

Agile

Fault-Error-Failure ■

Retry Component Fault

Handle-On-Delay Rule

Unchecked Exception

Exception 

Throw

Spare Handler

Bad Smell

Exception Handling Refactoring

● Recovery



Architecture Context

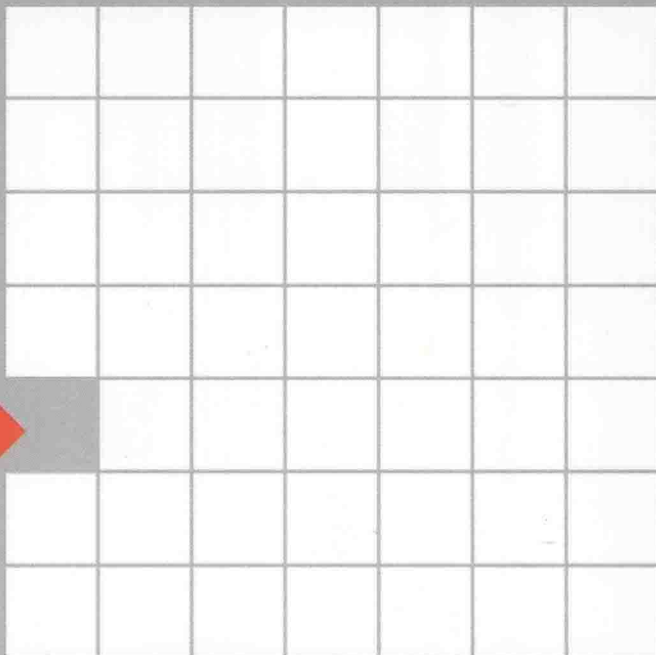
Design Fault

Termination Model

Checked Exception

Graceful Fault-Tolerant

Diversity——



| 笑 | 谈 | 软 | 件 | 工 | 程 |

异常处理的设计与重构

雅俗共赏，开发人员众口齐赞的明星博主
Teddy Chen

陈建村
许晓斌 审校
著

在这个世界上，大部分软件开发“大师”都只教你如何设计“软件的光明面”，很少有人愿意告诉你如何解决“软件的黑暗面”。
但很遗憾的是，软件的异常状态（黑暗面）并不会比正常状态（光明面）少，是时候重视“异常处理”了。

俗话说得好：“夜路走多了，总会碰到鬼。”

本书积累了作者众多“碰到鬼”的经验，
从异常处理设计出发，探讨“软件的黑暗面”的方方面面。
通过妥善的异常处理，协助读者开发出高质量的软件！



清华大学出版社



笑——谈——软——件——工——程

异常处理的 设计与重构

雅俗共赏，开发人员众口齐赞的明星博主

Teddy Chen

陈建村
著

清华大学出版社
北 京

内 容 简 介

本书凝聚作者十余年的思考与实践,从异常处理的设计与重构入手,介绍了如何提升软件的健壮性。全书共六部分,从异常处理的现状谈起,逐一条分缕析,阐述异常处理的基本概念、工作机制、异常处理的4+1观念,介绍软件健壮度等级和异常处理策略,讨论异常处理的坏味道和重构。

作者以诙谐幽默的文笔,于谈笑间揭示异常处理的来龙去脉,兼顾实用性和趣味性,非常适合善于思考和勤于实践的软件工程师阅读,也适合软件学院和计算机相关专业学生参考。

本书为精誠資訊股份有限公司-悦知文化授权清华大学出版社于中国大陆(台港澳除外)地区之中文简体版本。本著作物之专有出版权为精誠資訊股份有限公司-悦知文化所有。该专有出版权受法律保护,任何人不得侵害之。

北京市版权局著作权合同登记号 图字:01-2014-5241

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

笑谈软件工程:异常处理的设计与重构/陈建封著. —北京:清华大学出版社,2015

ISBN 978-7-302-39449-5

I. ①笑… II. ①陈… III. ①软件工程 IV. TP311

中国版本图书馆 CIP 数据核字(2015)第 036497 号

责任编辑:文开琪

封面设计:杨玉兰

责任校对:周剑云

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×230mm

印 张:22.75

字 数:254千字

版 次:2015年5月第1版

印 次:2015年5月第1次印刷

印 数:1~3500

定 价:69.00元

致谢/Acknowledgments

首先要感谢“生活压力”让 Teddy 下定决心写这本书。不是为了赚版税，因为版税真的少得可怜，而是希望藉由本书的出版，可以和更多人分享 Teddy 在“通过妥善的异常处理开发出高质量软件”这方面的经验，增加日后开课与顾问服务的生意。

异常处理对于绝大多数软件开发人员而言，是非常伤脑筋的难题。Teddy 在涉足这个领域（2005 年）之前，每当遇到异常处理的问题也是被搞得非常头痛。感谢指导教授郑有进老师与谢金云老师，当年在“半推半就”的情况下，他们让 Teddy “多做”了这个题目，使得自己有机会打破一个多年来存在自己心中未解的谜团。也感谢台北科技大学资工系软件系统实验室的学弟、学妹们，多年来持续耕耘着异常处理设计这个冷门课题，不但拓展了 Teddy 的学习领域，也让自己免于独学而无友的窘境。

本书许多灵感与内容受到多位前辈所撰写的书籍与文章的启发，族繁不及备载，在此一并谢过。也要感谢本书的编辑萧书瑜小姐与悦知文化幕后工作人员的辛劳，让本书得以顺利出版。还要感谢曾经参加“异常处理设计与重构实现班”的学员，你们对于课程的反馈意见，成为 Teddy 修正本书内容的参考。

最后，也是最重要的，感谢陪伴 Teddy 多年的另一半 Kay，持续忍受 Teddy 长时间在计算机前面写稿与耕耘自己的博客“搞笑谈软工”。还有 Teddy 的母亲，因为没有她也就没有本书的作者，也就不可能有本书的诞生。

序言/Preface

这是一本异常处理的书，

也是一本软件设计的书。

是的，身为一位专业的软件开发人员，从小到大你可能学过各式各样的软件设计技术与方法。从最基础的程序语言（C、C++、Java、C#、VB.NET、Objective-C、JavaScript、Ruby、Python 等）、数据结构与算法，到面向对象分析与设计、设计模式、软件架构以及各种敏捷开发实践，包含自动化测试、测试驱动开发、行为驱动开发、持续集成和敏捷设计原则等。以上，所有的“大师”费尽心力，都在告诉你“如何设计软件的光明面？”

什么？！软件的光明面？是的，软件跟宇宙万物一样，有光明面也有黑暗面。软件的光明面就是“正常行为”（normal behavior），而黑暗面则是“异常行为”（abnormal behavior）。就好像一般民众看到“黑道”与“坏人”，人人莫不避而远之。大部分的开发人员对于软件的“黑暗面”与“异常行为”，都感到很头大，在内心中抱持着“闭上眼睛就以为看不见”的心态：“反正程序看起来可以动就好了”，您说是吗？

软件设计包含着正常行为与异常行为，两者相互影响，忽略任何一方，都可能让原本精心规划的设计变得不堪一击。既然异常行为的设计那么重要，与正常行为相较，为什么很少人谈论异常行为？

因为很难。

对付异常行为很难，因为它“躲在正常行为的影子下”，很容易被有意、无意地忽略。其次，它与正常行为“水乳交融”、“我中有你，你中有我”。套一句软件开发的术语，它是一种“横切关注点”（cross-cutting concern），需要具备怪医黑杰克与庖丁解牛的能力，同时

关注方方面面，才有可能让你在对“病人”（程序）进行“外科手术”（处理异常）的时候，避免不小心切断动脉，导致“失血过多”（越改越乱）而让病人提早上天堂的窘境。

俗话说“夜路走多了，自然会碰到鬼”，本书累积了 Teddy “碰到鬼”的经验，从异常处理设计出发，专门探讨软件设计黑暗面的问题，希望补足开发人员长久以来所欠缺的“维他命”。

这是一本写给开发人员的书，
也是一本写给管理阶层的书。

看到书名，你可能会以为这是一本写给开发人员的书。错！开发人员的钱要赚，管理阶层的钱也要赚。嗯嗯，Teddy 的意思是说，异常处理做得好，可以提升软件的强健度，提高客户满意度，帮公司赚大钱，减少程序错误，早点下班回家陪家人，并顺便护肝。

本书分为 6 部分 42 章，有些地方技术含量相对较低，对于不想知道太多异常处理实现细节的管理阶层，可以只看这几个部分。

另外像是 Column A “倒霉的人很适合当神秘客”、Column C “网络又断”、Column D “这是你的问题，不是我的问题”、Column E “你如何评价成功”、Column G “升级、降级，傻傻分不清楚”、Column I “客户满意，老板赚钱，你护肝”，也很适合管理阶层当作小说来看。

至于开发人员，当然是要“怒看”整本书的内容才够“给力”。

是否与程序语言相关

异常处理设计重点不在程序语言和语法，而在背后的设计原理。虽然本书范例以 Java 语言为主，但也一并讨论了许多 Java 语言和 C# 语言（支持与不支持 checked exception 的双方代表）在异常处理设计精神与作法上的差异。书中所介绍的异常处理设计观念、名词、原则、方法与重构，只要是面向对象语言，不管是 Java、C#、VB.NET、Python、Ruby 等，差不多都适用。

至于采用 Java 语言作为例子的原因很简单：因为它是当代流行的商业语言里面，异常处理机制最困难（或是说最“讨厌”）的语言。搞懂了 Java 的异常处理，再应用到其它面向对象语言，就好像喝开水一样，变得非常容易。

以上用白话文来说就是，不管你使用什么语言，都应该买一本带回家。

书中用语和排版

以一本中文书而言，书中的英文稍微多了一些。有些英文术语译成中文之后，阅读起来总是少了点味道，所以 Teddy 会选择性地保留一些英文，例如 `checked exception` 和 `unchecked exception`。但英文名词第一次出现的时候，在它的“前后”或“附近”一定会有相对应的中文翻译或说明。

为了区分特别的名词，本书程序代码采用 Courier New 字体，而书中提到模式名称(pattern name)、坏味道(bad smell)、重构(refactoring)，则使用大写开头的英文斜体字表示。例如 *Command*、*Dummy Handler*。对于本书中所提出的异常处理重构，除了以斜体字表示以外，还会在名称之后加上《重构》一书中介绍该重构的章节页码，例如 *Replace Error Code with Exception (299)*。

一些软件领域常用的英文名词书中直接用中文表示，例如用 `function` 与 `method` 翻译成“函数”（有些中文书会使用“函数”或“方法”）。`class`、`object`、`instance` 和 `type` 用“类”、“对象”、“实例”和“类型”表示。`thread`、`process` 用“线程”和“进程”称呼。`Application Programming Interface`（应用程序编程接口）则直接使用 API。

最后，重要的概念和知识点以黑体字加以强调。

本书许多内容最初来自于 Teddy 的博士论文“Java 异常处理：模

型、重构与模式”（原题目为 Java Exception Handling: Models, Refactorings, and Patterns）以及工作上的实践经验，经改写后，从 2007 年起陆续发表在“搞笑谈软工”博客，最后集成成册再加上“强烈月光照射”之后，变身成为大家现在看到的这本书。书中内容经过 Teddy 多次修改与审校，但刻意保留原本在博客中以“乡民”称呼读者的方式。此为特意为之，并非 Teddy 偷懒，特此说明。

乡民们若对本书内容有任何问题、心得、称赞，或者发现错误，或是看到不爽想翻桌的时候，欢迎直接写信到 teddy.chen.tw@gmail.com 给 Teddy 或加入 Facebook 上的“搞笑谈软工社团”，网址为 <https://www.facebook.com/groups/teddy.tw/>，我们一起讨论。

目录/Contents

Part 1 异常处理的现状

Chapter 01	你对异常处理的了解有多少？	3
02	异常处理的重要性	6
03	“种花”电信，我真是猜不透你啊！	14
04	HTC One X——发现一个 Bug	21
05	HTC One X——我只是想打通电话呀！	25
06	HTC One X——拍照的错误消息描述	30
07	可靠性可以是一门生意	35
Column A.	倒霉的人很适合当神秘客	39

Part 2 异常处理的基本观念

Chapter 08	强健性大战首部曲——魅影危机	47
Column B.	找不到数据，要传回 Null 还是丢出异常？	54
09	异常处理的四种脉络	57
10	面向对象语言的异常处理机制	69
11	你的汽车有多耐撞？谈谈异常安全性	86

12 对比异常处理和容错设计	90
Column C. 网络又断了	95

Part 3 JAVA 语言的异常处理机制

Chapter 13 Java 的异常处理	101
14 我的异常被 finally 块覆盖了	110
15 被打压的异常——抢救异常大作战	116
16 清理失效的语义要表达清楚	122
17 自己制作“被打压的异常”	131
18 Try、Catch 和 Finally 的责任分担	144
Column D. 这是你的问题，不是我的问题	152
Chapter 19 异常处理失败，怎么办？	156
20 已查异常与未查异常的语义与问题	159
21 接口演进	164

Part 4 为什么异常处理那么难？异常处理的 4+1 观点

Chapter 22 用途观点	173
23 设计观点	177
24 处理观点	181

25	工具支持观点	185
26	流程观点	191
Column E.	你如何评价成功	198

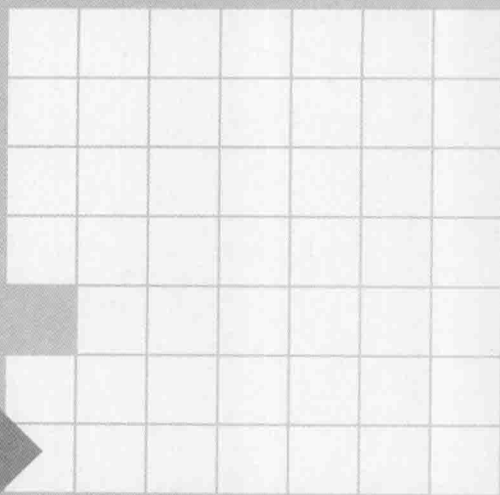
Part 5 强健度等级与异常处理策略

Chapter 27	异常处理设计的第一步——决定强健度等级	203
28	强健度等级 1——错误报告的实现策略	211
29	强健度等级 2——状态恢复的实现策略	216
30	强健度等级 3——行为恢复的实现策略	223
Column F.	VMWare 越狱之替代方案	231
Chapter 31	异常类的设计与使用技巧	235
32	终止或继续	245
33	自动化更新	249
Column G.	升级、降级，傻傻分不清楚	253

Part 6 异常处理坏味道与重构

Chapter 34	异常处理坏味道	261
Column H.	仙人打鼓有时错——谈谈 <i>Clean Code</i> 一书中介绍的异常处理	269

Chapter 35	用异常代替错误码.....	275
36	以未查异常取代忽略已查异常	281
37	以重新丢出异常代替空的处理程序	285
38	使用最外层 Try 语句避免意外终止	290
39	以函数取代嵌套的 Try 语句	295
40	引入 Checkpoint 类	301
41	引入多才多艺的 Try 块.....	309
Column I.	客户满意，老板赚钱，你护肝	317
Chapter 42	一个函数只能有一个 Try 语句	326
Column J.	眼力测试	332
Appendix A.	眼力测试参考答案.....	340
参考文献		349



PART 1

异常处理的现状

你对异常处理的了解有多少？

本书首先帮乡民们做一次简单的 YES or NO 的测验。以下有 20 题是非题，回答 YES 的数目越多，代表乡民们对于异常处理的了解越深入。请凭直觉回答以下问题，请准备好一支笔，计时 10 分钟，开始作答。

Q1 你曾经读过任何一本专门讨论异常处理的书籍吗？

Q2 你是否了解不良的异常处理，轻则造成用户不便，重则造成生命财产的损失？

Q3 你知道“缺陷”（fault）、“错误”（error）、“失效”（failure）、“异常”（exception）的差别吗？

Q4 你知道区分“设计缺陷”（design fault）和“组件缺陷”（component fault）对于异常处理有何帮助吗？

Q5 你听过面向对象语言异常处理机制的十大设计因素吗？

- Q6 你能说出 Ruby 和 Java/C# 的异常处理机制的最大差别吗？
-
- Q7 你知道异常处理设计的决策有哪些脉络(context)可以参考吗？
-
- Q8 你能够区分“异常处理”和“容错设计”的差别吗？
-
- Q9 你清楚 `NullPointerException` 或 `NullReferenceException` 的处理方法吗？
-
- Q10 你知道 `try-catch-finally` 各自负担何种责任吗？
-
- Q11 你能说出 Java 的“checked exception”（已查异常）与“unchecked exception”（未查异常）有何不同吗？
-
- Q12 你知道哪些 Java 异常类套用 *Exception Hierarchy* 设计模式，哪些又套用 *Smart Exception* 设计模式吗？
-
- Q13 你知道在敏捷方法中如何让异常处理设计逐步成长吗？
-
- Q14 你能够判断一个程序是否存在“异常处理坏味道”(exception handling bad smell) 吗？
-
- Q15 你会写“重试”(retry) 的异常处理程序吗？
-
- Q16 你知道一个函数如果只有一个 `try` 语句有什么好处吗？
-

Q17 你知道 C#和 Objective-C 的异常类为什么要实现 *Variable State* 设计模式吗？

Q18 你知道在数据库里面找不到数据要传回 null 还是丢出异常吗？

Q19 你知道异常处理程序执行失败该怎么办吗？

Q20 你知道花小钱却可大幅提升软件强健度的方法吗？

友藏内心独白：看完本书之后，就可以做到全部回答 YES。

01.

你对异常处理的了解有多少？