

# 实现模式



[实现模式 下载链接1](#)

著者:Kent Beck

出版者:人民邮电出版社

出版时间:2012-12

装帧:

isbn:9787115294609

在《实现模式(修订版)》中，作者Kent

Beck将自己多年形成的编程习惯以及阅读既有代码的体验凝练成了编程中的价值观、原则和77种实现模式。在77种实现模式中，每一种模式都覆盖了编写简洁、清晰、易扩展、易维护的代码这一原则的某个方面。它们为日常的编程提供了丰富翔实的参考依据，并告诉大家这些代码如何为降低沟通成本和提高有效产出提供保障。

《实现模式(修订版)》适用于各个阶段的开发者群体。刚刚涉足软件开发领域的新人能够透过大师的眼睛来看待编程，了解编程的价值观与原则；具有丰富经验的资深工程师则可以通过这些模式进行反思，探究成功实践背后的意义。把价值观、原则和开发实践结合之后，日常开发工作便会以崭新而迷人的形式呈现在我们面前。

## 作者介绍:

Kent

Beck，JUnit测试框架创始人，《测试驱动开发》和《解析极限编程》的作者，软件业中最具创新力，最富盛名的领袖之一。长期以来，他一直致力于挑战软件工程教条，推动模式、测试驱动开发以及极限编程等思想的应用和传播。

## 目录: 译者序

前言

第1章 引言

1.1 章节概览

1.2 那么，现在……

第2章 模式

第3章 一种编程理论

3.1 价值观

3.2 原则

3.3 小结

第4章 动机

第5章 类

5.1 类

5.2 简单的超类名

5.3 限定性的子类名

5.4 抽象接口

5.5 interface

5.6 抽象类

5.7 有版本的interface

5.8 值对象

5.9 特化

5.10 子类

5.11 实现器

5.12 内部类

5.13 实例特有的行为

5.14 条件语句

5.15 委派

5.16 可插拔的选择器

5.17 匿名内部类

5.18 库类

5.19 小结

第6章 状态

6.1 状态

6.2 访问

6.3 直接访问

- 6.4 间接访问
- 6.5 通用状态
- 6.6 可变状态
- 6.7 外生状态
- 6.8 变量
- 6.9 局部变量
- 6.10 字段
- 6.11 参数
- 6.12 收集参数
- 6.13 可选参数
- 6.14 变长参数
- 6.15 参数对象
- 6.16 常量
- 6.17 按角色命名
- 6.18 声明时的类型
- 6.19 初始化
- 6.20 及早初始化
- 6.21 延迟初始化
- 6.22 小结
- 第7章 行为
  - 7.1 控制流
  - 7.2 主体流
  - 7.3 消息
    - 7.4 选择性消息
    - 7.5 双重分发
    - 7.6 分解性(序列性)消息
    - 7.7 反置性消息
    - 7.8 邀请性消息
    - 7.9 解释性消息
  - 7.10 异常流
  - 7.11 卫述句
  - 7.12 异常
    - 7.13 已检查异常
    - 7.14 异常传播
  - 7.15 小结
- 第8章 方法
  - 8.1 组合方法
  - 8.2 揭示意图的名称
  - 8.3 方法可见性
  - 8.4 方法对象
  - 8.5 覆盖方法
  - 8.6 重载方法
  - 8.7 方法返回类型
  - 8.8 方法注释
  - 8.9 助手方法
  - 8.10 调试输出方法
  - 8.11 转换
  - 8.12 转换方法
  - 8.13 转换构造器
  - 8.14 创建
  - 8.15 完整的构造器
  - 8.16 工厂方法
  - 8.17 内部工厂
  - 8.18 容器访问器方法

8.19 布尔值设置方法

8.20 查询方法

8.21 相等性判断方法

8.22 取值方法

8.23 设置方法

8.24 安全复制

8.25 小结

第9章 容器

9.1 隐喻

9.2 要点

9.3 接口

9.4 实现

9.5 Collections

9.6 继承容器

9.7 小结

第10章 改进框架

10.1 修改框架而不修改应用

10.2 不兼容的更新

10.3 鼓励可兼容的变化

10.4 小结

附录A 性能度量

A.1 示例

A.2 API

A.3 实现

A.4 MethodTimer

A.5 冲抵额外开销

A.6 测试

A.7 小结

参考书目

• • • • • (收起)

[实现模式](#) [下载链接1](#)

标签

软件工程

编程

实现模式

敏捷开发

软件开发

计算机

coding

Pattern

## 评论

很薄的一本书，花了两个小时重读了一遍，基本上没太遇到什么新的东西。不过有些话还是很有意思的，代码最重要的沟通了，要关心他人

---

代码量是条界，代码量不够的人看不懂 代码量够的人大部分都已经用上了

---

讲了很基础细节的东西，但却蕴含了很多原则性和视角性的东西。虽然讲的很多东西都懂，但看看还是不错的

---

讲了满多较实用的编程原则，还是需要在实践中领悟吧。ps  
翻译还是觉得读起来费劲。

---

扣一星，因为本书太枯燥。整体来说这本书内容是很精深的，但是属于“懂的自然懂，不懂觉得就是废话”的情况，所以本书两极分化的评论基本可以预见。从我一个半吊子程序员来看，本书内容真的很深入，虽然都是“看上去简单”的东西，但是在实际使用中，真的能做下去的，我严重怀疑有多少。

---

Java。无感。

---

一本很枯燥的书。

-----  
太干巴巴

挺短的一本书，内容不多，就是一些基础的代码组织模式。最有意思的前几章，讲价值观、原则和动机，这些比后面具体的模式更加重要。价值观：沟通，简单和灵活。

-----  
kent  
beck的作品，看到模式就忍不住想看看。看过后感觉已经是从其它书中吸取过的营养了....

-----  
kent  
beck的书，看到就想看一下，虽然有几个点get到了，但大部分还是已经知道的东西。所以本没有太多的惊喜。而且可能文字的描述不能完全表达出代码的东西，所以有些点都不是非常容易理解。

[实现模式](#) [下载链接1](#)

## 书评

这个评分完全是个人体验，为了让豆瓣更准确地推荐书而选取的，不一定代表这本书质量差或对其他人无用。书不厚，100来页。两小时就能读完。书的第三章，A Theory of Programming，是写得颇为精当的总纲。刚入门的程序员可以从这本书学到很多，毕竟Kent Beck流畅地讨论了怎么才...

-----  
这书十分枯燥  
与其说枯燥,不如说乏味,是因为里面的内容不外乎两种,一种你觉的早就懂的,另一种你觉得没有意义的,其实还有一种,就是你觉得错误的内容.所以在09年这本书仅仅因为很薄被我买到了手中,第一次读花了大概4天,痛苦的看完之后得到上面的的感觉.  
作为拥有发现真相...

简单滴说， 实现模式和设计模式这样的书就是教你积木应该怎么摆才能堆起一个又坚固又可以随便扩建， 看起来也不至于太怪异房子， 而不是教你每块积木应该是用沙子还是水来做， 也不告诉你堆积木应该是用钉子把积木订起来， 还是用浆糊粘起来。这本书是基于java语言讲的~...

這本書陸續看了有一年多， 紿給這本書一個簡單的星級上的評價真的很難。首先， 這本書真的是相當的枯燥， 陸續看了一年多， 真的是太枯燥了。看完這本書真的要非常大的毅力。這本書是作者對代碼層級上各種實現方法的總結， 從變量開始一直到最後的框架構築， 非常的瑣屑。大量的純...

这是一本关注程序细节并具有浓厚实战意义的书籍，它主要是来告诉我们如何写出容易让人理解的代码。如果非要在书名前加一个限定词的话，“java实现模式”可能更贴切一些，因为里面的代码都是以java作为例子来说明的。作者是junit的贡献者之一，因此里面的很多例子都是从junit中来...

实现模式是介于设计模式和语言手册之间的一种指导原则。它较设计模式离实际编码更进一步，更为频繁的指导编码时的类似如何给变量，类命名，使用接口还是抽象类，这个数据应当放到这里还是那里等问题。而就我本身的编码经验来说，这些问题经常会让我纠结上很长时间，所以学习并...

传说中smalltalk best patterns 的java版， refactoring和dhh都极力推荐， 尽管他们推的都是smalltalk的版本。但smalltalk在国内基本没市场， 我也不懂smalltalk， 自然就拿java版来看了。书挺薄的， 看起来也不用花多少时间。但没有什么收获。这本书涉及的内容基本上《effective j...

《实现模式》是最近看过的一本技术类的书籍中我认为评价比较低的。我今年开始看了许多本技术类的书籍。有些书籍是我看了之后感到醍醐灌顶的。比如《持续交付》这本书，还有《重构与模式》这个书。这些书的特点就在于除了让你看了之后深深感叹原来是这么回事，原来这就是我在...

Kent大叔虽说是软件工程方面的大师，但写的书倒是一点架子没有，没有玄而又玄的理

论、抽象的名词。更让人高兴的是薄薄的100多页，旅行泛读、居家研读都不会有很大的负担。

作者多年来不懈余力的倡导敏捷开发，作为面向工程实践的读物，本书对于Java程序员来说就是一本实实在在...

---

这是一本关于如何写好代码的书。

如果你不认为写好代码是一件重要、困难并且有趣的事，请立即放下这本书。

什么是好的代码？可以工作的、性能良好的、不出bug的代码，就是好的代码吗？

所谓好的代码，除了其他所有要求以外，还应该清晰准确地传达写作者的想法。Martin Fowler...

---

[实现模式](#) [下载链接1](#)