

## 博客园**开发者**征途

**“给我一个支点，我就能撬起地球”**

关键在于力量有多大，而在于如何合理地利用力量。

### 软件设计同样如此

思想的确立，技巧的把握，将在很大程度上制约软件架构的合理性。本书力图通过大量的项目实践，并与软件设计思想完美结合，充分展示软件设计之美，以及设计“力量”的巧妙运用。

# 软件设计

(第2版)

## 精要与模式

张逸  
著



博客园**开发者**征途

# 软件设计 精要与模式

(第2版)

张逸 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书关注的焦点是软件设计，涵盖了大部分与设计有关的基本要素，包括面向对象思想、设计模式、重构、测试驱动开发以及软件体系架构。其中，尤以设计模式为主，深入探讨了软件设计过程中的原则与模式，并结合大量的实例与代码演示了如何合理运用设计模式，改善程序模块的可复用性、可扩展性，实现模块间的松散耦合。全书将软件设计理论与项目实践完美地结合起来，使其告别了纯理论研究的空泛，具有现实的指导意义。本书共分为5篇，包括：设计之要、.NET 框架与设计模式、媒体播放器的设计之旅、设计模式应用实践以及.NET 体系架构。

本书力求讲解浅显明白。在技术探讨上，尽可能地深入透彻；在每一字的描述上，尽可能地简单易懂。本书适用于所有希望提高软件设计水平的程序员、软件工程师，同时，对于软件设计师与系统架构师也具有一定的参考价值。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有，侵权必究。

### 图书在版编目 (CIP) 数据

软件设计精要与模式 / 张逸著. —2 版. —北京: 电子工业出版社, 2010.4

(博客园开发者征途)

ISBN 978-7-121-10453-4

I. 软… II. 张… III. 软件设计 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2010) 第 032026 号

责任编辑: 胡辛征

印 刷: 北京智力达印刷有限公司

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 30 字数: 720 千字

印 次: 2010 年 4 月第 1 次印刷

印 数: 4000 册 定价: 55.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及订购电话：(010) 88254888。

质量投诉请发邮件至 [zits@phei.com.cn](mailto:zits@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线：(010) 88258888。

## 推荐序一

张逸先生邀我为他的新著做序言，起初觉得难以应命。毕竟，一本书会成为很多人阅读学习的材料，并逐渐沉淀为社会文化的一部分而影响长远。长年的工程习惯告诉我，应该先认真阅读书稿，并且深刻理解书中鲜明的思想和观点后再下笔。但就个人目前的工作及精力，深感不能追求到如此完美的情景。然而，又意识到软件架构与设计工作对整个中国行业发展的重要和紧迫程度，决定借写序为契机，谈点关于架构和设计方面的点滴体会，作为本书的书序。

少年时代的我，充满了对科学的向往。儒勒·凡尔纳的科幻小说，把我带向了科学梦幻的世界，仿佛科学能够创造出美妙的未来世界。后来，对天文及天体物理的迷恋，把我强烈地吸引到了对伽利略、牛顿、爱因斯坦等伟人的崇拜。可笑的是，原想报考南京大学天文专业的我，被父母当头浇了一盆凉水。但是，一颗热爱科学的心一直在跳动。

整个大学的前两年，听课一直浑浑沌沌。直到大学三年级的时候，听了一位教授讲的数据结构课程，可以算是开启了我对计算机科学最初的认识。这是我第一次感知到计算机科学在很大程度上是研究人类智慧的学科，这也正是年轻的我所热望的专业！

毕业后，由于在国营单位这样的圈子中工作，又经历了一段浑浑沌沌。1992年后，面向对象的 Borland C++ 及 Turbo C++ 开始在世界乃至中国大陆范围内流行。半生半熟地阅读完这种全新的编程思想，仔细体会一番，又一次为人类智慧的结晶而震撼和赞叹。原来结构化的编程思想，虽然出自自然，但并不一定就是最好。人类居然可以模仿自然规律，来界定一个个关联的对象，可谓聪明和经典。

九十年代，是一个出国潮涌的时代，我也随着潮流，漂洋到北美。从那时开始，有两件事，真正把我从一个懵懵懂懂的年轻人，带到了计算机科学的智慧天堂，从而满足了专

业工作人员的第一个要求，即知识的储备。

第一件是把自己所从事的研究工作，定位到了状态依赖的系统。这个方向的研究，仿佛打开了一扇大门，让我从只知道传统计算机科学的基础知识，加上有限的编程经验，真正地走向了专业知识的研究工作。进而使我深刻理解了国外为什么能够领先中国很多年，就已经能研发出很多严重状态依赖的实时系统。这也是我平生第一次，从软件系统的结构上，知晓了人类智慧的创造力。

另外，我在这个阶段中也自然而然地接触到当时刚刚开始流行的 Java 这样相对纯净的面向对象编程语言所设计出来的一些系统。也很自然，工作中面对一个资深设计编程人员所设计出的模块结构和编写出的代码，科学之美的情感油然而生。期间，作为一个中国人，开始经常听到“架构”和“设计”这样两个有些陌生的词汇。最令我难忘的是，有位非常友善的同事，甚至还指导我去阅读一些有关架构和设计方面的知名著作。我也是从这个时刻开始，知道了 Gang of Four 的设计模式、Frank Buschmann（日后服务于西门子时，我的德国业务领导人）的架构和设计模式、Martin Fowler 的著作……遗憾的是，由于当时自己所处工作环境的限制，没有能够更深入地体验出更多的东西，也没有一个合适的场合锻炼一下自己。庆幸的是，我已经比很多中国人早一些读到了一些经典的著作，学到了一些知识。

混混沌沌的我，在 2003 年年底举家回到了祖国。当时的中国，正处于 IT 革命所带来的一片欣欣向荣的环境当中。由于是所谓的海归，自然有机会在这样的系统研发浪潮中冲锋在前，把自己一知半解的所谓经验使用到现实的系统研发工作中。着实轰轰烈烈地实践了一轮，却发现自己又一次迷失了：理论学习过了，实践也经历了，我该走向何方？

糊里糊涂地，无意间读到了一篇纪事报告，题目叫《最后的大师》。此文的作者应钱学森先生的邀请，来记录自己的导师、清华大学物理系及清华大学创始人之一的叶企孙先生。叶先生早年在美国留学期间，在物理方面做出过杰出的贡献。虽然大多数后人并不知道叶先生，但是他的学生没有一个会忘记他，这包括三钱、华罗庚、李政道、杨振宁等。可以这样说，你所知道的中国大师，大多都是他的学生。阅读完此文，颇受启发：真可谓“大师培养大师”。我非常欣赏这句话。既然我身边没有大师，就应该认真回味一下自己这些年来的学习和实践，看看是否能将既有的种种知识和经验，上升为智慧。毕竟，智慧是指导我继续工作的原始动力，并指导自己未来的创新工作。因此，我选择了阅读、学习和思考。

自己成长的这段历史，算是翻过去了。再回到张逸先生的这本《软件设计精要与模式》上来，我虽作粗略阅读，但从实践分享的视角来看，书的内容编写得非常认真。作者从自身工作的经历，分享了自己对软件设计的理解，并以设计原则这样的方式，来分享最宏观层面上的要点。总结、思考的分量，可见一斑。本书有些章节很有新意，注意到了利用自身实践过的设计模式，以真实示例的方式来介绍如何灵活使用各种设计模式。此举对读者的实际工作，颇有帮助，愿为推荐。

其实，个人成长的历程，也在一定程度上代表了中国专业从业人员的成长轨迹。中国正面临一次深刻的变革，需要更多优秀的编程人员、优秀的设计人员、优秀的架构人员、优秀的创新人员。毕竟，一个要立足于世界之林的强国，迫切地需要能把事情做得精彩和经典的行动人员。

谨记所感，提供讨论。

李伟

西门子中国中央研究院 首席架构师

2010年02月12日

北京

## 推荐序二

在 InfoQ 中文站 .NET 社区首席编辑朱永光介绍张逸,并希望邀请其加入编辑团队之后,我特别在网上找到张逸的相关资料研读,并粗略翻阅了他的著作《软件设计精要与模式》及译著《WCF 服务编程》。一番考察之后,认为其符合 InfoQ 编辑人员所必备的两个特点:态度认真和技术扎实,然后郑重向其发出邀请。最终,张逸答应了我们的邀请,而我也为能和这样一位优秀的架构师合作而荣幸。通过其在 InfoQ 中文站上发表的作品,我和永光均为当初的决定欣慰,而张逸的作品也让网站.NET 社区的内容更加充实。

很长一段时间以来,与模式相关的话题都特别引人注目。在去年 InfoQ 中文站举办的 QCon 北京大会上,来自知名网站 eBay 的架构师 Randy Shoup 介绍了 eBay 架构设计过程中的经验,场面之热烈,让很多参会者现在依然记忆犹新。在今年 QCon 北京大会上, Jolt 图书大奖获得者 Michael Nygard 会带来他在系统设计过程中的反模式,虽未开场,已有多位朋友表示对此非常期待。诚然,在软件研发过程中,要找到一劳永逸的“银弹”并非易事,但类如“模式”和“反模式”这样的经验总结总会给后来者一些启发。

每个软件系统都有其独特的一面,研发所用的技术或者平台也多有不同,但仔细考察其背后的设计思想,总能发现几丝共性。正是这些共性经过抽象之后,形成模式,然后被后来者不断传承和演进,使得我们的软件系统愈加庞大和健壮。希望通过本书,读者能够了解作者张逸对技术的真诚,也能够帮助自己在软件设计的道路上找到知音和共鸣。

霍泰稳

InfoQ 中文站 (infoq.com/cn) 总编辑

## 推荐序三

很早就在博客园上拜读过张逸的文章，尤其对他在设计模式方面的经验和见解印象深刻。在我把他邀请进 InfoQ 中文站编辑团队后，经过深入而频繁的交流，对他在整个 .NET 方面的学识和实践甚感佩服。尤其佩服于他深厚的文学功底，其能用优雅的文字把生涩枯燥的技术解析得浅显易懂。

理所当然，张逸在这本《软件设计精要与模式（第2版）》中，将用优雅的文字、贴切的比喻、精彩的示例为大家剖析设计的奥妙，分享他在 .NET 方面的丰富知识和经验。第一篇“设计之要”为目前流行的软件设计思想进行了提纲挈领；第二篇“.NET 框架与设计模式”用我们每天工作的基础——.NET 框架——作为最好的例子来讲解重要的设计模式；第三篇“媒体播放器的设计之旅”可以说是设计之要的完整实战演示；第四篇“设计模式应用实践”用一些与工作息息相关的实例展示设计模式的妙用；第五篇“.NET 体系架构”则指引我们进入 .NET 应用程序架构的殿堂。

对于架构与设计模式，一直以来也是我最感兴趣的技术领域。而面向对象、设计模式、重构、测试驱动开发、敏捷编程这些思想，在我看来，即是武学中的各种秘籍，指引着我们修炼成为武林高手。但光拿到武学秘籍还不够，光勤学这些秘籍也不够，还需要苦练其中的招数，并在一次次的比武中实践和体会。正如武学的最高境界是无招胜有招一样，软件设计的最高境界也最终是要把本书详细讲解的“招数”悉数忘记，让这些招数成为自己的习惯、自己的思维，在设计过程中自然而然地融会贯通——当然，这不是一朝一夕的事情。



通览本书，里面提到的很多思想和见解都能极大地引起我的共鸣。每当如此，常常会感叹自己没有足够时间和精力与大家分享类似的思想。因而，特向大家推荐本书，一方面可以宽慰自己，更重要的一方面就是，很高兴张逸能给大家贡献这样一本软件设计的好书。

朱永光

微软 MVP

InfoQ 中文站.NET 社区首席编辑

2010/3/1 于成都

## 推荐序四

有人总喜欢拿“软件危机”说事——我们总是无法足够快、足够低成本或足够好地开发软件。软件开发确实非常复杂。那么软件成本是否缘于其复杂性？其实，很多时候软件成本高的原因来自于使用了错误的行为来管理软件的高成本。一个架构图可能非常有帮助，而缺乏优秀的设计，软件开发就会陷入混乱。设计模式是针对面向对象设计过程中反复出现的问题的解决方案，必须经过反复分析，才能确保正确性。就如同任何数学方面的分析，你需要图标和公式，而且你需要展示你的工作。软件设计不需要故弄玄虚，而需要务实。

软件设计模式与实践的相关会议每年都会吸引众多业界享有盛誉的大师级人物参加，更有众多来自顶尖软件开发企业的技术决策者与资深架构师参与。由此，软件模式也成为近几年全球软件业最为关注的高端话题，该领域的每一项重大研究成果与突破都会为全球软件产业带来革命性的变化。

回想最初接手 IT168 的 .NET 技术频道时，博览国内众多技术人员网站及博客，张逸先生的博客便跃然眼前。面对这样一个优秀架构师所设计的模块结构和编写的代码，直观地彰显了设计之美，其深厚的技术功底更是不言而喻。之后几年的合作当中，张逸先生的稿件总是优质的，被广大网友所乐道，因此也成为了我的优秀作者。

“设计没有标准，只有目标。如果硬要制定一个标准，那么标准就是快捷、适用与优雅。对于软件设计而言，满足客户需求的设计就是最好的标准！然而，前提是怎样通过设计来满足客户需求？”张逸先生将软件设计理论与项目实践完美地结合起来得出《软件设计精

要与模式》，后来更是精益求精，得出第2版。不要再去苦恼那些理论研究的空泛，相信这本书可以帮助你找到你的软件设计之道。

胡铭娅

IT168 技术频道资深编辑  
微软最有价值专家 (MVP)

## 再版前言

我希望告别冗长的前言，仅述说第2版的变更。写作第2版的我，疯狂地吸收了诸多大师的设计思想，这一点可以从参考文献的前后差别看到端倪。两年以来，我又参与了几个项目的设计与开发工作，所谓“实践出真知”，在佐证大师观点的同时，自己对设计的认识更进了一步。或许，第2版不会比第1版优秀太多，但至少会减少诸多不足。囿于版本，我无法做出新的突破。我期待能创作一本全新的书，全面论述我对软件设计的认识。现在的我，还不足以写出梦想中的软件设计之道。

言归正传。

整体而言，我对第1版的所有章节都进行了一定程度的修订。或者更正了过去的错误，或者进一步完善了原有内容。本书的内容仍然是散漫而自由的，然而形散而神不散，大体遵循了设计的基本原则。

在第1篇“设计之要”中，我新增了“对象法则”一章，言简意赅地介绍了面向对象思想的核心要素与设计原则。这基于我的一贯理念，即设计模式的核心本质是面向对象设计思想的运用。只有掌握了面向对象设计思想，才能真正体会设计模式的精髓，并将其运用在实际的项目开发过程中。“对象法则”一章可以有机地与“封装变化”一章结合起来，再加上第23章“软件体系架构”的内容，基本上勾勒出软件设计的脉络，从面向对象思想到设计模式，再到软件体系架构。

在“封装变化”一章中，我不仅完善了项目实例，还增加了关于如何“解耦具体依赖”的几种技巧。对于软件设计而言，这是非常有益的指导。我整个儿删去了第1版的第5章“设计，由你掌握”，并将其中的部分内容转移到“封装变化”一章中。这使得第1篇的内容更为紧凑，虽然删去了讨论极限编程的相关内容，却可以使得我们能够更加关注于设计，而不是方法学。

第2篇“.NET 框架与设计模式”增加了对.NET 3.X 的源代码分析。我无法做到与时俱进，因为.NET 4.0 即将走进.NET 开发人员的程序生活。或许在本书出版之后的不久，还会有5.0, 6.0……我只是希望我的书不要被时代抛弃得太远。好在设计模式本身属于经典，而经典总是能够经得起时间考验的。本书讲述经典，自然能讨得一定好处。

更新最明显的是迭代器模式在.NET 中的实现。C# 2.0 引入的 `yield return` 以及.NET 3.0 引入的 `Lambda` 表达式都为迭代器模式在.NET 中成为一种惯用法贡献了一份心力。我对此的分析，可以在一定程度上帮助读者更好地理解迭代器模式。在第2篇中，我新增了一章“.NET 中的命令模式”，通过解析.NET 3.0 引入的 WF (Windows Workflow Foundation)，展现命令模式的非凡价值。第2篇的内容虽然与.NET 平台息息相关，但对于其他平台的开发人员而言，仍有可观之处。我在撰写本书第2版时，同样参考了 Java 平台的设计理念，以及 Ruby 中的设计模式。

从章节名称来看，我对第3篇“媒体播放器的设计之旅”进行了颠覆性的革新。事实不然，虽然内容仍有调整，但并未动摇其根本。在对本篇进行修订时，我扮演了一名重构者的角色，利用重命名和搬移内容的方法，极大地改善了既有章节的合理性。我抛开原有的以设计模式为核心的论述方式，转而从软件设计的角度看待问题。模拟真实的软件开发，我讨论了如何运用面向对象设计思想，如何对接口进行分离。当客户需要引入第三方软件时，我提出了接口适配的方案。当需求发生变化时，我则对接口行为进行了扩展或装饰。

第4篇“设计模式应用实践”仍然体现了本书的重要价值。我对第18章“命令模式应用”的实例进行了极大的完善，使得该实例在表现命令模式方面，更加丰富与完整。第19章“职责链模式应用”完全面貌一新，替换为最近完成的一个项目实例，并通过领域进行建模，辅以用例图、时序图、通信图和类图推导详细设计，展现了“用例驱动开发”设计思想的冰山一角。

本书对软件架构着墨不多，主要的架构思想均放在第5篇“.NET 体系架构”中。利用 `PetShop` 实例，对于指导读者初窥架构之美，仍有不可低估的作用。第2版对软件体系架构的内容有所补充与增强，更多地引入了企业应用架构模式和领域驱动设计的内容。第23章“软件体系架构”算得上是技术架构的入门读物，主要介绍了分层架构模式与相关设计要素。在第24章“数据访问层”中，我特别引入了.NET 4.0 中的 `Entity Framework`，算是一次有益的尝鲜。利用这样的 `ORM` 框架，还可以极度方便地实现资源库模式与工作单元模式，在诸多分层架构中，我们都可以看到它们的身影。在第28章“表现层”中，我设想了如何在 `PetShop` 中引入 `ASP.NET MVC` 框架。我本希望能有大量篇幅介绍 `Silverlight`，以及 `MVP` 模式的运用，如此对于.NET 的表现层设计方才显得完整。可惜我对 `Silverlight` 所知不多，心有余而力不足。

第2版还有诸多变化不能体现在目录中。例如，我对本书的全部设计图进行了更新，更加准确、完善和美观，并保持了图形风格的一致性。第2版加入了诸多注解，大多数内容都是正文的补充与扩展，乃至思想点滴。阅读这些注解，可以帮助读者更好地理解我的设计意图，获得更多的模式知识。

本书面对哪些读者？读者又该如何阅读本书？第1版前言已经给出了答案。本书的再版并不打算彻底改头换面。

本书希望能够持续演进，敬请读者提出宝贵意见，并通过 [bruce.e.zhang@gmail.com](mailto:bruce.e.zhang@gmail.com) 联系我。如需获取本书代码，以及了解后续可能给出的勘误与更新，可以访问我的个人主页 <http://www.agiledon.com>。

非常荣幸能够邀请到西门子中国中央研究院首席架构师李伟先生，InfoQ 中文站总编辑霍泰稳先生，微软 MVP 朱永光先生以及 IT168 技术频道资深编辑、微软 MVP 胡铭娅小姐拨冗为我写序。四篇序言各有特色，通过他们的序，可以了解他们的技术心得与心路历程。寥寥千言，或许已胜过我的数十万文字。这样缤纷的文章，无疑为本书增色添彩。

好了，除了致谢，我不打算重复唠叨了。

钱锺书先生认为，献书仿佛魔术师玩的飞刀，放手而并没有脱手。随你怎样把作品奉献给人，作品总是作者自己的。可我还是希望把本书献给我的孩子——子瞻。当他宁静地呆在母亲肚子里时，本书的第2版同样也在孕育之中。现在，子瞻已经过了周岁生日，没有什么礼物可以比得上这本书更加让我感到自豪。我还要把本书献给我亲爱的妻子。写作虽然痛苦，可哪里及得上你分娩痛楚的万分之一。抚养子瞻的辛劳，更让虚弱的你身心憔悴。本书献给你，可否给你一丝安慰？

感谢我的父母。尤其感谢我的母亲。这一年多以来，调皮的子瞻折磨得您腰酸背痛，您却没有任何怨言，反而甘之如饴。我能有时间写作本书，您功不可没。

书一旦出版就不再属于作者。余下的，交给读者去品评吧！

张逸

2010年3月

# 前 言

## 缘起

用过去几年互联网上最酷，而在当下已经被用滥的名词来说，我在 2004 年成为了一名博客，用日志的方式记录自己成长的经历。坦白说，技术的成长远远比身体的发育更加地艰辛与缓慢，尤其是当今信息爆炸的年代，我们担忧的不是吃不饱，而是应该怎么吃，吃什么？营养不良固然令人堪忧，营养过剩却也不是健康之道。如果我们对软件技术做一次全方位的扫描，收获的无疑是面对岔路口的彷徨与迷茫，就像博尔赫斯笔下的曲径分岔的花园。

这是一种梦魇，就像在我的儿童时代，每次发高烧都会做的同一个恶梦一样，跑不掉，挣不脱，惊醒之后却又说不清的感觉。没人愿意走迷宫，除了那些以解谜题为乐趣的天才们。所以，我们在软件设计的迷宫门前停住了脚步；然后，四处顾盼寻找通过迷宫的地图。

不知道世界上是否真的存在穿过软件设计迷宫的地图，但至少有人曾经通过，并且留下了淡淡的足迹。这些足迹或者交互重叠，或者纷繁杂乱，分不清哪里才是走过的正确的路。于是，寻找、识别与尝试就成为我们面对技术更新的时候要经历的三部曲。

经典的技术，特别是经典的设计思想，完全可以免去这几个步骤。例如设计模式，在面向对象世界里，它已经成为了经典的存在，我们不必浪费时间去质疑它的重要性。省去了寻找、识别与尝试的过程，我们可以直接将它设定为亟待攻克的堡垒。正是基于这样的目标，我开始尝试与广大博友分享我的战斗攻略与心得。

博客的风格是“童言无忌”，所以我能够自由写意地耕耘博客园的一块田地。俗语云：

种瓜得瓜，种豆得豆。我种下了技术的种子，吸收着评论的养料，最后收获的却是现在这本呈现在读者面前的《软件设计精要与模式》，连我自己也要感到莫名惊诧了。书的出版缘起偶然。在我做完了一个长达一年多的项目之后，又参加了另外一个大型项目最后阶段的开发与测试，最后拒绝了一个周期可能长达几年的项目安排，结束了在北京的漂泊回到故乡。我开始了悠闲自得的放假生涯。一次偶然与博客园站长杜勇先生的闲谈，结束了我的休假状态，开始了数月的写书生涯。对于杜勇先生，我想把感谢的话放在本文的末尾，此时只想表达我的“愤慨”，是你，谋杀了我的闲适生活 ☺。

好在我这本书成不了指引人们走出迷宫的地图，所以我可以“没有责任心”地回过头来欣赏自己在迷宫墙上的涂鸦，即使是一个人的艺术，对于自己而言，也是一种美。萨特说：“存在即合理”，我相信本书能够体现这种逻辑的合理性。

## 本书有些什么

既然本书的内容源自于博客的内容，就必然继承了一些散漫自由的风格。我甚至建议读者把书中各章看作是一篇篇技术随笔，因为我实在厌倦了那种捧着高文大册时的战战兢兢了。于是，我努力营造一种华丽的风格，以此来冲击那些习惯了枯燥与平实的读者口味。

散文的要旨是“形散而神不散”，本书基本符合散文的精神。虽然每一章都可以看作是一个独立的部分，然而其中蕴涵的思想却是一致的，因而又可以合为一个整体。如果观察整体的局部，我们又可以将其割裂开来，成为一个单独存在的系列。如此安排，固然与博客的风格有关，却能够兼得一个好处，就是读者可以有选择地根据自身情况进行阅读，而不必囿于整部书的约束。

全书围绕着软件设计的核心内容，结合大量的实例与代码，使其告别了纯理论研究的空泛，将软件设计理论与项目实践完美地结合起来。遵循这样的原则，是因为我们在设计现实的软件，而不是在乌托邦中空想。软件开发，根本就没有乌托邦的乐园。

根据表述内容的不同，全书以渐进但绝非进阶的形式分成了 5 大篇。

第 1 篇为“设计之要”，相当于本书的总纲，列举了软件设计的要素，包括设计模式、重构、测试驱动开发、极限编程等。从涵盖面来看，“设计之要”一篇包含了软件设计中最重要“流行元素”，它们是程序员向设计师“涅槃”的基石，是从小工到专家的修炼法门。



受篇幅所限，我在阐述这些重要的设计要素时，只能是浅尝辄止，颇有几分意犹未尽的感觉。然而，我并不希望自己的论述浮于表面，而是希望能够结合具体的实例，深入浅出地解析它们的本质与精神。我希望读者在阅读完这些章节之后，有一种饥饿的感觉，不停地叩问与质疑，然后再“上穷碧落下黄泉”地去搜寻相关的资料，以求获得更加深入的理解。

在第2篇“.NET Framework 与设计模式”中，我偷懒地从.NET Framework 的现有设计中寻找到有关设计模式的实践，包括 Factory Method 模式、Composite 模式、Decorator 模式、Iterator 模式与 Strategy 模式。为了帮助读者更好地理解这些模式，在每一章中，我都采用了“抛砖引玉”的方式，首先分析相关模式的实质，并结合具体实例详解模式的目的与应用场景。最后，.NET Framework 粉墨登场，尽显自己主角的魅力。由于.NET Framework 是一个庞大的框架，包含的代码浩如烟海，不可能在短短的一章中穷尽其详细的设计。因而，我在引入.NET Framework 实例时，力求简洁，删去了与文章主旨无关的代码，并辅以 UML 图体现设计思想。对于.NET Framework 在实现相关模式时独具匠心的一面，我则不遗余力地给予详尽的介绍，力求深入透彻。

.NET Framework 是一道主食大餐，虽然美味可口，却未免有几分油腻。所以，我在第3篇中又上了一道清炒时蔬，希望能够压一压油荤。“媒体播放器的设计之旅”一篇，引入了一个虚构的媒体播放器项目。之所以称为“设计之旅”，是因为媒体播放器的最后完善，确实是一段艰难的征程，设置重重路障的是为我们提供需求的客户。本篇的独有之处是各章既相互独立，又有着藕断丝连般的联系。虽然是项目实践，但无疑更带了几分对相关设计模式探讨与研究的意味。本篇论述的设计模式包括 Factory Method 模式、Adapter 模式、Decorator 模式与 Visitor 模式。

或许第4篇“设计模式应用实践”最能体现本书的价值。它仍然是设计模式项目实践的延续，凸现其价值的原因在于这些项目实践，均是我亲自参与设计与开发的项目。无疑，它们的实现更具有现实的指导意义。我常常在想，究竟是什么阻碍了程序员对设计模式的理解？他们的普遍意见是不患设计模式之艰深难懂，而患设计模式运用之困难重重。如何将设计模式运用到实际的项目开发中，是他们面对的最大问题。所谓“读书百遍，其义自现”，我想，唯一的解决之道就是实践，实践，再实践。西方有谚语说“Practice make perfect”，翻译为中文就是“熟能生巧”。然而，盲目的实践终归是不成的，我们需要向他们展示一些已经实现了的示例。在对设计模式进行讲解时，我们固然需要示例的精巧、生动与形象，但最重要的特质还是要与实际的开发结合，否则，再贴切的示例都会成为“空中楼阁”。如