

代码架构的风趣解读，项目开发的实际操练

几个并肩战斗的兄弟，共同完成一个软件项目的开发

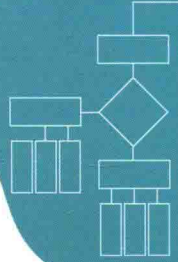
看似拉家常式的对话，实则蕴含着解决问题的智慧

◇ 于身临其境之中，轻松愉快地做一套软件开发的思维操练

项目实战版

代码架构

田伟 郎小娇 著





田伟 郎小娇 著

图书在版编目 (CIP) 数据

大话代码架构: 项目实战版/田伟, 郎小娇著. —北京: 机械工业出版社, 2017.7

ISBN 978-7-111-57701-0

I. 大… II. ①田… ②郎… III. C语言—程序设计 IV. TP312.8

中国版本图书馆CIP数据核字 (2017) 第194024号

大话代码架构 (项目实战版)

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 欧振旭 李华君

责任校对: 姚志娟

印刷: 中国电影出版社印刷厂

版次: 2017 年 9 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 20.5

书号: ISBN 978-7-111-57701-0

定价: 69.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

内容简介

你是否还在为了“三层”而“三层”？甚至不知道为什么分层。

你是否还在为项目中难以维护的SQL语句而头疼不已？

你是否还不知道MongoDB这样的非关系型数据库也可以使用ORM框架？

你是否对前端或测试这些领域也一无所知？

你是否面对高并发时手足无措？

.....

本书将以轻松幽默的风格解决你所面临的这些问题。

本书以一个真实的项目案例——“晋商卡”从无到有的整个开发过程为主线，用大白的语言风格，风趣幽默地讲解了代码架构的相关知识。本书通过5个人物角色，模拟实际的项目开发过程，以对话形式抛出问题，然后解决问题，让你在身临其境中轻松地掌握代码架构的知识。

本书涵盖的主要内容有敏捷开发的方法论、项目开发流程、传统的三层架构、源代码管理、几种常见的实体关系模型、使用IoC和接口、使用缓存和静态页面减少服务器压力、在项目中使用消息队列、尝试使用前端框架、微信公众号开发及小程序开发。

本书适合对代码架构感兴趣的初学者和爱好者阅读。另外，高校学生和参加软件开发的培训学员也可将本书作为兴趣读物。对于初入职场还比较迷茫的程序员，本书可以作为一本提高读物来阅读。建议阅读本书的读者具有一定的C#语言基础。

作者简介

田伟

自称MOL。长期从事软件开发及团队管理工作。擅长代码框架的搭建和优化。善于将敏捷方法论用于项目开发中，从而提高团队的开发效率。坚持一个原则，即自己认为好的东西应该分享给大家。提倡软件开发不仅需要“工匠精神”，也需要“懒人”精神。喜欢以幽默风趣的语言风格讲述技术问题，并以此风格著有《ASP.NET入门很简单》一书，颇受读者好评。



郎小娇

毕业于北京工业大学。现任职于某著名互联网公司，任产品经理。对方法论有独特的见解，尤其对敏捷开发的方法论颇有见解。经常把“哲学思维”用于工作。善于使用不同的方法论指导项目成员的工作，规避项目的风险。曾主导过某大型购物网站的架构工作，以及主要模块的设计实现。



欢迎IT领域的各位技术牛人洽谈出书事宜。如果有写书或投稿意向，请加QQ或者微信具体商谈。

QQ: 627173439

微信: oyzx_sp

序言

2017 年是不平凡的一年。

时隔 4 年，Nokia 终于带着情怀回归了。

苹果也迎来了 10 周年纪念。

微信小程序对个人用户开放了。

2017 年是一个非常强调“工匠精神”的一年，但是 MOL（即笔者本人）在本书中强调的是“懒人精神”。不管你承认与否，所有的人都希望自己能不劳而获。当然，这只是一个美好的愿望。MOL 只能教大家做最少的事情来赚取更多的休息时间及陪伴家人的时间，这就是我所谓的“懒人精神”。

有些读者可能好奇为何笔者给自己起了 MOL 这个奇怪的笔名。关于这个问题，笔者在 2014 年出版的《ASP.NET 入门很简单》一书中有过交代，有兴趣的读者可以去看看那本书。

凡属过往，皆为序章。

写《ASP.NET 入门很简单》的时候，MOL 刚刚结婚。在写本书的时候，MOL 已经有了幸福的三口之家，宝宝已经可以通过一些简单的词汇来表达自己的情绪和意愿，并且还会跟 MOL 抢键盘。我的妻也在为这个幸福之家努力奋斗。想想自己真是幸运。虽然本书写得艰苦，家庭工作琐事也繁多，但是为了自己的这份幸运和广大期待本书已久的“摩丝”（MOL 的粉丝），即使再艰苦，MOL 都没有放弃。

所谓更牛，只是换个“罪”受。

作为一个技术宅男，MOL 更愿意每天只对着计算机写写代码就可以完成自己养家糊口的任务。理想总是那么丰满，而现实又是如此骨感。对于一个职业程序员来说，MOL 的经历还算比较丰富。记得图书市场上出版过一本《不想当厨子的裁缝不是好司机》，后来这个有点无厘头的书名成了一句经常被人引用的调侃语。在此 MOL 也想把这句话改改，和朋友们说“不想当程序员的艺术家不是好魔术师”。非常幸运，这几种职业 MOL 都做过，也希望读者朋友们的职业经历丰富一些。

在我带领自己的技术团队做项目的时候，经历过痛苦，也经历过欢笑。我一直都觉得自己非常幸运，因为在本书中出现的刘朋、岳鹏辉、李冲冲他们 3 个人，悟性非常高，而且颇有自己的见解。在征得他们的同意后，他们将以真实名字在本书中出现。

MOL 不是大牛，只是愿意把自己的经历与更多的人分享。所以，本书中并没有讲

解非常高、精、尖的技术，而是带领大家走进了“懒人”的世界。每个程序员都会进入迷茫期，不知道自己要干什么。所以希望本书能从另一个也许大家从未思考过的角度给大家一些启发。

从你翻开本书的第一页开始，MOL 相信你已经准备好换一种“受罪”的方式了，那你离“更牛”也就不远了！

最后，MOL 要响应习主席的号召，撸起袖子，加油干！对于 MOL 和大多数的“摩丝”来说，恐怕我们要脱掉秋裤，加油跑了！

先给自己定一个小目标，今年，2017 年，我要成为一个“懒人”！

田伟

前言

架构 (Architecture) 是什么? 可能每个人给出的答案都不同。业界流行一句笑话:

Architecture is like teenage sex, everybody talks about it, nobody really knows what it is.

当然, MOL 也不可能给出一个关于架构的准确定义。MOL 更愿意把架构归为哲学的范畴。架构本身其实和软件开发并无太大关系。一个国家有自己的管理体系; 一个公司有自己的组织架构; 一个家庭也有独特的男权或女权的特色, 小到一个人; 也是可以分为自我、本我和超我的。每个事物都是由一个个更小的事物组合而成的, 而这些其实都与架构相关。

在宏观世界里, 所有的国家公民构成了一个国家主体, 国家主体对每个公民进行管理和约束, 这是架构。

在微观世界里, 电子绕着原子核高速转动, 始终不会脱离原子核的管辖范围。而原子核和电子又组成一个原子。原子对电子、质子、中子的管理就是架构。

本书的读者一定是软件行业的高手或菜鸟, 那我们就回到软件编程的世界里。

每个软件项目都是由代码和服务器构成的, 如何统筹安排代码和服务器, 就是架构的范畴了。

一个项目可能要使用多台服务器, 如 Web 服务器、数据库服务器、文件服务器、CDN……如何针对不同的要求对服务器进行选型, 这是架构; 如何统一管理这些服务器, 这是架构; 如何让这些服务器平稳运行, 这也是架构。

开发项目使用什么语言, 是 Java 还是 Node.js? 选用什么数据库, 是 Oracle 还是 MongoDB? 这是架构。

具体到开发过程中, 某个模块应该如何安排, 是交给 DBA (数据库管理员) 用存储过程来实现, 还是让 C# 程序员访问数据库实现? 这是架构。

在写 C# 代码的时候, 采用三层架构, 还是 MVC? 这是架构。

如何写日志, 是使用 I/O 读写文件? 还是采用 log4net? 或者是 AOP 切片写日志? 这是架构。

甚至具体到某种技术的时候, 也有架构。比如 MOL 规定项目要使用 MVC 架构, 那么使用微软的 MVC, 还是 Spring 的 MVC? 这都是架构。

可见, 架构涉及的范围非常之广。弱水三千, MOL 只给一瓢。本书将从代码架构的角度来让大家一窥架构的真面目。

C#是一门非常优雅的编程语言（当然 MOL 并无编程语言的偏见），所以本书中所有的代码都以 C#语言来描述。

本书特色

1. 风趣幽默

MOL 一直比较反对平铺直叙的讲解方式，所以本书的语言风格是比较幽默的。在本书的内容中将出现 3 个与 MOL 并肩作战的兄弟（公司老大邓总不在此列），以对话形式抛出问题并解决问题。

2. 案例分析

本书中只有一个项目“晋商卡”，但 MOL 会带着大家见证“晋商卡”从无到有的过程，大家可以在这个过程中获得很多意想不到的收获。

3. 向循规蹈矩说NO

正如 MOL 在结语中所说，2017 年是一个强调“工匠精神”的一年。几乎所有的人都在精益求精地做自己的事情。但 MOL 要分享给大家的是一种懒人精神，我们不愿意日复一日地重复昨天的自己，我们要站在更高的层面，做更少的事情，却有更多的收获。

本书内容及体系结构

第1篇 需求与三层架构（第1~3章）

本篇详细介绍了项目开发的前置节点——需求，并对常见的三层架构给出了分析。在第 1 篇中提出了面向对象的重点概念，并让大家初步认识抽象的过程。

相信很多人一定被书中大段的 SQL 代码搞得云里雾里，不用担心，在第 3 章中 MOL 将带领大家完成懒人的第一步——如何不写 SQL 代码。

第2篇 NoSQL和测试（第4、5章）

NoSQL 是现在比较流行的一个话题和技术。在第 4 章中将通过讲解 MongoDB 来介绍 NoSQL 如何使用，并且纠正大家的一个错误观念：NoSQL 和 ORM 不能搭配使用。

第 5 章分享了测试的工作，并讲解了单元测试、黑盒测试、白盒测试……让大家在收获的同时，也能理解测试工程师在工作中所要面临的一些痛苦。

第3篇 高精尖技术（第6~9章）

任何一个网站项目，似乎都绕不开“缓存”这个神奇的空间。缓存用得好，可以加快系统的反应速度。如果缓存用得不好，不仅用户体验差，还可能造成服务器宕机。第6章就分享了如何使用缓存。

每个程序员都有一个全栈的梦想，而前端又是全栈中必不可少的一部分，第7章讲解了如何使用 EasyUI 来搭建前端。

现在越来越多的电商网站都会做一些抢购或促销活动，当然这就使得网站不可避免地面临高并发。如何处理高并发呢？第8章将通过讲解消息队列，来说明如何应对高并发。

微信已经成了人们生活中必不可少的一部分。在2017年3月27日这一天，微信小程序也对个人用户开放了。我们如何把“晋商卡”挂到微信公众平台上，又如何开发微信小程序呢？这些问题都将在第9章中解决。

本书读者对象

- 对代码架构感兴趣的初学者；
- 对代码架构感兴趣的爱好者；
- 高校学生和相关培训学校的学员；
- 初入职场需要提高开发水平的开发人员。

因为书中所有的代码都以 C#语言来描述，所以本书读者如果有一定的 C#语言基础更佳。

本书配套资源及获取方式

为了方便读者高效地学习，本书特意提供了以下配套资源：

- 本书源代码文件；
- 本书涉及的一些开发工具的安装包。

这些配套资源需要读者自行下载。请读者登录机械工业出版社华章公司的网站 www.hzbook.com，然后搜索到本书页面，按照页面上的说明进行下载。

本书作者

本书主要由田伟（就是笔者 MOL）和郎小娇主笔编写。其他参与编写的人员还有李小妹、周晨、桂凤林等。

读者阅读本书时若有疑问，可以发邮件到 hzbook2017@163.com 以获得帮助。

目录

序言
前言

引言	1
一个架构师的例子	1
背景及人物介绍	3
我们的目标	3

第 1 篇 需求与三层架构

第 1 章 故事从一个电商网站开始	6
1.1 需求? 需求!	6
1.2 敏捷开发简介	8
1.3 UI——用户界面	9
1.3.1 从 UE 说起	9
1.3.2 HTML 5 & CSS 3	9
1.3.3 微信	10
1.4 数据库	10
1.4.1 PowerDesigner 设计工具	11
1.4.2 关于 SQL 语句	12
第 2 章 为什么是三层	14
2.1 MOL 带兄弟们去吃饭	14
2.2 动手写一个三层结构	16
2.2.1 DAL 层的实现	17
2.2.2 BLL 层的实现	18
2.2.3 UI 层的实现	19
2.3 简说 MVC	20
2.3.1 纠正一下老师的说法	20
2.3.2 MVC 的第一个程序	21
2.3.3 为什么要用 MVC 之我见	25

2.4	向三层代码中加入面向对象	25
2.4.1	实例化数据库表	26
2.4.2	增加数据库操作类	29
2.4.3	加强版的数据库操作类	30
2.5	小说代码管理	33
2.5.1	什么是代码管理软件	34
2.5.2	TFS 管理软件	36
2.5.3	SVN 管理软件	40
2.5.4	Git 管理软件	41
2.6	小结	51
第 3 章	ORM 实体关系映射	52
3.1	说说 OCP 开放封闭原则	52
3.2	解耦第一步——接口要上位	53
3.2.1	代码结构的前提	53
3.2.2	创建接口层	56
3.3	解耦第二步——工厂模式解决 new 的问题	58
3.4	Spring.NET 横空出世	61
3.4.1	酒文化发展史	61
3.4.2	神奇的 IOC	62
3.4.3	引出 Spring.NET	64
3.5	我不想写 SQL 语句	67
3.5.1	什么是 ORM	68
3.5.2	ORM 之 iBATIS.NET	70
3.5.3	ORM 之 NHibernate	75
3.5.4	ORM 之 EF	86
3.5.5	懒人无敌	93
3.5.6	完成查询操作	106
3.5.7	数据库先行、模型先行、代码先行	109
3.6	小结	110

第 2 篇 NoSQL 和测试

第 4 章	换个数据库试试	112
4.1	客户总有一些非分的想法	112
4.2	MongoDB 简介	113
4.2.1	安装&配置	114
4.2.2	可视化工具	116
4.2.3	MongoDB 的基本操作	119

4.3	.NET 操作 MongoDB	121
4.4	让 NoSQL 面向对象	123
4.4.1	实体抽象	124
4.4.2	操作抽象	132
4.5	NoSQL 题外话	135
4.6	关于日志	136
4.6.1	为啥要写日志	136
4.6.2	如何写日志	138
4.6.3	注意事项	144
4.7	小结	145
第 5 章	越俎代庖搞搞测试	146
5.1	简单说测试	146
5.2	冒烟测试	147
5.3	黑盒测试	148
5.4	单元测试	149
5.4.1	单元测试是什么	150
5.4.2	如何进行单元测试	150
5.4.3	测试驱动开发	161
5.5	白盒测试	162
5.6	压力测试	163
5.7	其他测试	163
5.8	小结	163

第 3 篇 高精尖技术

第 6 章	神奇的缓存	166
6.1	网站崩溃了	166
6.2	缓存是什么	168
6.3	微软提供了缓存	170
6.4	自己做缓存	173
6.5	第三方缓存	176
6.5.1	MemCache 缓存	178
6.5.2	Redis 缓存	189
6.5.3	把缓存抽象出来	197
6.5.4	读写分离	198
6.6	利用模板引擎生成静态页面	198
6.6.1	初识 NVelocity	200
6.6.2	使用 NVelocity 生成静态页面	207

6.6.3 静态页面带来的问题	211
6.7 CDN 的加入会大大减少服务器的压力	212
6.8 小结	213
第 7 章 程序员眼中的前端	215
7.1 常见的前端框架	216
7.2 尝试 EasyUI	220
7.2.1 基础框架	220
7.2.2 常用的控件	232
7.2.3 小说权限分配	247
7.3 其他的前端框架	264
7.3.1 Bootstrap 框架	264
7.3.2 jQuery UI 框架	265
7.4 小说 HTML 5	265
7.5 小结	266
第 8 章 人生中的第一次高并发	267
8.1 网站又崩溃了!	267
8.2 从相亲说起	268
8.3 简述消息队列	270
8.3.1 串行设计	270
8.3.2 多线程设计	271
8.3.3 消息队列设计	274
8.4 常见的消息队列	275
8.4.1 MSMQ 消息队列	275
8.4.2 RabbitMQ 消息队列	282
8.5 小结	288
第 9 章 微信公众号	289
9.1 李老板出场, 请热烈鼓掌	289
9.2 初探微信公众号	290
9.2.1 申请订阅号	291
9.2.2 小机器人	294
9.2.3 把晋商卡挂到微信公众平台上	298
9.3 微信小程序	299
9.3.1 微信小程序是什么	299
9.3.2 写一个简单的 Demo	300
9.4 微服务	310
9.5 MongoDB 数据库	312
9.6 大数据	314
9.7 小结	316

引言

我叫 MOL，如果你是 MOL 的读者，那么一定知道“摩丝”了。MOL 者，“摩尔”也。摩丝者，MOL 的粉丝也。MOL 在本书里将带领大家一起做一个属于自己的代码架构。代码架构和架构是一样的吗？且看 MOL 如何分解。

一谈到架构（Architecture），大家一定会觉得它是一个非常“高大上”的东西，当然，大部分人都是这样宣传的。为了让大家有一个更好的感性认识（因为本书并不是讲架构的，所以只要有感性认识就可以了），MOL 决定用一个简单的例子来告诉大家什么是架构。

一个架构师的例子

在 2015 年的时候，MOL 主导了一个 B2C（Business-To-Customer，商家对客户）网站，MOL 在这个项目里面充当了非常多的角色，如产品经理、项目经理、架构师、DBA（数据库管理员）、程序员、QA（质量管理员）、Tester（测试员）……现在来看一下 MOL 作为架构师时所做的事情。

MOL 作为架构师，是以大体需求为前提的，也就是说，我们在这里不去讲如何获取需求，因为这不是架构师份内的事。当 MOL 拿到需求以后，就可以进行架构了。架构大体上分为两部分：硬架构和软架构（这并不是标准的叫法，只是想让大家更好地理解架构）。

1. 硬架构

顾名思义，硬架构就是关于硬件的架构。MOL 根据目标用户量和业务要求，采购了 3 台服务器，分别作为文件服务器、数据库服务器和 Web 服务器（同时兼任缓存服务器），同时，建议用户在客户量增加到一定数量级以后，增加 CDN（Content Delivery Network，内容分发网络）服务器，以加快访问速度。

OK，架构师的输出已经完成了。但大家以为架构师就买几台服务器就完事儿了吗？那就大错特错了。在买服务器的背后，MOL 对需求进行了研究，分析了网站需要承担的平均访问量和网站的业务内容。这个网站的访问量并没有大到令人发指的地步，所以我们暂时不用考虑硬件负载均衡的问题，只需要提供一台 Web 服务器就可以了。这台 Web 服务器需要处理用户发来的请求并做出响应给用户。为了提高 Web 服务器的性能，我们将

会在这台 Web 服务器上安装虚拟机，并用 Nginx（一个 HTTP 服务器，类似于 IIS）做软负载均衡。

由于本系统中需要保存大量的用户文件，如果把这些文件都放在 Web 服务器上，那会给软件负载均衡产生额外的负担，并且大量的文件很容易把一台服务器“撑爆”。所以，我们用一台服务器专门来保存文件信息，这台服务器就叫文件服务器。

数据库是一个项目中必不可少的一部分，数据库的本质其实就是文件和内容的组合，所以数据库的空间增长也是不可小觑的。而且数据库的操作，将会耗费大量宝贵的 CPU 和内存资源。所以我们将数据库专门放在一台服务器上，这台服务器就叫数据库服务器。

好，硬件资源采购完成了，我们要把它们组合到一起。Web 服务器向网络公开，让用户可以访问，而文件服务器和数据库服务器只在局域网中，并不对外公开，这样可以在一定程度上保证安全性，而且也为客户节省了费用。Web 服务器可以访问文件服务器和数据库服务器。

其实硬架构是非常复杂的，但本书不是主要讲硬架构，所以这里讲得非常简单，有兴趣的读者可以自行找“某度”或“某狗”进行搜索。

2. 软架构

本系统将采用 .NET 平台下的 C# 语言进行开发，采用 SQL Server 数据库，使用微软的 Cache 缓存配合 Redis 来做。前台将以 3 种终端展示，分别是 PC、手机、平板。所以前端将会使用 HTML 5 来做。Android 手机 APP 使用 Java 语言来开发，iOS 平台上的 APP 暂时不做。

同样的，软架构也只是让大家看到了输出，并没有讲为什么这样做，因为这也不是本书的重点。

上面粗略地讲了一下什么是架构。下面来具体看一下代码架构。废话不多说，直接来看图 1 和图 2。



图 1 正常的代码架构



图 2 不普通、非文艺的代码架构

图 1 是一个常见的代码架构，而图 2 可能是新手程序员最喜欢的“代码结构”。代码架构的目的是让不同的代码块去干不同的事，最后再把这些代码块整合在一起，组成一个项目。这是本书要讲的内容。

可以看到，代码架构和软件架构基本上不属于同一个层次元素。但不可否认的是，代码架构是一个程序员应该具备的技能，也是通往架构师的必经之路。

⚠PS: 各位读者可能对用户和客户这两个概念理解有误，这里统一一下，用户（User）是指使用系统的人，客户（Customer）是指要求我们做这个系统并且支付开发费用的人。

背景及人物介绍

本书中，我们将延续 MOL 活泼幽默的风格。为了剧情的需要，书中将会出现 5 位主人公。

MOL: 姑且称之为老鸟吧，从事代码工作多年，在公司中主要负责项目管理和代码架构。

邓总: 公司老大，身材消瘦如马云，有一双深邃的眼睛。虽然是公司的老大，但在本书中出现的次数并不多。

刘朋: 新入职员工，因为名字中的“朋”字“占地”面积很大，所以大家一般都会把“朋”字分开来念，所以他有另一个名字“月月”，性格比较幽默。

岳鹏辉: 新入职员工，长得比较帅气（已有女朋友）。

李冲冲: 新入职员工，打得一手好乒乓球。

这几位主人公将会在本书中出现，他们可能是说相声的，可能是打酱油的，也可能是回家跪搓衣板的。但是他们更重要的作用是构造一个个生动的故事，在这些故事里面，大家可以看到一个真正的项目是如何搭建的。

故事来源于生活，却不高于生活，这里的人是真实的，故事是真实的，当然，最重要的是，项目是真实的，知识也是真实的。MOL 尽量用最接地气的语言，最平实的故事，来讲述一个个“高大上”的知识和技术。

我们的目标

我们经常听到这样的话：“加班不是目的，目的是不加班”。这样的话让人无比窝火，但又无比正确。我们还会听到这样的话：“我这样严厉地要求你，是为了让你成长更快”。你有理由反驳吗？其实，换一种方式，我们可以快乐地成长。这也是 MOL 带团队的宗旨，也是本书的目的。MOL 要让你快乐地看完这本书，最后发出的感慨一定是：“代码架构其实挺简单！”

好了，我们要开始了！