Contents

Foreword xi

Acknowledgments xiii

Welcome! xv

1. **Start Cleaning Up** 1
   - Avoid Unnecessary Comparisons 2
   - Avoid Negations 4
   - Return Boolean Expressions Directly 6
   - Simplify Boolean Expressions 8
   - Avoid NullPointerException in Conditionals 10
   - Avoid Switch Fallthrough 12
   - Always Use Braces 14
   - Ensure Code Symmetry 16
   - What Have You Learned? 18

2. **Level Up Your Code Style** 19
   - Replace Magic Numbers with Constants 20
   - Favor Enums Over Integer Constants 22
   - Favor For-Each Over For Loops 24
   - Avoid Collection Modification During Iteration 26
   - Avoid Compute-Intense Operations During Iteration 28
   - Group with New Lines 30
   - Favor Format Over Concatenation 32
   - Favor Java API Over DIY 34
   - What Have You Learned? 36

3. **Use Comments Wisely** 37
   - Remove Superfluous Comments 38
   - Remove Commented-Out Code 40
   - Replace Comments with Constants 42

Replace Comments with Utility Methods 44
Document Implementation Decisions 46
Document Using Examples 48
Structure JavaDoc of Packages 50
Structure JavaDoc of Classes and Interfaces 52
Structure JavaDoc of Methods 54
Structure JavaDoc of Constructors 56
What Have You Learned? 58

4. **Name Things Right** 59
Use Java Naming Conventions 60
Follow Getter/Setter Conventions for Frameworks 62
Avoid Single-Letter Names 64
Avoid Abbreviations 66
Avoid Meaningless Terms 68
Use Domain Terminology 70
What Have You Learned? 72

5. **Prepare for Things Going Wrong** 73
Fail Fast 74
Always Catch Most Specific Exception 76
Explain Cause in Message 78
Avoid Breaking the Cause Chain 80
Expose Cause in Variable 82
Always Check Type Before Cast 84
Always Close Resources 86
Always Close Multiple Resources 88
Explain Empty Catch 90
What Have You Learned? 92

6. **Assert Things Going Right** 93
Structure Tests Into Given-When-Then 94
Use Meaningful Assertions 96
Expected Before Actual Value 98
Use Reasonable Tolerance Values 100
Let JUnit Handle Exceptions 102
Describe Your Tests 104
Favor Standalone Tests 106
Parametrize Your Tests 108
Cover the Edge Cases 110
What Have You Learned? 112

7. Design Your Objects 113
Split Method with Boolean Parameters 114
Split Method with Optional Parameters 116
Favor Abstract Over Concrete Types 118
Favor Immutable Over Mutable State 120
Combine State and Behavior 122
Avoid Leaking References 124
Avoid Returning Null 126
What Have You Learned? 128

8. Let Your Data Flow 129
Favor Lambdas Over Anonymous Classes 130
Favor Functional Over Imperative Style 132
Favor Method References Over Lambdas 134
Avoid Side Effects 136
Use Collect for Terminating Complex Streams 138
Avoid Exceptions in Streams 140
Favor Optional Over Null 142
Avoid Optional Fields or Parameters 144
Use Optionals as Streams 146
What Have You Learned? 148

9. Prepare for the Real World 149
Use Static Code Analysis Tools 150
Agree On the Java Format in Your Team 152
Automate Your Build 153
Use Continuous Integration 154
Prepare for and Deliver Into Production 155
Favor Logging Over Console Output 156
Minimize and Isolate Multithreaded Code 158
Use High-Level Concurrency Abstractions 159
Speed Up Your Program 160
Know Your Falsehoods 162
What Have You Learned? 164

Bibliography 165
Index 167