

06讲模式与框架：它们的关系与误区？



在学习程序设计的路上，你一定会碰到“设计模式”，它或者给你启发，或者让你疑惑，并且你还会发现在不同的阶段遇到它，感受是不同的。而“开发框架”呢？似乎已是现在写程序的必备品。那么框架和模式又有何不同？它们有什么关系？在程序设计中又各自扮演什么角色呢？

设计模式

设计模式，最早源自 GoF 那本已成经典的《设计模式：可复用面向对象软件的基础》一书。该书自诞生以来，在程序设计领域已被捧为“圣经”。

软件设计模式也是参考了建筑学领域的经验，早在建筑大师克里斯托弗·亚历山大（Christopher Alexander）的著作《建筑的永恒之道》中，已给出了关于“模式”的定义：

每个模式都描述了一个在我们的环境中不断出现的问题，然后描述了该问题的解决方案的核心，通过这种方式，我们可以无数次地重用那些已有的成功的解决方案，无须再重复相同的工作。

而《设计模式》一书借鉴了建筑领域的定义和形式，原书中是这么说的：

本书中涉及的设计模式并不描述新的或未经证实的设计，我们只收录那些在不同系统中多次使用过的成功设计；尽管这些设计不包括新的思路，但我们用一种新的、便于理解的方式将其展现给读者。

虽然该书采用了清晰且分门别类的方式讲述各种设计模式，但我相信很多新入门的程序员在看完该书后还是会像我当年一样有困扰，无法真正理解也不知道这东西到底有啥用。

早年我刚开始学习 Java 和面向对象编程，并编写 JSP 程序。当我把一个 JSP 文件写到一万行代码时，自己终于受不了了，然后上网大量搜索到底怎样写 JSP 才是对的。之后，我就碰到了《设计模式》一书，读完了，感觉若有所悟，但再去写程序时，反而更加困扰了。

因为学“设计模式”之前，写程序是无所顾忌，属于拿剑就刺，虽无章法却还算迅捷。但学了一大堆“招式”后反而变得有点瞻

前顾后，每次出剑都在考虑招式用对没，挥剑反倒滞涩不少。有人说：“设计模式，对于初窥门径的程序员，带来的麻烦简直不逊于它所解决的问题。”回顾往昔，我表示深有同感。

后来回想，那个阶段我把《设计模式》用成了一本“菜谱”配方书。现实是，没做过什么菜只是看菜谱，也只能是照猫画虎，缺少好厨师的那种能力——火候。初窥门径的程序员其实缺乏的就是这样的“火候”能力，所以在看《设计模式》时必然遭遇困惑。而这种“火候”能力则源自大量的编程设计实践，在具体的实践中抽象出模式的思维。

“设计模式”是在描述一些抽象的概念，甚至还给它们起了一些专有名字，这又增加了一道弯儿、一层抽象。初窥门径的程序员，具体的实践太少，面临抽象的模式描述时难免困惑。但实践中，经验积累到一定程度的程序员，哪怕之前就没看过《设计模式》，他们却可能已经基于经验直觉地用起了某种模式。

前面我说过我刚学习编程时看过一遍《设计模式》，看完后反而带来更多的干扰，不过后来倒也慢慢就忘了。好些年后，我又重读了一遍，竟然豁然开朗起来，因为其中一些模式我已经在过往的编程中使用过很多次，另一些模式虽未碰到，但理解起来已不见困惑。到了这个阶段，其实我已经熟练掌握了从具体到抽象之间切换的思维模式，设计模式的“招数”看来就亲切了很多。

在我看来，模式是前人解决某类问题方式的总结，是一种解决问题域的优化路径。但引入模式也是有代价的。设计模式描述了抽象的概念，也就在代码层面引入了抽象，它会导致代码量和复杂度的增加。而衡量应用设计模式付出的代价和带来的益处是否值得，这也是程序员“火候”能力另一层面的体现。

有人说，设计模式是招数；也有人说，设计模式是内功。我想用一种大家耳熟能详的武功来类比：降龙十八掌。以其中一掌“飞龙在天”为例，看其描述：

气走督脉，行手阳明大肠经商阳...此式跃起凌空，居高下击，以一飞冲天之式上跃，双膝微曲，提气丹田，急发掌劲取敌首、肩、胸上三路。

以上，前半句是关于内功的抽象描述，后半部分是具体招数的描述，而设计模式的描述表达就与此有异曲同工之妙。所以，设计模式是内功和招数并重、相辅相成的“武功”。

当你解决了一个前人从没有解决的问题，并把解决套路抽象成模式，你就创造了一招新的“武功”，后来的追随者也许会给它起个新名字：某某模式。

开发框架

不知从何时起，写程序就越来越离不开框架了。

记得我还在学校时，刚学习 Java 不久，那时 Java 的重点是 J2EE（现在叫 Java EE 了），而 J2EE 的核心是 EJB。当我终于用“JSP + EJB + WebLogic（EJB 容器）+ Oracle数据库”搭起一个 Web 系统时，感觉终于掌握了 Java 的核心。

后来不久，我去到一家公司实习，去了以后发现那里的前辈们都在谈论什么 DI（依赖注入）和 IoC（控制反转）等新概念。他们正在把老一套的 OA 系统从基于 EJB 的架构升级到一套全新的框架上，而那套框架包含了一堆我完全没听过的新名词。

然后有前辈给我推荐了一本书叫 J2EE Development Without EJB，看完后让我十分沮丧，因为我刚刚掌握的 Java 核心技术 EJB 还没机会出手就已过时了。

从那时起，我开始知道了框架（Framework）这个词，然后学习了一整套的基于开源框架的程序开发方式，知道了为什么 EJB 是重量级的，而框架是轻量级的。当时 EJB 已步入暮年，而框架的春天才刚开始来临，彼时最有名的框架正好也叫 Spring。如今框架已经枝繁叶茂，遍地开花。

现在的编程活动中，已是大量应用框架，而框架就像是给程序员定制的开发脚手架。一个框架是一个可复用的设计组件，它统

一定定义了高层设计和接口，使得从框架构建应用程序变得非常容易。因此，框架可以算是打开“快速开发”与“代码复用”这两扇门的钥匙。

在如今这个框架遍地开花的时代，正因为框架过于好用、易于复用，所以也可能被过度利用。

在 Java 中，框架很多时候就是由一个或一些 jar 包组成的。在前几年（2012 年的样子）接触到一个 Web 应用系统，当时我尝试去拷贝一份工程目录时，意外发现居然有接近 500M 大小，再去看依赖的 jar 包多达 117 个，着实吓了一跳。在 500M 工程目录拷贝进度条缓慢移动中，我在想：“如今的程序开发是不是患上了框架过度依赖症？”

我想那时应该没有人能解释清楚为什么这个系统需要依赖 117 个 jar 包之多，也许只是为了完成一个功能，引入了一个开源框架，而这个框架又依赖了其他 20 个 jar 包。

有时候，框架确实帮我们解决了大部分的脏活累活，如果运气好，这些框架的质量很高或系统的调用量不大，那么它们可能也就从来没引发过什么问题，我们也就不需要了解它们是怎么去解决那些脏活、累活的。但若不巧，哪天某个框架在某些情况下出现了问题，在搞不懂框架原理的情况下，就总会有人惊慌失措。

如今，框架带来的束缚在于，同一个问题，会有很多不同框架可供选择。如何了解、评估、选择与取舍框架，成了新的束缚。

一些知名框架都是从解决一个特定领域问题的微小代码集合开始发展到提供解决方案、绑定概念、限定编程模式，并尝试不断通用化来扩大适用范围。

这样的框架自然不断变得庞大、复杂、高抽象度。

我一直不太喜欢通用型的框架，因为通用则意味着至少要适用于大于两种或以上的场景，场景越多我们的选择和取舍成本越高。另外，通用意味着抽象度更高，而现实是越高的抽象度，越不容易被理解。例如，人生活在三维世界，理解三维空间是直观的，完全没有抽象，理解四维空间稍微困难点，那五维或以上理解起来就很困难了。

框架，既是钥匙，也是枷锁，既解放了我们，也束缚着我们。

两者关系

分析了模式，解读了框架，那么框架和模式有什么关系呢？

框架和模式的共同点在于，它们都提供了一种问题的重用解决方案。其中，框架是代码复用，模式是设计复用。

软件开发是一种知识与智力的活动，知识的积累很关键。框架采用了一种结构化的方式来对特定的编程领域进行了规范化，在框架中直接就会包含很多模式的应用、模式的设计概念、领域的优化实践等，都被固化在了框架之中。框架是程序代码，而模式是关于这些程序代码的知识。

比如像 Spring 这样的综合性框架的使用与最佳实践，就隐含了大量设计模式的套路，即使是不懂设计模式的初学者，也可以按照这些固定的编程框架写出符合规范模式的程序。但写出代码完成功能是一回事，理解真正的程序设计又是另外一回事了。

小时候，看过一部漫画叫《圣斗士》。程序员就像是圣斗士，框架是“圣衣”，模式是“流星拳”，但最重要的还是自身的“小宇宙”啊。

我相信在编程学习与实践的路上，你对设计模式与开发框架也有过自己的思考。欢迎给我留言，说说你有过怎样的认识变化和体会，我们一起讨论。

程序员进阶攻略

每个程序员都应该知道的成长法则

胡峰 京东成都研究院 技术专家



精选留言



third

心得如下：

1.模式和框架都是前人总结之下的经验

模式是代码层面，解决单个问题的成功方法

框架是设计层面，解决一系列问题的成功方法

挺感谢前人的奋进的

2.我们一边在使用工具，获得便捷，一边在被工具所驯化，讨厌麻烦，变得简单。比如重新理解框架原理。

2018-08-15 14:31

作者回复

总结的好棒

2018-08-15 20:34



feifei

现在还记得第一次看设计模式的那个场景，完全懵逼，然后看解决的问题，是如此的优雅！责任链，将接口容器化，链式调用，将业务逻辑的扩展变得轻松，然后我花了2周将代码做到能，不看源码，手打出来，然后慢慢的在花了2周终于理解了！老大推荐了一本《head first 设计模式》，当我看完这本书后，感觉豁然开朗，抽象的思维就开始在脑袋里慢慢形成！

2018-08-16 20:44



who am i

i

作者使用类比和排比的手法，让我们更清楚框架，模式的联系和区别。希望后面的文章也能这样，会让我们更能理解明白。

2018-08-19 09:13

作者回复

恩，会的

2018-08-19 21:31



godtrue

我的理解：

框架-是非业务代码功能的复用

模式-是经验优秀开发方式的复用

一个牛逼的框架，一定会使用一些牛逼的设计模式。框架，是牛人编写出来具体解决一类问题的半成品（相对于使用者），使

用户主要的工作是按照框架的结构，填充自己业务代码。

设计模式是解决一类特定问题的最佳实践经验，类似内功，怎么用什么时候什么地方用什么是使用者根据需求和积累选择的

。

比喻一下：

框架有点类似游泳圈

模式有点类似潜水时该怎么办或仰泳时候该怎么办的最佳经验。

2018-08-16 07:30

作者回复

感觉还是我最后的比喻更贴切些，哈哈

2018-08-19 21:32



Franklin.du

看过好几遍设计模式的书，开发中能想到一些相关的设计模式，用起来就是“菜谱”的感觉。编程实践还是不够多，达不到融会贯通的地步。看来有了秘笈“降龙十八掌”也不是那么容易练成的。

2018-08-15 08:02

作者回复

编程，是实践的艺术啊

2018-08-15 20:39



小特

胡峰老师您好！1.我之前看过《大话设计模式》等设计模式的书籍，上面的demo也实践了，但是在实践中似乎从来没应用过，请问具体实践中应该应用什么方法去学习设计模式？如何才能真正做到掌握并且应用设计模式？2.Java相关的框架实在太多，自己在项目开发中用了很多框架，很多问题基本百度能解决，但是自己还是停留在会用的阶段，具体原理还是不懂，请问有必要去研究框架原理吗？学习框架达到什么样的水准才算可以呢？

2018-10-09 08:52

作者回复

框架会用只是第一步，原理一定要明白，不懂就要去看文档和代码，框架搞透了，设计模式的应用自然就明了了。框架的实现中大量应用了设计模式

2018-10-09 18:33



云学

设计模式这段有同感，刚开始硬着头皮学根本消化不良，若干年后自然而然就领悟了，所以我坚持“以术悟道”而不是“以道生术”

2018-08-21 22:34

作者回复

其实并不矛盾，我觉得

2018-08-22 19:00



夜海

小宇宙，圣衣，流星拳的比喻很形象。立马就理解了

2018-08-17 20:34

作者回复

2018-08-17 23:48



Jiantao

模式学习：反复实践，理解思想！框架：解决问题的高效工具。

2018-08-16 23:57



L.

小公司还没有属于自己的开放框架，怎么能搭建一个开发框架呢

2018-08-15 20:34

作者回复

开源的吧，基本能满足，遇到不满足的情况再扩展

2018-08-16 20:37



董某越

是否推荐深入研究某个框架的组成和应用？某种语言的框架又是否大同小异。感觉工作中每种语言所搭建的系统都倾向于使用一种框架，这是否是是一种提高代码阅读性的趋势呢？

2018-08-15 15:14

作者回复

推荐深入研究一次，以后可能就触类旁通了

2018-08-15 19:15



June
Yuan

相比编程而言，感觉设计模式和框架的学习是比较有门槛的，理解它们的前提，是要理解它们所解决的问题。曾经学习设计模式，看看书还能一知半解，但是像 IoC、AOP, 当时在没有任何 WEB 项目经验的情况下，看了一些博客文章试图简单了解，结果依然一头雾水。实际工作以后，在真实的项目和业务需求里泡得多了，有的原先云里雾里的东西才逐渐变得自然。

2018-08-15 14:07

作者回复

是这样的，实践和理论要交叉进行，反复几次可能就豁然开朗了

2018-08-15 20:36



Sawyer

关于设计模式的使用中，提到一个词是“火候”。这个才是真正区分一个新手和老手的关键因素。

2018-08-15 12:42

作者回复

火候感，其实挺微妙的

2018-08-15 20:37



绝露

框架是半成的产品，而设计模式只是一种针对特定问题的通用方案。产品里采纳了对应领域的一系列方案。

2018-08-15 09:01



艾尔欧唯伊

设计模式是一种编程方式或风格，框架更多是提供类库或简化一些重复性，比较复杂的通用代码。不是一个类型的定义。另外，一个框架依赖的jar真的很可怕，最困惑的就是不知道哪些是必要，哪些不是。。。有时候可能只要这个jar里面的一个类，就得全引入，不然编译都过不去。。。

2018-08-15 08:47



小董

在看设计模式那块的时候，也是似懂半懂的。实战中运用了，才明白设计模式的意义存在。我想问胡老师下有什么更好的方式去更深层次的理解它吗？有时候觉得，可能还是没有达到那个境界。希望能指导点拨下.....

2018-10-23 08:44

作者回复

有时就是代码量和时间经验够了，突然就明白，抽象的东西总是突然就顿悟了

2018-10-23 18:13



Dunizb

笔记：

.....

模式和框架都是前人总结之下的经验

1. 模式是代码层面，解决单个问题的成功方法
2. 框架是设计层面，解决一系列问题的成功方法

引入模式也是有代价的。设计模式描述了抽象的概念，也就在代码层面引入了抽象，它会导致代码量和复杂度的增加。而衡量应用设计模式付出的代价和带来的益处是否值得，这也是程序员"火候"能力另一层面的体现。

框架采用了一种结构化的方式来对特定的编程领域进行了规范化，在框架中直接就会包含很多模式的应用、模式的设计概念、领域的优化实践等，都被固化在了框架之中。框架是程序代码，而模式是关于这些程序代码的知识。

2018-10-15 01:01



Dunizb

模式和框架都是前人总结之下的经验

1. 模式是代码层面，解决单个问题的成功方法
- 框架是设计层面，解决一系列问题的成功方法

引入模式也是有代价的。设计模式描述了抽象的概念，也就在代码层面引入了抽象，它会导致代码量和复杂度的增加。而衡量应用设计模式付出的代价和带来的益处是否值得，这也是程序员"火候"能力另一层面的体现。

框架采用了一种结构化的方式来对特定的编程领域进行了规范化，在框架中直接就会包含很多模式的应用、模式的设计概念、领域的优化实践等，都被固化在了框架之中。框架是程序代码，而模式是关于这些程序代码的知识。

2018-10-15 00:59



小特

峰哥，工作中用到了Struts2, Springmvc, Spring, mybatis, hibernate。如果看源码搞懂原理，您建议从哪个框架着手更容易一些？这些框架都需要搞明白原理看懂源码吗？

2018-10-09 21:21

作者回复

搞懂原理不一定都要看源码吧，先从文档开始，想想如果框架提供的这个功能我会怎么实现，想不出来，就去看它怎么实现的

2018-10-10 18:47



Starbug

继续补学习笔记：

1. 设计模式

解决一类问题的套路，抽象成了一个“武功秘籍”

内功：看待问题的思维模式

招式：代码结构

2. 框架

解决一系列特定问题（往往是一个应用场景）而产生的代码结构。

类似于开发人员的脚手架，软件大厦的基础框架给你搭好了，只用实现具体一个窗户之类的具体功能就好

2018-09-23 22:17