

What Is Refactoring?	2
What's in this Book	3
Who Should Read this Book?	5
Building on the Foundations Laid by Others	5
Chapter 1: Refactoring, a first example	1
The Starting Point	2
The First Step in Refactoring	10
Decomposing and Redistributing the Statement Method	11
Replacing the Conditional Logic on Price Code with Polymorphism	38
Final Thoughts	56
Chapter 2:	
Principles in Refactoring	57
A Definition of Refactoring	57
Why Should You Refactor?	59
Problems with Refactoring	61
When Shouldn't you Refactor?	68
Refactoring and Up Front Design	69
Refactoring and Evolutionary Development	71
Pair Programming	72
Refactoring and Performance	74
Where did refactoring come from?	77
Tools for Refactoring	80
Chapter 3: Bad Smells in Code	83
Duplicated code	84
Long method	85
Large Class	86
Long Parameter List	87
Divergent Change	88
Shotgun Surgery	88
Feature Envy	89
Data Clumps	89
Case Statements	90
Parallel Inheritance Hierarchies	91
Lazy Class	91
Similar subclasses	91



Temporary Field	92
Middle Man	92
Alternative classes with different interfaces	92
Comments	93
Chapter 4: Building Tests	95
The Value of Self Testing Code	95
The JUnit Testing Framework	98
Adding More Tests	105
Chapter 5: Towards a Catalog of Refactorings	109
Chapter 6: Dealing with Long Methods	113
Extract Method	114
Inline Method	120
Inline Temp	121
Split Temporary Variable	124
Remove Assignments to Parameters	126
Replace Method with Method Object	130
Substitute Algorithm	132
Chapter 7: Simplifying Conditional Expressions	135
Decompose Conditional	136
Consolidate Conditional Expression	137
Consolidate duplicate conditional fragments	140
Remove Control Flag	141
Replace Nested Conditional with Guard Clauses	145
Replace Switch with Polymorphism	147
Introduce Null Object	151
Chapter 8: Moving Features between Objects	159
Move Method	160
Move Field	164
Extract Component	166
Inline Component	170
Hide Delegate	172
Remove Middle Man	174
Create Foreign Method	176
Create Extension	178
Chapter 9: Organizing Data	183
Self Encapsulate Field	184

Replace Data Value with Object	187
Change Value to Reference	191
Replace Array with Object	194
Duplicate Data From Presentation to Domain	197
Change Unidirectional Association to Bidirectional	204
Change Bidirectional Association to Unidirectional	207
Replace Magic Number with Symbolic Constant	210
Encapsulate Field	211
Encapsulate Collection	212
Replace Record with Data Class	218
Replace Type Code with Class	219
Replace Type Code with Subclasses	224
Replace Type Code with State/Strategy	227
Chapter 10: Making Method Calls Simpler	233
Rename Method	234
Separate Query from Modifier	236
Parameterize Method	240
Preserve Whole Object	241
Replace Parameter with Method	245
Introduce Parameter Object	247
Remove Setting Method	252
Hide Method	255
Replace Constructor with Factory Method	256
Encapsulate Downcast	258
Replace Error Code With Exception	259
Replace Exception With Test	264
Chapter 11: Dealing with Generalization	269
Pull Up Field	270
Pull Up Method	271
Pull up Constructor Body	273
Push Down Method	276
Push Down Field	276
Extract Subclass	277
Extract Superclass	282
Extract Interface	286
Collapse Hierarchy	288



Form Template Method	289
Replace Inheritance with Delegation	295
Replace Delegation with Inheritance	297
Chapter 12: Others	301
Add Invariant	301
Replace Program With Class	305
Chapter 13: Refactoring, Reuse & Reality	307
A Reality Check	308
Why are Developers Reluctant to Refactor Their Programs?	310
Understanding How & Where to Refactor	311
Refactoring to Achieve Near-Term Benefits	314
Refactoring Safely	317
A Reality Check (Revisited)	322
Implications Regarding Software Reuse	323
A Final Note	324
References:	325
Chapter 14: Refactoring with Tool Support	329
The Starting Program	330
The Refactoring	333
Chapter 15: A Longer Example	355
The Initial Program	356
The First Step in Refactoring	369
Starting a Hierarchy of Sites	372
Simplifying the Charge Methods	386
Decomposing Site's Long Methods	407
Adding Lifeline Site to the Hierarchy	429
Adding Business Site to the Hierarchy	442
References	453