<<简约之美>>

图书基本信息

书名：<<简约之美>>

13位ISBN编号：9787115302380

10位ISBN编号：7115302383

出版时间：2013-1

出版时间：人民邮电出版社

作者：[美] Max Kanat-Alexander

页数：105

字数：94000

译者：余晟

前言

好程序员和差程序员的区别在于理解能力。
差劲的程序员不理解自己做的事情，优秀的程序员则相反。
信不信由你，道理就是这么简单。

写这本书，是为了帮助各位程序员，以适用于各种编程语言、各种项目的广阔视角来理解软件开发。
本书以普通人容易理解的方式，讲解了软件开发的科学规律。

如果你是程序员，这些规律能够说明，为什么有些开发方法有效，另一些无效。
这些规则也会指引你在日常工作中做出开发决策，帮助你的团队进行高质量的交流，最终制订出合理的计划。

如果你不是程序员，但身在软件行业，仍然可以享受到本书的价值：　　·它既是提供给初级程序员的优秀教材，又包含对高级程序员相当有 用的知识；　　·它帮助你更深入地理解软件工程师某些行为的原因，以及软件为何 要以某种方式来开发；　　·它帮助你理解优秀的软件工程师做决定的基本原理，让你与开发人 员更顺畅地沟通。

理想的状态是，软件行业中的每个人都可以阅读并理解这本书，即便他们没有多少编程经验，甚至母语不是英语也无所谓。
如果你已经有相当的技术积累，把握书中的概念会更加容易，但是大部分内容不需要编程经验就能理解。

实际上，本书虽然讲的是软件开发，却没有多少代码。
这怎么可能呢？
答案是，其中的思想适用于各种软件项目、各种语言。
要明白如何运用这些思想，并不需要懂得某一门具体的编程语言。
相反，本书中包含了大量的实例和比喻，它们会让你更好地理解所表述的每条原理。

最重要的是，这本书是为了帮助你而写的，希望能助你在软件开发中保持头脑清醒、遵守秩序、写出简洁代码。
我希望它读起来是一种享受，它有助于改善你的生活，你的软件。

排版约定　　本书中格式约定如下。

黑体：表示新术语。

等宽字体：用于代码示例，在段落中使用时，表示与程序有关的部分，比如变量或者函数名。

此图标表示提示、建议或者普通的旁注。

致谢　　Andy Oram和 Jolie Kanat两位编辑为本书作了巨大的贡献。
Andy的建议和意见深入且充满智慧；Jolie的坚持和支持促成了本书的最后出版，她为早期手稿所做的大量编辑工作尤其值得感谢。

Rachel Head是本书的文字编辑，做整理和校对的工作，她的才华无与伦比。

还要感谢的是与我在开源社区中一同工作或讨论过问题的程序员——尤其是在 Bugzilla项目中共事的几位开发人员，有了你们的帮助，我才有清楚的思维，讲解这些年来真实存在的，活生生的软件系统。

这些年来，我的 blog上收到的评论和反馈，帮我确定了本书的形式和内容。
在这里要感谢参与其中的所有人，即使你们仅仅给我鼓励，或者是告诉我你读过我的文章。

从个人来说，我尤其要感谢 Jevon Milan、Cathy Weaver，以及与他们工作过的所有人。
确切地说，有了他们，我才能写出这本书。
最后，要向我的朋友 Ron致敬，没有他，这本书根本不可能出现。

使用示例代码　　让我们助你一臂之力。
也许你要在自己的程序或文档中用到本书中的代码。
除非大段大段地使用，否则不必与我们联系取得授权。
例如，无需请求许可，就可以用本书中的几段代码写成一个程序。

但是销售或者发布O’Reilly图书中代码的光盘则必须事先获得授权。
引用书中的代码来回答问题也无需授权。
将大段的示例代码整合到你自己的产品文档中则必须经过许可。
　　我们非常希望你能标明出处，但并不强求。
出处一般含有书名、作者、出版商和ISBN，例如“ Code Simplicity：The Science of Software Development by Max Kanat-Alexander（O’Reilly，2012）版权所有。
　　如果有关于使用代码的未尽事宜，可以随时与我们取得联系。
　　Safari · 在线图书　　Safari在线图书是应需而变的数字图书馆。
它能够让你非常轻松地搜索7500多种技术性和创新性参考书以及视频，以便快速　　地找到需要的答案。
　　订阅后就可以访问在线图书馆内的所有页面和视频。
可以在手机或其他移动设备上阅读，还能在新书上市之前抢先阅读，也能够看到还在创作中的书稿并向作者反馈意见。
复制粘贴代码示例、放入收藏夹、下载部分章节、标记关键点、做笔记甚至打印页面等有用的功能可以节省大量时间。
这本书（英文版）也在其中。
欲访问本书英文版的电子版，或者由O’Reilly或其他出版社出版的相关图书，请到网站免费注册。
　　我们的联系方式　　请把对本书的评论和问题发给出版社。

<<简约之美>>

内容概要

《简约之美：软件设计之道》将软件设计作为一门严谨的科学，阐述了开发出优雅简洁的代码所应该遵循的基本原则。
作者从为什么以前软件设计没有像数学等学科一样成为一门科学开始入手，道出了软件以及优秀的软件设计的终极目标，并给出了具体的指导规则。

这是一本软件思想著作，适合任何背景、使用任何语言的程序员。

作者简介

Max Kanant-Alexander开源项目 Bugzilla总架构师， Google软件工程师，作家，8岁开始修电脑， 14岁开始编程。
Wdesimphicity.com和
fedorafaq.com网站维护者，目前居住在北加州。

<<简约之美>>

书籍目录

章节摘录

 程序的代码应当采用什么结构？
是程序的速度重要，还是代码容易阅读重要？
为满足需求，应该选择哪种编程语言？
软件设计与下列问题无关：公司的结构应该是怎样的？
什么时候召开团队会议？
程序员的工作时间应该如何安排？
程序员的绩效如何考核？
这些决策与软件本身无关，只与组织有关。
显然，保证这类决策的合理性也是重要的——许多软件项目之所以失败，就是管理失当。
但是这不是本书的主题，本书关注的是，如何为你的软件制订合理的技术决策。
软件系统中任何与架构有关的技术决策，以及在开发系统中所做的技术决策，都可以归到“软件设计”的范畴里。
2.1 程序员也是设计师 在软件项目中，每个程序员的工作都与设计有关。
首席程序员负责设计程序的总体架构；高级程序员负责大的模块；普通程序员则设计自己的那一小块，甚至只是某个文件的一部分。
但是，即便仅仅是写一行代码，也包含设计的因素。
哪怕是单干，也离不开设计。
有时候你在敲键盘之前，就做了决定，这就是在做设计。
有的夜晚，你躺在床上，还在思考要怎样编程。
每个写代码的人都是设计师，团队里的每个人都有责任保证自己的代码有着良好的设计。
任何软件项目里，任何写代码的人，在任何层面上，都不能忽略软件设计。
这不是说设计应该采取民主程序进行。
设计决不应该由某个委员会负责，那样的结果必然很差劲。
相反，所有的开发人员都应有权在自己的工作中做出良好的设计决策。
如果某位开发人员做了糟糕或者平庸的决策，资深开发人员或者首席程序员就应当推翻这些决策，重新来过。
对下属的设计，这些人应当拥有否决权。
不过，软件设计的责任应当落实在真正写代码的人身上。
身为设计师，必须时时愿意聆听建议和反馈，因为程序员大都比较聪明，有不错的想法。
但是考虑了所有这些建议和反馈之后，任何决策都必须由单独的个人而不是一群人来做出。
2.2软件设计的科学 现在，软件设计仍然不算科学。
什么是科学？
词典里的定义有点儿复杂，但简单来说，一门学问要成为科学，必须符合下列标准。
科学必须包含汇总而来的知识。
也就是，它必须包含事实而不是意见，且这些事实必须汇总起来（比如集结成书）。
这些知识必须具有某种结构。
知识必须能分类，其中的各个部分必须能够依据重要性之类的指标，妥善建立起与其他部分的联系。
科学必须包括一般性的事实或者基本的规则。
科学必须告诉你在现实世界中如何做一些事情。
它必须能够应用到工作或生活中。
通常，科学是经由科学方法来发现或证明的。
科学方法必须观察现实世界，提炼出关于现实世界的理论，通过验证理论，而且这些实验必须是可重复的。
这样才能说明理论是普适的真理，而不仅仅是巧合或者特例。

<<简约之美>>

编辑推荐

没有人喜欢复杂的东西，所以软件开发的简约之道一定会受读者青睐。
本书作者Max Kanat-Alexander创建的关于Linux的简约单页网站Unofficial Fedora FAQ，月访问量超过10万人次。
本书作者还是著名的开源Bugzilla Project的首席架构师、社区创始人和发布经理。

版权说明