首先我们有一大堆策略对象

【吐槽】小瘦仔 13:28:14

然后你的目标是，在某个条件下，执行某个策略

【吐槽】小瘦仔 13:28:20

那我就抽象一个对象

【吐槽】小瘦仔 13:28:29

条件策略关系类。

【吐槽】小瘦仔 13:28:40

将条件与策略进行结合

【吐槽】小瘦仔 13:29:11

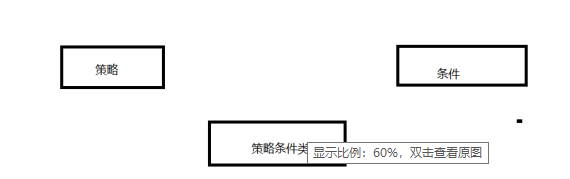
好了，这个类的功能是什么呢？

【吐槽】小瘦仔 13:29:47

你给我传递一个具体的num，然后我条件策略类来判断，我是否支持你这个num

【吐槽】小瘦仔 13:30:22

如果支持，你就可以拿条件策略类关联的策略对象来执行你的操作



【吐槽】小瘦仔 13:31:32  
然后呢，我需要一个域对象，腔体对象  
  
【吐槽】小瘦仔 13:31:43  
将所有的策略条件关系类维护起来

【吐槽】小瘦仔 13:32:28

这个类提供的功能就是，根据用户提供的条件

【吐槽】小瘦仔 13:32:40

去获取对应的条件策略类

【吐槽】小瘦仔 13:33:06

然后，去执行对应的策略

【吐槽】小瘦仔 13:33:17

不知道我说清楚没有

【吐槽】小瘦仔 13:35:39  
其实这种问题没有定势，但是我这样确实合理解决问题了。我们要学会，通过问题来抽象对象，用对象来交互逻辑。  
  
【吐槽】小瘦仔 13:36:01  
甚至你传递的条件是int也好，对象也罢，字符串也好，我都可以抽象  
  
【吐槽】小瘦仔 13:36:29  
你的问题可以更复杂，我传数字1，执行策略1  
  
【吐槽】小瘦仔 13:36:42  
我传一个Stirng对象“aaa”，你执行策略2  
  
【吐槽】小瘦仔 13:37:47  
你甚至可以传一个对象  
  
【吐槽】小瘦仔 13:38:00  
在对象多个条件的组合下，决定执行哪个策略

【吐槽】小瘦仔 13:38:37  
我这里目前抽象的是int  
  
【吐槽】小瘦仔 13:38:52  
如果业务需要，你可以抽象，对象，在内部执行复杂的逻辑