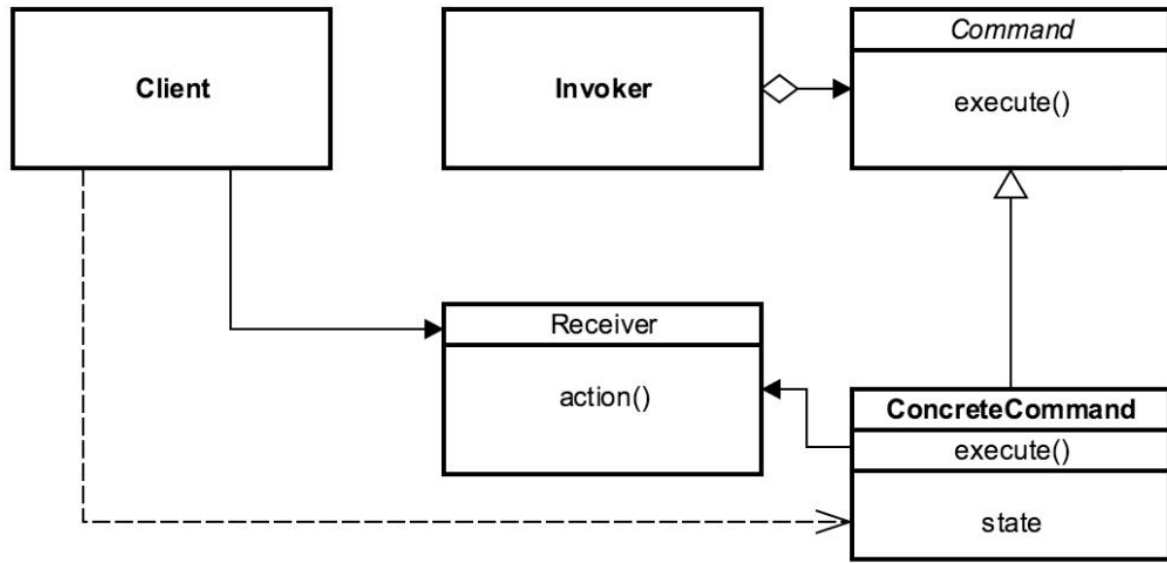
مفاهيم



طراحی

- ساخت یک شیء به ازای هر **command**
- **Interface Command**
- پیاده سازی متود اجرا
- **Invoker - Reciever**

طراحی

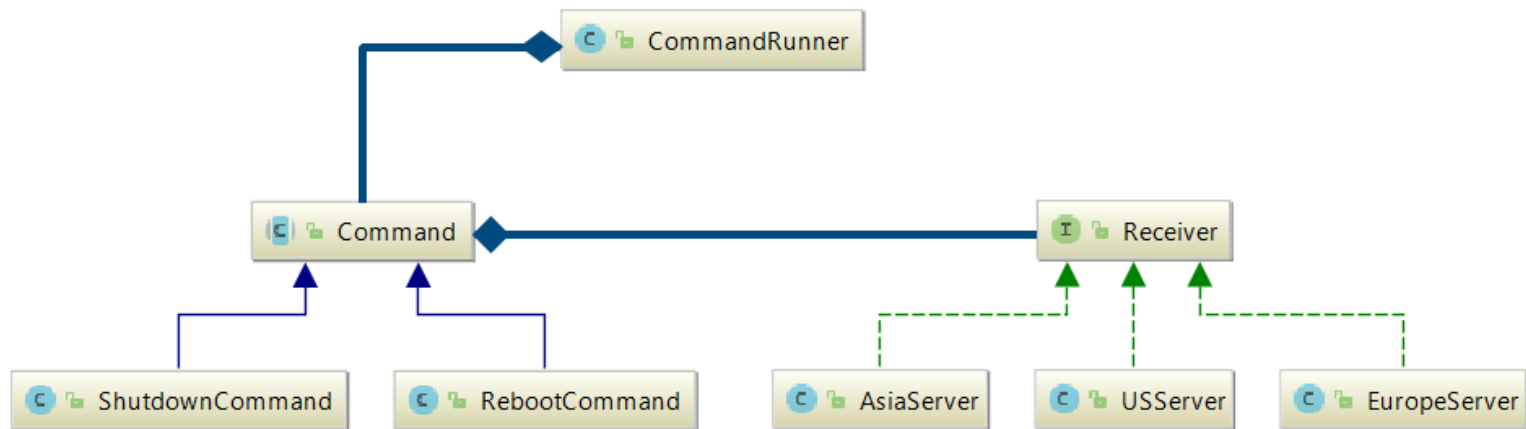


مثال 1

```
public static void main(String[] args) {  
    Task task1 = new Task(10, 12); //encapsulates request  
    Task task2 = new Task(11, 13);  
    Thread thread1 = new Thread(task1);  
    thread1.start(); //invoker  
    Thread thread2 = new Thread(task2);  
    thread2.start();  
}
```

```
class Task implements Runnable {  
    int num1;  
    int num2;  
    Task(int num1, int num2) {  
        this.num1 = num1;  
        this.num2 = num2;  
    }  
    @Override  
    public void run() { //execute method  
        System.out.println(num1 * num2); //receiver  
    }  
}
```

مثال 1



یک نمونه

```
public interface Receiver {  
    void connect();  
    void reboot();  
    void shutdown();  
    void disconnect();  
}
```

```
public class AsiaServer implements Receiver {...
```

```
public class EuropeServer implements Receiver {...
```


یک نمونه

```
public abstract class Command {  
  
    protected final Receiver receiver;  
  
    public Command(Receiver receiver) {  
        this.receiver = receiver;  
    }  
  
    public abstract void execute();  
}
```

یک نمونه

```
public class ShutdownCommand extends Command {  
  
    public ShutdownCommand(Receiver receiver) {  
        super(receiver);  
    }  
    @Override  
    public void execute() {  
        receiver.connect();  
        receiver.shutdown();  
        receiver.disconnect();  
    }  
}
```

یک نمونه

```
public class CommandRunner {  
  
    private final Command command;  
  
    public CommandRunner(Command command) {  
        this.command = command;  
    }  
  
    public void run() {  
        command.execute();  
    }  
}
```

یک نمونه

```
public static void main(String[] args) {  
    Receiver receiver = new AsiaServer();  
    Command command = new RebootCommand(receiver);  
    CommandRunner commandRunner = new CommandRunner(command);  
    commandRunner.run();  
}
```

جمع بندی

- **Encapsulate** کردن هر درخواست در یک شی
- گاهی اوقات برای پیاده سازی عملیات **undo** مورد استفاده قرار میگیرد
- ممکن است از الگوی **memento** برای نگهداری آخرین **command** استفاده کند
- ممکن است برای نگهداری آخرین **command** از **prototype** برای کپی آن شی استفاده کند
- درخواست کننده را از پردازشگر جدا میکند