

```

package com.sxt.strategy;

/**
 * 负责和具体的策略类交互
 * 这样的话，具体的算法和直接的客户端调用分离了，使得算法可以独立于客户端的变化
 * 如果使用spring的依赖注入功能，还可以通过配置文件，动态的注入不同的策略对象，
    动态的切换不同的算法
 * @author 江
 *
 */
public class Context {
    private Strategy strategy;    //当前采用的算法对象

    //可以通过构造器来注入
    public Context(Strategy straegy) {
        super();
        this.strategy = straegy;
    }

    //可以通过set方法来注入
    public void setStrategy(Strategy strategy) {
        this.strategy=strategy;
    }

    public void printprice(double s) {
        System.err.println("您的报价："+strategy.getPrice(s));
    }

}

```