链滴

[阅读] 敏捷软件开发 —— 薪水支付案例研究（二）

作者：lizhongyue248

原文链接：https://ld246.com/article/1554562233661

来源网站：链滴

放假总是悠闲的，找不到事做就看看书吧。

那些在任何情况下都是e迷离的事物，其美丽是就其本性而言的。美丽的终结也是就其本性而言的，美并不是其本性的一部分。

就第一篇写的记录开始学习。

推迟考虑数据库

我们构建一个应用程序，最先考虑到的应该是数据库，往往是先设计数据库然后再来设计，对于该类题同样，他可以使用某些关系型数据库，并且从需求中可以清楚地知道表和字段的可能样子。可以容设计出一个可用的数据库模式，然后再构建一些查询。不过，在这种方法产生的应用程序中，数据库成了关注的中心。

我们换个思路，把数据库看成我们实现的细节！尽可能地推迟考虑数据库。按照**抽象**的定义，我们应把本质的部分放大，无关紧要部分的去掉，在项目的当前阶段数据库就是无关紧要的，他不过是一项来存储和访问数据的技术而已。

用例

那我们如何分析呢？在一开始的 xp 编程中，提到一个非常重要的概念——用户素材。**用例**就是用稍一点的细节描述的用户素材，在进行分析的时候，我们关注用户素材和验收测试。不陷入过多的细节。

搜寻

来看看这章做了哪些事

1. 将用户描述转化为了具有细节的用例，而不是过于简单用户素材。
2. 对用例进行一个一个的分析，弄清楚系统怎样去响应这些操作。

3. 前半部分通过对六个用例的分析，初步确定 COMMAND 模式 以及可能的 静态模型。

4. 通过一个用例以及他的变体，改变初步确定的设计模式，由 COMMAND 模式 改为 STRATEGY 模式 ，同时加上了 NULL OBJECT 模式设计出修订后的系统类图。

5. 通过协同图明确了可能出现的用例变体/情景。

6. 寻找描述中的抽象。

7. 通过抽象发现从属关系，移除 NULL OBJECT 模式。

设计

在设计中，**不包含任何对数据库的内容**

1. 首先将用户素材n加沙改一些适当的细节描述成为用例

2. 通过对用例的分析，获取到有用的信息以及设计的洞察力

3. 设计不是一成不变的，它可以随着迭代的进行不断修改，最终迭代完成后确定下来

4. 遵循面向对象设计的五大原则

5. 寻找出有用的抽象

6. 设计模式不是用的越多越好

总结

这章节似乎是一个小型会议一般，在商讨第一次迭代的点滴与内容，值得注意的是，多次强调妖抵制数据库的诱惑。i回想一下，我们似乎总是在**面向数据库编程**，这并不是一个好习惯。同时也多次提到五大原则，在设计时，五大原则是能够架构一个友好的软件架构的重要因素。这次的目的只是为了发一种思考，但是依旧都是可变的，只是一个快速设计的一个会话展现。