

# 开篇词 | 带你吃透DDD

2022-12-06 钟敬 来自北京



天下无鱼

<https://shikey.com/>

[课程介绍 >](#)

《手把手教你落地DDD》



讲述：钟敬

时长 12:48 大小 11.70M



你好，我是钟敬，欢迎和我一起来学习领域驱动设计。

我是一名在 IT 界奋斗了二十多年的老兵。目前在 Thoughtworks 公司担任首席咨询师，也是数字化转型与运营团队的 DDD 交付负责人。平时的工作主要是给客户提供软件开发方法学、架构治理、研发效能提升等方面的帮助。

DDD，也就是“领域驱动设计”，在软件工程里，属于软件开发方法学的范畴。它继承了面向对象和敏捷方法的精华，并提炼了一套更容易掌握的原则、模式和实践，特别适合复杂的企业应用的开发。

这些年，随着数字化转型的浪潮，我们为很多家大中型企业，在几十个项目里引入了 DDD，解决了很多实际的痛点和问题。

比如说，有的企业的业务很复杂，业务人员和开发人员总是很难把需求完全说清楚，直到写完代码，用户测试的时候，才知道开发人员的理解跑偏了。采用了 DDD 以后，就可以利用领域建模，把业务知识充分地严格化和可视化。这样，在分析设计阶段，就能对齐业务和开发的理解。

还有一些系统的需求变化很快，开发很难跟上业务的节奏。有时为了能够在短期内应付变化，又会写出“打补丁”式的代码，给系统留下隐患。这种情况下，采用领域建模的方法，把领域知识抽象化，把相对稳定和相对变化的部分分开，并把变化的部分设计成可插拔、可配置的，就能开发出灵活、易扩展的系统。

另外，还有一些项目希望采用微服务架构，但是缺乏拆分的依据，不知道怎么确定服务的粒度和边界。这时通过 DDD 的领域建模和限界上下文的划分，就可以给微服务的拆分提供坚实的基础。

还有一种常见的问题，就是有些企业系统的架构和代码已经严重腐化了，希望通过重构来提高质量和可维护性。但是，应该往哪个方向重构呢？怎么在重构过程中降低风险，并且兼顾业务需求呢？

借助 DDD，就可以通过一套系统化的方法，为系统建立目标模型和设计目标架构。同时，再利用 DDD 的“柔性设计”，并且结合演进式架构和代码重构等等技术，就能很好地解决问题。

## 落地 DDD 有哪些困难？

事实上，我们发现，在开发微服务，遗留系统改造、架构治理、企业架构、研发效能提升、架构师培养等很多方面，DDD 都有用武之地。

DDD 有这么多的优势，可是在实践的时候，不少小伙伴都觉得不太容易学习和落地。在我们推广的过程里，发现主要有下面这几个难点。

**第一，是领域建模不好掌握。**不论是在 DDD，还是传统的面向对象方法学里面，领域建模都是整个过程的核心，这里面还是有不少的方法和技巧的。不过《领域驱动设计：软件核心复杂性应对之道》（以下简称《DDD》）原书的作者，大概是假设读者已经有一定的面向对象建模基础了，所以有些东西讲得也不太透。

**第二，领域模型不好实现。**也就是说，就算有人把领域模型建好了，不少团队也不会按照领域模型去设计数据库，设计架构，并且编写代码。而 DDD 强调，领域模型的实现必须和模型本身保持高度一致。那么，同样是因为面向对象的基础不牢，所以不懂模型的实现方法。这方面，在原书里，也没有说得太明白。

**第三，是概念混乱。**有些小伙伴可能已经看了不少书，听了不少讲座，但是发现 DDD 的很多概念，老师们说的好像都不一样，不知道到底该信谁的。这是因为很多人没有看原著，或者没看懂。结果，反而接受了很多二手、三手的材料。有些材料又是理解不一，甚至以讹传讹。

**第四，是时代的发展。**现在离 DDD 的提出，已经有二十年了。不论是技术环境还是开发理念，都已经发生了不小的变化。如果不能与时俱进，对 DDD 原来的一些理念和方法进行发展，那也很难落地。

**最后，还有很多具体落地时才会遇到的问题。**就算某个人自己已经掌握得很不错了，但是在团队里一推广，又会有很多障碍。比如说怎么让新、旧开发方法共存；怎么把 DDD 融入到日常开发流程；怎么评估 DDD 实施的成效等等。

## DDD 怎样才能“更好学”？

虽然说了不少困难，但是你也大可不必担心。DDD 的学习还是有方法，有套路的。我希望通过这门课，把 DDD 的原理和我们在实践里总结的经验，分享给你。

为了帮你和你的团队顺利掌握和落地 DDD，解决前面说的疑难问题，这门课在设计时遵循了四个原则，我们一个一个说一下。

### 第一，基于认知规律

这门课基于一个贯穿始终的案例（企业管理系统）。这个案例模拟了敏捷软件开发过程，分成了三个迭代。每个迭代分成若干节课，每节课一方面引入新的知识，另一方面也会呼应和深化之前的知识。这三个迭代是这么安排的。

#### 迭代一：夯实基础

第一个迭代，我们会开启一个“麻雀虽小，五脏俱全”的项目。通过这个迭代，你就可以打通一个“需求 - 模型 - 代码”的最小闭环，对 DDD 的过程初步形成一个完整的“感觉”。

首先，我会利用 DDD 世界里一种常用的方法“事件风暴”，来和你一起梳理行为需求。在这个过程中，也会介绍“统一语言”。之后，我们一起来实操 DDD 的核心技能“领域建模”，并引入实体、关联、模块等几个重要的模式。接下来是模型的实现，也就是根据模型建立数据库和编写代码。你会学到 DDD 的分层架构，工厂、仓库、领域服务等模式。还会涉及到怎样封装、怎样实现领域逻辑等等。

学完这个迭代，你应该能打下一个比较好的基础，并且有能力试着处理一些不太复杂的项目了。

## 迭代二：渐入佳境

第二个迭代，我们进一步深入，讲几个 DDD 里争议较大，不太好掌握的内容。

我会先从理论、模型和编码层面帮你理解“聚合”。同时，会进一步带你提升领域建模能力，深化对分层架构和代码封装的理解。接着，我们会用几节课讲值对象，带你理解值对象的本质和优点，并解决值对象在建模和编程上的一些具体问题。最后，我们会学习一个重要的建模技巧——泛化，这是领域建模由初级走向中、高级的关键技能。

经过第二个迭代的揣摩，你应该会有渐入佳境的感觉了。

## 迭代三：掌握更高级的技能

经过前两个迭代，你对一个开发组范围内的项目应该有一定信心了。但是如果项目范围更大，该怎么处理呢？

在第三个迭代，我会介绍“限界上下文”模式，通过分而治之来维护概念的一致性。在这个基础上，再进一步学习微服务设计。

接下来，我会带你了解事件驱动和 CQRS 这两个重要的架构模式。然后，我们会讨论怎样为更加灵活多变的业务建模，并深化对泛化的理解。最后，我们会解决在实际落地过程中会遇到的几个问题，比如 DDD 切入点的选择，遗留系统的改造等等。

经过这三个迭代的学习，你已经掌握了 DDD 最核心的技能，再经过一段时间的实践和消化，就可以向 DDD“专家”迈进了。

## 第二，是立足原书，有所发展

这门课基于《DDD》原书的理论框架，力求准确地反映书里的知识体系。但这也并不意味着本本主义，不是说书里的一定就是对的。我会首先解释清楚原书的概念，并对值得商榷的地方做深入分析。对原书里没有讲透的部分，也会通过实例进一步讲解。



随着时代的发展，还出现了和 DDD 相关的新内容和新观点，比如前面提到过的微服务设计、事件驱动架构、CQRS，还有对贫血模型和富领域模型的权衡等等，这些你都会在课程里讲到。

### 第三，是补足面向对象基础

前面说过，《DDD》原书的作者，其实是假设读者已经有了一定面向对象方法学的基础，比如说，分析和设计的基本理念、UML 的使用、模型到代码的转换等等。不过学习这门课，不需要你系统地掌握面向对象方法学。我会在讲解过程中，结合 DDD 的知识点，详细介绍必要的面向对象知识和技能，帮助你扫清学习障碍。

### 最后，是面向实践，避免空谈

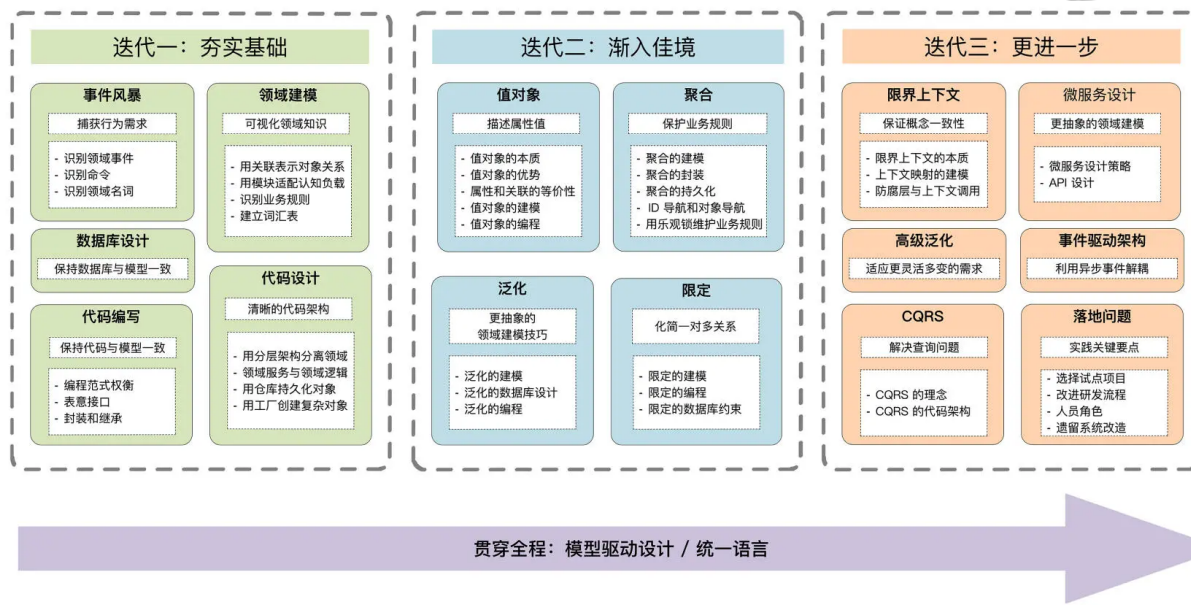
这门课基本上都是按照“问题 - 方案”的模式。也就是说，首先由需求引发要解决的问题，然后找到解决问题的知识点，并进行讲解。每个知识点都有具体的建模或编码实现来说明，避免空谈理论。

### 写在最后

和很多 IT 技术一样，DDD 不是纯理论的，而是一门“手艺”，必须动手练习才能真正学懂。

你可以跟着我建模型、写代码，拾级而上。最终，让你在心里潜移默化地建立起完整的知识图景，掌握 DDD 最核心的技能。如果可以在自己的项目里用起来，感受应该会更加深刻。





知识地图

当你学到课程中间部分的时候，就可以读一下《DDD》原书来加深理解，还可以通过其他参考书扩展视野。比如说，如果想深入学习 UML，可以读一下《UML 用户指南》和《UML 精粹》；想了解真正复杂的领域模型长什么样，可以看看《分析模式：可复用的对象模型》等等。另外，你还可以把自己学习到的知识，讲给团队的小伙伴听，达到教学相长的目的。

最后，我想说的是，DDD 不仅是一门技术，更是一门艺术。而你既然想学习 DDD，恐怕也不仅仅是为了应付某个具体的问题，而是多少带着一些情怀来的。你可能想要了解，几十年来，那些研究软件开发方法学的大师们，到底总结出了怎样的经验；也可能希望发扬工匠精神，成为一个技艺卓越的软件开发着。

那么，就让我们一起，开始这个充满乐趣的探索之旅吧。

分享给需要的人，Ta 购买本课程，你将得 18 元

生成海报并分享

## 精选留言 (11)

写留言



南山 置顶

2022-12-08 来自广东

搞了这么久的DDD，自己可能理解也不够深入，也是感觉道理都懂，落地也大概知道怎么落地。

还有一点比较深的体会是DDD不是一个人就能搞起来的，而是各个角色（产品、业务、研发、测试）都要参与进来才具备落地的基础，如果是这样又会面临如何能让这些角色积极参与进来的难题，可能很多公司根本就具备这个条件。

自己也始终相信这条路是对的，就是开头比较难，只能选择性的应用，扩大影响力，先营造好环境，再谈落地实践

作者回复: 后续课程会有一些思路。不过实践中这类问题确实不容易解决。正如您所说的，万事开头难，共勉！



2



业余草

2022-12-07 来自广东

DDD 就和懂道理一样，自以为懂了 DDD（道理），实际上还是驾驭不了。

正如我们「听过很多道理，依然过不好这一生」一样。

究其原因，就是软件设计没有银弹！

DDD 诞生这多年以来，多数人学会的只有一样东西，那就是“四层”代码结构：Interface、Application、Domain、Infrastructure。

注意，我并不是否定 DDD！

作者回复: DDD里面确实有很多道理，但也有很多实操落地性的东西。但是由于种种原因，导致现在大家觉得DDD不好落地，这正是我们做这门课的起因。您继续学下去，看看能否达到我们这个初衷：)



3



方勇(gopher)  
2022-12-06 来自广东

老师你好，项目将采用什么语言编程呢



天下无鱼  
<https://shikey.com/>

作者回复: 你好，用的是Java。

共 2 条评论 >



3



Jxin  
2022-12-06 来自广东

36讲要讲清楚这些，难度颇大啊。  
与钟老师一起学习。

谁都可以抽象，但只有智者知道在哪里停下。

希望学完我们都能在度的把握上更进一步。

作者回复: 希望我们能借此机会共同提高。关于“度”的把握，正是架构师思维的体现，这个必须赞



2



aoe  
2022-12-06 来自广东

一起探索

作者回复: 路漫漫其修远兮：)



1



Geek\_d1bbb0  
2022-12-06 来自广东

老师您好，这里提到的领域建模，是不是针对业务建模。比如用统一建模语言UML画用例图，活动图，时序图，状态图等。

作者回复: 你好！是的，是针对业务建模，传统上也叫概念模型或分析模型。主要用的就是UML，其中最重要的还是类图，必要时再结合您说的那几种图。

共 2 条评论 >



1



张申傲  
2022-12-10 来自北京



一直很想好好学习下 DDD，恰好遇到了这门课程。请问老师有具体的项目源码吗？



天下无鱼

<https://shikey.com/>



李威

2022-12-08 来自广东

提出一个问题，给出一个解法，这种接地气的教学方式不要太赞，期待，搞起来

作者回复: 谢谢鼓励，多提宝贵意见！



杨迪

2022-12-07 来自广东

老师您好，会有领域模型到代码的案例吗

作者回复: 后续课程会有的。



请叫我和尚

2022-12-07 来自广东

老师想请问一下，课程内容的迭代是否会从传统的 MVC 架构，然后一步一步教你如何去优化到分层架构，这样由浅入深的授课内容。

作者回复: 是由浅入深一步一步来。首先介绍需求，然后梳理需求，领域建模，再根据模型写代码。在写代码之前，会详细介绍分层架构。不过课里没有提MVC。直接讲分层架构，应该就能看懂。如果你很关心DDD分层架构和MVC的关系，我们可以找另外一个时间专门说说。说不定，您心目中的MVC和我心目中的不太一样。



escray 

2022-12-06 来自广东

看了开篇词，再加上 Thoughtworks 的背书，感觉这个专栏还是比较有吸引力的，打算跟着专栏更新的节奏学习一下。极客时间之前还有一个《DDD实战课》的专栏，可以互为参照。

作者回复: 欢迎学习，有问题一起讨论哈。



