

---

---

# Refactoring to Patterns

Joshua Kerievsky

◆ Addison-Wesley

---

Boston • San Francisco • New York • Toronto • Montreal  
London • Munich • Paris • Madrid  
Capetown • Sydney • Tokyo • Singapore • Mexico City

# Contents

Foreword by Ralph Johnson .....	xv
Foreword by Martin Fowler .....	xvii
Preface .....	xix
What Is This Book About? .....	xix
What Are the Goals of This Book? .....	xix
Who Should Read This Book? .....	xx
What Background Do You Need? .....	xx
How to Use This Book .....	xxi
The History of This Book .....	xxii
Standing on the Shoulders of Giants .....	xxiii
Acknowledgments .....	xxiii
Chapter 1: Why I Wrote This Book .....	1
Over-Engineering .....	1
The Patterns Panacea .....	2
Under-Engineering .....	3
Test-Driven Development and Continuous Refactoring .....	4
Refactoring and Patterns .....	6
Evolutionary Design .....	8
Chapter 2: Refactoring .....	9
What Is Refactoring? .....	9
What Motivates Us to Refactor? .....	10
Many Eyes .....	11
Human-Readable Code .....	12



Keeping It Clean . . . . .	13
Small Steps . . . . .	14
Design Debt . . . . .	15
Evolving a New Architecture . . . . .	16
Composite and Test-Driven Refactorings . . . . .	17
The Benefits of Composite Refactorings . . . . .	19
Refactoring Tools . . . . .	20
<b>Chapter 3: Patterns . . . . .</b>	<b>23</b>
What Is a Pattern? . . . . .	23
Patterns Happy . . . . .	24
There Are Many Ways to Implement a Pattern . . . . .	26
Refactoring to, towards, and away from Patterns . . . . .	29
Do Patterns Make Code More Complex? . . . . .	31
Pattern Knowledge . . . . .	32
Up-Front Design with Patterns . . . . .	33
<b>Chapter 4: Code Smells . . . . .</b>	<b>37</b>
Duplicated Code . . . . .	39
Long Method . . . . .	40
Conditional Complexity . . . . .	41
Primitive Obsession . . . . .	41
Indecent Exposure . . . . .	42
Solution Sprawl . . . . .	43
Alternative Classes with Different Interfaces . . . . .	43
Lazy Class . . . . .	43
Large Class . . . . .	44
Switch Statements . . . . .	44
Combinatorial Explosion . . . . .	45
Oddball Solution . . . . .	45
<b>Chapter 5: A Catalog of Refactorings to Patterns. . . . .</b>	<b>47</b>
Format of the Refactorings . . . . .	47
Projects Referenced in This Catalog . . . . .	49
XML Builders . . . . .	50
HTML Parser . . . . .	50
Loan Risk Calculator . . . . .	51
A Starting Point . . . . .	51
A Study Sequence . . . . .	52



<b>Chapter 6: Creation</b> .....	<b>55</b>
Replace Constructors with Creation Methods .....	57
Motivation .....	57
Mechanics .....	60
Example .....	60
Variations .....	65
Move Creation Knowledge to Factory .....	68
Motivation .....	69
Mechanics .....	72
Example .....	73
Encapsulate Classes with Factory .....	80
Motivation .....	81
Mechanics .....	82
Example .....	83
Variations .....	86
Introduce Polymorphic Creation with Factory Method .....	88
Motivation .....	89
Mechanics .....	90
Example .....	92
Encapsulate Composite with Builder .....	96
Mechanics .....	99
Example .....	100
Variations .....	111
Inline Singleton .....	114
Motivation .....	115
Mechanics .....	117
Example .....	118
<b>Chapter 7: Simplification</b> .....	<b>121</b>
Compose Method .....	123
Motivation .....	123
Mechanics .....	125
Example .....	126
Replace Conditional Logic with Strategy .....	129
Motivation .....	130
Mechanics .....	131
Example .....	133



Move Embellishment to Decorator . . . . .	144
Motivation . . . . .	144
Mechanics . . . . .	148
Example . . . . .	150
Replace State-Altering Conditionals with State . . . . .	166
Motivation . . . . .	167
Mechanics . . . . .	168
Example . . . . .	169
Replace Implicit Tree with Composite . . . . .	178
Motivation . . . . .	179
Mechanics . . . . .	181
Example . . . . .	183
Replace Conditional Dispatcher with Command . . . . .	191
Motivation . . . . .	192
Mechanics . . . . .	193
Example . . . . .	195
<b>Chapter 8: Generalization . . . . .</b>	<b>203</b>
Form Template Method . . . . .	205
Motivation . . . . .	206
Mechanics . . . . .	207
Example . . . . .	208
Extract Composite . . . . .	214
Motivation . . . . .	215
Mechanics . . . . .	215
Example . . . . .	216
Replace One/Many Distinctions with Composite . . . . .	224
Motivation . . . . .	225
Mechanics . . . . .	227
Example . . . . .	228
Replace Hard-Coded Notifications with Observer . . . . .	236
Motivation . . . . .	236
Mechanics . . . . .	238
Example . . . . .	239
Unify Interfaces with Adapter . . . . .	247
Motivation . . . . .	248
Mechanics . . . . .	249
Example . . . . .	250

Extract Adapter .....	258
Motivation .....	259
Mechanics .....	261
Example .....	261
Variations .....	268
Replace Implicit Language with Interpreter .....	269
Motivation .....	270
Mechanics .....	272
Example .....	273
<b>Chapter 9: Protection</b> .....	<b>285</b>
Replace Type Code with Class .....	286
Motivation .....	287
Mechanics .....	288
Example .....	290
Limit Instantiation with Singleton .....	296
Motivation .....	296
Mechanics .....	297
Example .....	298
Introduce Null Object .....	301
Motivation .....	302
Mechanics .....	304
Example .....	305
<b>Chapter 10: Accumulation</b> .....	<b>311</b>
Move Accumulation to Collecting Parameter .....	313
Motivation .....	313
Mechanics .....	315
Example .....	315
Move Accumulation to Visitor .....	320
Motivation .....	321
Mechanics .....	325
Example .....	330
<b>Chapter 11: Utilities</b> .....	<b>339</b>
Chain Constructors .....	340
Motivation .....	341
Mechanics .....	341
Example .....	341



## CONTENTS

Unify Interfaces .....	343
Motivation .....	344
Mechanics .....	344
Example .....	345
Extract Parameter .....	346
Motivation .....	346
Mechanics .....	347
Example .....	347
<b>Afterword by John Brant and Don Roberts .....</b>	<b>349</b>
<b>References .....</b>	<b>351</b>
<b>Index .....</b>	<b>355</b>