

《软件设计重构》

图书基本信息

书名：《软件设计重构》

13位ISBN编号：9787115431248

出版时间：2016-8

作者：[印度] Girish Suryanarayana,[印度] Ganesh Samarthyam,[印度] Tushar Sharma

页数：210

译者：袁国忠

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《软件设计重构》

内容概要

本书主要介绍25个软件架构坏味，它们在确定设计问题时的作用以及可用的重构方法，并结合图表和示例给出了详尽说明，旨在引领读者掌握代码易读、易修改的关键，让代码具备重构能力。另外，本书将何时应该重构、重构时遇到的一些常见问题穿插在了示例讲解中。

《软件设计重构》

作者简介

作者简介：

Girish Suryanarayana

印度班加罗尔西门子公司研究与技术中心高级核心专家、高级研究科学家。

Ganesh Samarthyam

CodeOps Technologies创始人之一，曾任西门子公司软件架构与开发小组成员、独立顾问、企业培训师

。

Tushar Sharma

雅典经济与商业大学研究员、IEEE高级会员，曾任印度班加罗尔西门子公司研究与技术中心技术专家

。

译者简介：

袁国忠

自由译者；2000年起专事翻译，主译图书，偶译新闻稿、软文；出版译著40余部，其中包括《C++ Prime Plus中文版》《CCNA学习指南》《CCNP ROUTE学习指南》《面向模式的软件架构：模式系统》《Android应用UI设计模式》《风投的选择：谁是下一个十亿美元级公司》等，总计700余万字；专事翻译前，从事过三年化工产品分析和开发，做过两年杂志和图书编辑。

书籍目录

第1章 技术债务	1
1.1 何为技术债务	2
1.2 技术债务的组成部分	2
1.3 技术债务的影响	3
1.4 引发技术债务的因素	5
1.5 如何管理技术债务	6
第2章 设计坏味	7
2.1 为何要关心坏味	8
2.2 导致坏味的原因	9
2.2.1 违反设计原则	10
2.2.2 不恰当地使用模式	10
2.2.3 语言的局限性	11
2.2.4 面向对象中的过程型思维	11
2.2.5 粘滞性	11
2.2.6 未遵循最佳实践和过程	12
2.3 如何消除坏味	12
2.4 本书涵盖的坏味	12
2.5 一种设计坏味分类方案	13
2.5.1 基于设计原则的坏味分类	13
2.5.2 坏味命名方案	14
2.5.3 坏味记录模板	15
第3章 抽象型坏味	16
3.1 缺失抽象	19
3.1.1 理据	19
3.1.2 潜在的原因	19
3.1.3 示例	20
3.1.4 重构建议	21
3.1.5 影响的质量指标	22
3.1.6 别名	22
3.1.7 现实考虑	23
3.2 命令式抽象	23
3.2.1 理据	23
3.2.2 潜在的原因	23
3.2.3 示例	24
3.2.4 重构建议	25
3.2.5 影响的质量指标	26
3.2.6 别名	28
3.2.7 现实考虑	28
3.3 不完整的抽象	28
3.3.1 理据	28
3.3.2 潜在的原因	29
3.3.3 示例	29
3.3.4 重构建议	31
3.3.5 影响的质量指标	32
3.3.6 别名	33
3.3.7 现实考虑	33
3.4 多方面抽象	34

3.4.1	理据	34
3.4.2	潜在的原因	34
3.4.3	示例	35
3.4.4	重构建议	36
3.4.5	影响的质量指标	37
3.4.6	别名	37
3.4.7	现实考虑	37
3.5	不必要的抽象	37
3.5.1	理据	38
3.5.2	潜在的原因	38
3.5.3	示例	38
3.5.4	重构建议	40
3.5.5	影响的质量指标	41
3.5.6	别名	41
3.5.7	现实考虑	41
3.6	未用的抽象	42
3.6.1	理据	42
3.6.2	潜在的原因	42
3.6.3	示例	43
3.6.4	重构建议	44
3.6.5	影响的质量指标	45
3.6.6	别名	46
3.6.7	现实考虑	46
3.7	重复的抽象	46
3.7.1	理据	47
3.7.2	潜在的原因	47
3.7.3	示例	48
3.7.4	重构建议	50
3.7.5	影响的质量指标	51
3.7.6	别名	51
3.7.7	现实考虑	52
第4章	封装型坏味	53
4.1	不充分的封装	55
4.1.1	理据	55
4.1.2	潜在的原因	55
4.1.3	示例	56
4.1.4	重构建议	60
4.1.5	影响的质量指标	62
4.1.6	别名	62
4.1.7	现实考虑	62
4.2	泄露的封装	63
4.2.1	理据	63
4.2.2	潜在的原因	64
4.2.3	示例	64
4.2.4	重构建议	67
4.2.5	影响的质量指标	69
4.2.6	别名	69
4.2.7	现实考虑	69
4.3	缺失封装	70

4.3.1	理据	70
4.3.2	潜在的原因	71
4.3.3	示例	71
4.3.4	重构建议	73
4.3.5	影响的质量指标	76
4.3.6	别名	77
4.3.7	现实考虑	77
4.4	未利用封装	77
4.4.1	理据	77
4.4.2	潜在的原因	78
4.4.3	示例	78
4.4.4	重构建议	80
4.4.5	影响的质量指标	80
4.4.6	别名	82
4.4.7	现实考虑	82
第5章	模块化型坏味	83
5.1	拆散的模块化	85
5.1.1	理据	86
5.1.2	潜在的原因	86
5.1.3	示例	86
5.1.4	重构建议	88
5.1.5	影响的质量指标	90
5.1.6	别名	90
5.1.7	现实考虑	91
5.2	不充分的模块化	91
5.2.1	理据	91
5.2.2	潜在的原因	92
5.2.3	示例	92
5.2.4	重构建议	95
5.2.5	影响的质量指标	96
5.2.6	别名	96
5.2.7	现实考虑	96
5.3	循环依赖式模块化	97
5.3.1	理据	97
5.3.2	潜在的原因	98
5.3.3	示例	99
5.3.4	重构建议	101
5.3.5	影响的质量指标	105
5.3.6	别名	106
5.3.7	现实考虑	106
5.4	轮毂式模块化	107
5.4.1	理据	107
5.4.2	潜在的原因	107
5.4.3	示例	107
5.4.4	重构建议	109
5.4.5	影响的质量指标	110
5.4.6	别名	110
5.4.7	现实考虑	110
第6章	层次结构型坏味	111

6.1	缺失层次结构	115
6.1.1	理据	115
6.1.2	潜在的原因	115
6.1.3	示例	115
6.1.4	重构建议	117
6.1.5	影响的质量指标	119
6.1.6	别名	120
6.1.7	现实考虑	120
6.2	不必要的层次结构	121
6.2.1	理据	121
6.2.2	潜在的原因	121
6.2.3	示例	122
6.2.4	重构建议	125
6.2.5	影响的质量指标	126
6.2.6	别名	126
6.2.7	现实考虑	127
6.3	未归并的层次结构	127
6.3.1	理据	127
6.3.2	潜在的原因	128
6.3.3	示例	128
6.3.4	重构建议	132
6.3.5	影响的质量指标	134
6.3.6	别名	135
6.3.7	现实考虑	135
6.4	过宽的层次结构	136
6.4.1	理据	136
6.4.2	潜在的原因	137
6.4.3	示例	137
6.4.4	重构建议	138
6.4.5	影响的质量指标	139
6.4.6	别名	139
6.4.7	现实考虑	140
6.5	凭空想象的层次结构	140
6.5.1	理据	140
6.5.2	潜在的原因	140
6.5.3	示例	141
6.5.4	重构建议	141
6.5.5	影响的质量指标	142
6.5.6	别名	142
6.5.7	现实考虑	143
6.6	过深的层次结构	143
6.6.1	理据	143
6.6.2	潜在的原因	143
6.6.3	示例	144
6.6.4	重构建议	145
6.6.5	影响的质量指标	146
6.6.6	别名	147
6.6.7	现实考虑	148
6.7	叛逆型层次结构	148

6.7.1	理据	148
6.7.2	潜在的原因	148
6.7.3	示例	149
6.7.4	重构建议	150
6.7.5	影响的质量指标	153
6.7.6	别名	154
6.7.7	现实考虑	154
6.8	支离破碎的层次结构	157
6.8.1	理据	158
6.8.2	潜在的原因	158
6.8.3	示例	158
6.8.4	重构建议	163
6.8.5	影响的质量指标	164
6.8.6	别名	164
6.8.7	现实考虑	165
6.9	多路径层次结构	166
6.9.1	理据	166
6.9.2	潜在的原因	167
6.9.3	示例	167
6.9.4	重构建议	170
6.9.5	影响的质量指标	171
6.9.6	别名	171
6.9.7	现实考虑	171
6.10	循环层次结构	172
6.10.1	理据	172
6.10.2	潜在的原因	173
6.10.3	示例	173
6.10.4	重构建议	173
6.10.5	影响的质量指标	175
6.10.6	别名	176
6.10.7	现实考虑	176
第7章	坏味生态系统	177
7.1	具体情况的影响	177
7.2	坏味的相互影响	180
7.2.1	坏味通常不单独出现	180
7.2.2	坏味可能昭示着存在更深层的问题	183
第8章	技术债务偿还实战	185
8.1	工具	185
8.1.1	理解工具	186
8.1.2	评估工具、代码克隆检测器和度量工具	186
8.1.3	技术债务量化和可视化工具	187
8.1.4	重构工具	187
8.1.5	实际使用工具	187
8.2	流程	188
8.2.1	重构面临的挑战	188
8.2.2	让人认可重构	188
8.2.3	IMPACT——一个重构流程模型	189
8.2.4	技术债务偿还重构最佳实践	192
8.3	人员	193

8.3.1	培训	193
8.3.2	研讨会和讲座	193
8.3.3	以身作则	193
附录A	软件设计原则	194
附录B	技术债务偿还工具	197
附录C	示意图使用的表示法	200
附录D	推荐读物	202
参考文献		204

《软件设计重构》

精彩短评

1. 前几章翻译有点拗口；2. 内容侧重程序结构的设计；3. 引用了一打十几年前的书，有些想看都不太找得到了；4. 其实可以算一本《设计模式》的安利书，看完再看一眼设计模式去了；5. Java utils包要被黑哭了
- 2、基本是重构一本书的翻版，或者说挑取了一些结合案例讲解了一下如何重构，可以认为是一本简化版的重构，没事可以翻翻。
- 3、虽不如《重构》大而全，但事实例子丰富，有说服力，不像前者这么"高冷"。
部分地方翻译得有点生硬
- 4、作者详细地列举了代码设计和开发中的问题所在及可能原因。算是一本不错地代码坏味指南。
- 5、重构案例都是基于JDK的，很有示范作用，人无完人JDK也是有很多设计缺陷的。

《软件设计重构》

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：www.tushu000.com