

<<实现模式>>

图书基本信息

书名：<<实现模式>>

13位ISBN编号：9787115187093

10位ISBN编号：7115187096

出版时间：2008-11

出版时间：人民邮电出版社

作者：Kent Beck

页数：157

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<实现模式>>

前言

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《实现模式》。

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。

MartinFowler在《重构：改善既有代码的设计》里说，“任何一个傻瓜都能写出机器能懂的代码。好的程序员应该写出人能懂的代码。

” 如果你不同意这句话，请立即放下这《实现模式》。

因为这是一本关于如何用代码与他人（而非机

<<实现模式>>

内容概要

在本书中，作者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。

沟通、简单和灵活的价值观应当被所有开发人员所铭记。

局部影响、最小化重复、将逻辑与数据捆绑等原则同样是通用性的指导思想，比价值观更贴近编程场景，在价值观和模式之间搭建了桥梁。

在77个实现模式中，每一个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。

本书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；而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迷人的形式呈现在我们面前。

<<实现模式>>

书籍目录

Chapter 1: Introduction	1	Tour Guide	3	And Now	4	Chapter 2: Patterns	5	Chapter 3: A Theory of Programming	9	Values	10	Communication	10	Simplicity	11	Flexibility	12	Principles	13	Local Consequences	13	Minimize Repetition	14	Logic and Data Together	14	Symmetry	15	Declarative Expression	16	Rate of Change	17	Conclusion	18	Chapter 4: Motivation	19	Chapter 5: Class	21	Class	22	Simple Superclass Name	23	Qualified Subclass Name	24	Abstract Interface	24	Interface	26	Abstract Class	26	Versioned Interface	27	Value Object	28	Specialization	31	Subclass	32	Implementor	34	Inner Class	34	Instance-Specific Behavior	36	Conditional	36	Delegation	38	Pluggable Selector	40	Anonymous Inner Class	41	Library Class	41	Conclusion	42	Chapter 6: State	43	State	44	Access	45	Direct Access	46	Indirect Access	47	Common State	47	Variable State	48	Extrinsic State	50	Variable	50	Local Variable	51	Field	52	Parameter	53	Collecting Parameter	55	Optional Parameter	56	Var Args	56	Parameter Object	57	Constant	58	Role-Suggesting Name	58	Declared Type	60	Initialization	61	Eager Initialization	61	Lazy Initialization	62	Conclusion	62	Chapter 7: Behavior	63	Control Flow	64	Main Flow	64	Message	65	Choosing Message	65	Double Dispatch	66	Decomposing (Sequencing) Message	67	Reversing Message	67	Inviting Message	68	Explaining Message	69	Exceptional Flow	70	Guard Clause	70	Exception	72	Checked Exceptions	72	Exception Propagation	73	Conclusion	73	Chapter 8: Methods	75	Composed Method	77	Intention-Revealing Name	79	Method Visibility	80	Method Object	82	Overridden Method	83	Overloaded Method	83	Method Return Type	84	Method Comment	85	Helper Method	85	Debug Print Method	86	Conversion	87	Conversion Method	87	Conversion Constructor	88	Creation	88	Complete Constructor	89	Factory Method	90	Internal Factory	91	Collection Accessor Method	91	Boolean Setting Method	93	Query Method	93	Equality Method	94	Getting Method	95	Setting Method	96	Safe Copy	97	Conclusion	98	Chapter 9: Collections	99	Metaphors	100	Issues	101	Interfaces	103	Array	103	Iterable	104	Collection	104	List	104	Set	105	SortedSet	105	Map	106	Implementations	107	Collection	108	List	108	Set	108	Map	109	Collections	110	Searching	111	Sorting	112	Unmodifiable Collections	113	Single-Element Collections	114	Empty Collections	114	Extending Collections	114	Conclusion	115	Chapter 10: Evolving Frameworks	117	Changing Frameworks without Changing Applications	117	Incompatible Upgrades	118	Encouraging Compatible Change	120	Library Class	121	Objects	121	Conclusion	129	Appendix A: Performance Measurement	131	Example	131	API	132	Implementation	133	MethodTimer	134	Canceling Overhead	136	Tests	136	Comparing Collections	137	Comparing ArrayList and LinkedList	139	Comparing Sets	140	Comparing Maps	141	Conclusion	142	Bibliography	145	General Programming	145	Philosophy	147	Java	148	Index	149
-------------------------	---	------------	---	---------	---	---------------------	---	------------------------------------	---	--------	----	---------------	----	------------	----	-------------	----	------------	----	--------------------	----	---------------------	----	-------------------------	----	----------	----	------------------------	----	----------------	----	------------	----	-----------------------	----	------------------	----	-------	----	------------------------	----	-------------------------	----	--------------------	----	-----------	----	----------------	----	---------------------	----	--------------	----	----------------	----	----------	----	-------------	----	-------------	----	----------------------------	----	-------------	----	------------	----	--------------------	----	-----------------------	----	---------------	----	------------	----	------------------	----	-------	----	--------	----	---------------	----	-----------------	----	--------------	----	----------------	----	-----------------	----	----------	----	----------------	----	-------	----	-----------	----	----------------------	----	--------------------	----	----------	----	------------------	----	----------	----	----------------------	----	---------------	----	----------------	----	----------------------	----	---------------------	----	------------	----	---------------------	----	--------------	----	-----------	----	---------	----	------------------	----	-----------------	----	----------------------------------	----	-------------------	----	------------------	----	--------------------	----	------------------	----	--------------	----	-----------	----	--------------------	----	-----------------------	----	------------	----	--------------------	----	-----------------	----	--------------------------	----	-------------------	----	---------------	----	-------------------	----	-------------------	----	--------------------	----	----------------	----	---------------	----	--------------------	----	------------	----	-------------------	----	------------------------	----	----------	----	----------------------	----	----------------	----	------------------	----	----------------------------	----	------------------------	----	--------------	----	-----------------	----	----------------	----	----------------	----	-----------	----	------------	----	------------------------	----	-----------	-----	--------	-----	------------	-----	-------	-----	----------	-----	------------	-----	------	-----	-----	-----	-----------	-----	-----	-----	-----------------	-----	------------	-----	------	-----	-----	-----	-----	-----	-------------	-----	-----------	-----	---------	-----	--------------------------	-----	----------------------------	-----	-------------------	-----	-----------------------	-----	------------	-----	---------------------------------	-----	---	-----	-----------------------	-----	-------------------------------	-----	---------------	-----	---------	-----	------------	-----	-------------------------------------	-----	---------	-----	-----	-----	----------------	-----	-------------	-----	--------------------	-----	-------	-----	-----------------------	-----	------------------------------------	-----	----------------	-----	----------------	-----	------------	-----	--------------	-----	---------------------	-----	------------	-----	------	-----	-------	-----

章节摘录

Introduction Here we are together . You ' ve picked up my book(its yours now) . You already write code . You have probably already developed a style of your own through your own experiences . The goal of this book is to help you communicate your intentions through your code . The book begins with an overview of programming and patterns (chapters 2-4) . The remainder of the book(chapters 5-8)is a series of short essays , patterns , on how to use the features of Java to write readable code . It closes with a chapter on how to modify the advice here if you are writing frameworks instead of applications . Throughout , the book is focused on programming techniques that enhance communication . There are several steps to communicating through code . First I had to become conscious while programming . I had been programming for years when I first started writing implementation patterns . I was astonished to discover that , even though programming decisions came smoothly and quickly to me , I couldn ' t explain why I was SO sure a method should be called such . and . SO or that a bit of logic belonged in this object over here . The first step towards communicating was slowing down long enough to become aware of what I was thinking , to stop pretending that I coded by instinct .

The second step was acknowledging the importance of other people . I found programming satisfying , but I am self-centered . Before I could write communicative code I needed to believe that other people were as important as I was . Programming is hardly ever a solitary communion between one man and one machine . Caring about other people is a conscious decision , and one that requires practice . Which brings me to the third step . Once I had exposed my thinking to sunlight and fresh air and acknowledged that other people had as much right to exist as I did , I needed to demonstrate my new perspective in practice . I use the implementation patterns here to program consciously and for others as well as myself .

<<实现模式>>

编辑推荐

在《实现模式》中，作者收集了77种模式，用于处理日常编程任务。编写可读性更高的代码。

这个新的模式集合涵盖了开发中的诸多方面，包括类、状态、行为、方法、容器及框架等。

图表、故事、示例和随笔的配合使用，使作者笔下的种种模式显得异彩纷呈，从中可以发现许多经受过实践检验的解决方案，小至变量命名，大到异常检查。

Kent Beck，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。

他是Three Rivers Institute的创始人和董事，也是Agitar Software的成员。

他在Addison-Wesley出版了多《实现模式》(英文版)籍，其中包括Test-Development(中译版《测试驱动开发》)。

以及与Cyrilthia Andres合著的EXtreme Programming, Explained(中译版《解析极限编程》)。

《实现模式》(英文版)包括以下内容：

- 通过代码进行交流的重要性和隐藏在模式背后的思想；

- 何时创建类，如何创建类，类又是怎样书写逻辑的；
- 存储和获取状态的最佳实践；

- 行为，即用于表述逻辑(包括可选执行路径)的模式；
- 编写方法。

为方法命名，以及对方法进行分解；

- 选择和使用容器；
- 实现模式在用于构建框架时的变体。

“ Kent是用代码来沟通的大师，他的代码条分缕析，明晓清晰，如益友携手，如良师解惑，令人豁然开朗。

” ——Erich Gamma，IBM杰出工程师

<<实现模式>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>