

## #📖 日积笔记#

延续责任链模式的具体实现（填坑）：

直接上代码：（很多地方都是可以根据实际的业务场景去做出相应的调整 and 改变的）

### 1. 抽象父类

```
package Test.com.test;

import lombok.Setter;

/**
 * @author : laoli
 * @date : 2021/08/26 下午 8:07
 */
@Setter
public abstract class Action {

    /**
     * 可有可无的name...看需要
     */
    private String name;

    private Action nextAction;

    public Action(String name) {
        this.name = name;
    }

    /**
     * 指令开始方法
     *
     * @return
     */
    protected abstract boolean start();

    /**
     * 指令执行判断分支
     * 一. 执行成功
     * 1. 打印成功
     * 2. 打印失败，跳出
     * 二. 往下执行
     */
    public final boolean action() {
        if (start()) {
            done();
        } else {
            fail();
            return false;
        }
        if (nextAction != null) {
            nextAction.action();
        }
        return true;
    }
}
```

```

        protected void fail() {
            System.out.println(this.name + ", 执行失败!");
        }

        protected void done() {
            System.out.println(this.name + ", 执行完成!");
        }
    }
}

```

## 2. 节点实现类

```

/**
 * 按需加载驱动接通设备 生成语音
 *
 * @author : laoli
 * @date : 2021/08/26 下午 8:37
 */
public class OpenDeviceAction extends Action {

    public OpenDeviceAction(String name) {
        super(name);
    }

    @Override
    protected boolean start() {
        System.out.println("按需加载驱动接通设备");
        return true;
    }
}

```

```

/**
 * 初始化, 校验可拨外线
 *
 * @author : laoli
 * @date : 2021/08/26 下午 8:37
 */
public class InitDialAction extends Action {

    public InitDialAction(String name) {
        super(name);
    }

    @Override
    protected boolean start() {
        System.out.println("初始化, 校验可拨外线");
        return true;
    }
}

```

```

/**
 * 打电话
 *
 * @author : laoli
 * @date : 2021/08/26 下午 8:37

```

```

*/
public class DialAction extends Action {

    private final String phoneNum;

    public DialAction(String name, String phoneNum) {
        super(name);
        this.phoneNum = phoneNum;
    }

    @Override
    protected boolean start() {
        System.out.println("打电话" + phoneNum);
        return true;
    }
}

```

```

/**
 * 播放语音
 *
 * @author : laoli
 * @date : 2021/07/29 下午 8:47
 */
public class PlayFileAction extends Action {

```

```

    private final String filePath;

    public PlayFileAction(String name, String filePath) {
        super(name);
        this.filePath = filePath;
    }

    @Override
    protected boolean start() {
        System.out.println("播放语音" + filePath);
        return true;
    }
}

```

### 3.使用

```

/**
 * @author : laoli
 * @date : 2021/08/27 下午 7:27
 */
public class Call {

    public static boolean AlarmCall() {
        Action openDeviceAction = new OpenDeviceAction("打开设备");
        Action initDialAction = new InitDialAction("验证拨号");
        Action dialAction = new DialAction("拨号", "10086");
        // 文字转语音可以使用可以了解下，后面还可以跟很多执行节点我就懒得写了
        Action playFileAction = new PlayFileAction("播放语音", "D:\\test.wav");
        openDeviceAction.setNextAction(initDialAction);
        initDialAction.setNextAction(dialAction);
        dialAction.setNextAction(playFileAction);
        return openDeviceAction.action();
    }
}

```

```
public static void main(String[] args) {  
    System.out.println(AlarmCall());  
}  
}
```