

Robert C. Martin Series

PRENTICE
HALL
PTR

Agile Java™

Crafting Code
with Test-Driven
Development

```
private Object monitor = new Object();

public void testClock() throws Exception {
    final int seconds = 2;
    final List<Date> tics = new ArrayList<>();
    ClockListener listener = new ClockListener() {
        private int count = 0;
        public void update(Date date) {
            tics.add(date);
            if (++count == seconds) {
                synchronized(monitor) {
                    monitor.notifyAll();
                }
            }
        }
    };
    Clock clock = new Clock(listener);
    synchronized(monitor) {
        monitor.wait();
    }
    clock.stop();
    verify(tics, seconds);
}
```

Foreword by Ron Jeffries

Jeff Langr

Contents

About the Author	.xix
Foreword	.xxi
Acknowledgments	.xxiii
Introduction	1
Who Is This Book For?	3
What This Book Is Not	4
How to Use This Book	5
Exercises	6
Conventions Used in This Book	6
An Agile Overview	9
What Is “Agile?”	9
What Is Java?	11
Why OO?	13
What Is an Object?	13
What Are Classes?	14
Why UML?	16
What Is Inheritance?	17
Why Test-Driven Development?	18
Setting Up	21
Software You’ll Need	21
Does It Work?	26
Compiling Hello World	27
Executing Hello World	27
Still Stuck?	28
Lesson 1: Getting Started	31
Testing	31
Design	32

A Simple Test	33
JUnit	35
Adding a Test	37
Creating a Student	39
Creating the Student Class	40
Constructors	41
Local Variables	42
Returning a Value from a Method	44
Assertions	46
Instance Variables	49
Summarizing the Test	52
Refactoring	53
this	55
private	57
Naming Conventions	59
Whitespace	61
Exercises	62
 Lesson 2: Java Basics	63
CourseSession	63
Enrolling Students	65
int	65
Initialization	68
Default Constructors	69
Suites	70
The SDK and java.util.ArrayList	71
Adding Objects	73
Incremental Refactoring	75
Objects in Memory	76
Packages and the import Statement	78
The java.lang Package	79
The Default Package and the package Statement	79
The setup Method	81
More Refactoring	83
Class Constants	84
Dates	86
Overload Constructors	86
Deprecation Warnings	91

Refactoring	92
Creating Dates with Calendar	95
Comments	96
Javadoc Comments	97
Exercises	100
Lesson 3: Strings and Packages	103
Characters and Strings	103
Strings	106
StringBuilder	107
System Properties	109
Looping Through All Students	110
Single-Responsibility Principle	112
Refactoring	115
System.out	118
Using System.out	119
Refactoring	120
Package Structure	121
Access Modifiers	122
Using Ant	128
Exercises	131
Lesson 4: Class Methods and Fields	133
Class Methods	133
Class Variables	137
Operating on Class Variables with Class Methods	138
Static Import	141
Incrementing	144
Factory Methods	145
Simple Design	147
Static Dangers	147
Using Statics: Various Notes	148
Jeff's Rule of Statics	149
Booleans	150
Tests as Documentation	154
More on Initialization	157
Exceptions	158
Revisiting Primitive-Type Field Initialization	159
Exercises	160

Lesson 5: Interfaces and Polymorphism	163
Sorting: Preparation	163
Sorting: Collections.sort	164
CourseReportTest	165
Interfaces	166
Why Interfaces	168
Implementing Comparable	169
Sorting on Department and Number	171
The if Statement	172
Grading Students	173
Floating-Point Numbers	173
Testing Grades	175
Refactoring	178
Enums	179
Polymorphism	181
Using Interface References	187
ArrayList and the List Interface	189
Exercises	189
 Lesson 6: Inheritance	 195
The switch Statement	195
Case Labels Are Just Labels	196
Maps	199
Inheritance	201
Abstract Classes	204
Extending Methods	206
Refactoring	207
Enhancing the Grade Enum	209
Summer Course Sessions	210
Calling Superclass Constructors	211
Refactoring	215
More on Constructors	218
Inheritance and Polymorphism	220
The Principle of Subcontracting	220
Exercises	228
 Lesson 7: Legacy Elements	 231
Looping Constructs	232
Breaking Up a Student's Name	232

The while Loop	234
Comparing Java Loops	240
Refactoring	241
Looping Control Statements	242
The Ternary Operator	245
Legacy Collections	246
Iterators	247
Iterators and the for-each Loop	249
Casting	250
Wrapper Classes	252
Arrays	255
Refactoring	264
Exercises	266
Lesson 8: Exceptions and Logging	269
Exceptions	270
Dealing With Exceptions	272
Checked Exceptions	273
Exception Hierarchy	275
Creating Your Own Exception Type	276
Checked Exceptions vs. Unchecked Exceptions	278
Messages	279
Catching Multiple Exceptions	280
Rethrowing Exceptions	282
Stack Traces	285
The finally Block	285
Refactoring	287
Logging	290
Logging in Java	291
Testing Logging	294
Logging to Files	298
Testing Philosophy for Logging	300
More on FileHandler	302
Logging Levels	302
Logging Hierarchies	304
Additional Notes on Logging	305
Exercises	305
Lesson 9: Maps and Equality	311
Logical Operators	311

Short-Circuiting	313
Hash Tables	313
Courses	315
Refactoring Session	317
Equality	323
The Contract for Equality	326
Apples and Oranges	327
Collections and Equality	329
Hash Tables	330
Collisions	332
An Ideal Hash Algorithm	333
A Final Note on hashCode	335
More on Using HashMaps	337
Additional Hash Tables and Set Implementations	341
toString	343
Strings and Equality	345
Exercises	346
Lesson 10: Mathematics	349
BigDecimal	350
More on Primitive Numerics	353
Integer Math	354
Numeric Casting	355
Expression Evaluation Order	356
NaN	357
Infinity	358
Numeric Overflow	359
Bit Manipulation	360
Java.lang.Math	368
Numeric Wrapper Classes	370
Random Numbers	371
Exercises	375
Lesson 11: IO	379
Organization	379
Character Streams	380
Writing to a File	385
Java.io.File	388
Byte Streams and Conversion	389
A Student User Interface	390

Testing the Application	393
Data Streams	395
CourseCatalog	396
Advanced Streams	399
Object Streams	399
Random Access Files	407
The Student Directory	408
sis.db.DataFileTest	410
Static Nested Classes and Inner Classes	412
sis.db.DataFile	413
sis.db.KeyFileTest	416
sis.db.KeyFile	418
sis.util.IOUtilTest	419
sis.util.IOUtil	420
sis.util.TestUtil	421
Developing the Solution	422
Exercises	422

Lesson 12: Reflection and Other Advanced Topics	425
Mock Objects Revisited	425
The Jim Bob ACH Interface	427
The Mock Class	429
The Account Class Implementation	431
Anonymous Inner Classes	433
Adapters	435
Accessing Variables from the Enclosing Class	437
Tradeoffs	439
Reflection	440
Using JUnit Code	441
The Class Class	442
Building the Suite	444
Class Modifiers	446
Dynamic Proxy	448
A Secure Account Class	449
Building the Secure Account Solution	451
The SecureProxy Class	456
Problems With Reflection	459
Exercises	459

Lesson 13: Multithreading	461
Multithreading	462
Search Server	462
The Search Class	463
Less Dependent Testing	466
The Server	470
Waiting in the Test	472
Creating and Running Threads	473
Cooperative and Preemptive Multitasking	477
Synchronization	478
Creating Threads with Runnable	480
Synchronized	481
Synchronized Collections	482
BlockingQueue	483
Stopping Threads	485
Wait/Notify	486
Additional Notes on wait and notify	491
Locks and Conditions	492
Thread Priorities	494
Deadlocks	495
ThreadLocal	495
The Timer Class	499
Thread Miscellany	501
Summary: Basic Design Principles for Synchronization	506
Exercises	506
 Lesson 14: Generics	 509
Parameterized Types	509
Collection Framework	510
Multiple Type Parameters	511
Creating Parameterized Types	511
Erasure	514
Upper Bounds	516
Wildcards	518
Implications of Using Wildcards	520
Generic Methods	522
Wildcard Capture	522
Super	523

Additional Bounds	525
Raw Types	526
Checked Collections	528
Arrays	530
Additional Limitations	531
Reflection	532
Final Notes	532
Exercises	533
Lesson 15: Assertions and Annotations	535
Assertions	535
The assert Statement vs. JUnit Assert Methods	537
Annotations	537
Building a Testing Tool	538
TestRunnerTest	539
TestRunner	540
The @TestMethod Annotation	543
Retention	545
Annotation Targets	546
Skipping Test Methods	548
Modifying TestRunner	548
Single-Value Annotations	550
A TestRunner User Interface Class	552
Array Parameters	553
Multiple Parameter Annotations	555
Default Values	557
Additional Return Types and Complex Annotation Types	558
Package Annotations	559
Compatibility Considerations	561
Additional Notes on Annotations	561
Summary	562
Exercises	562
Additional Lesson I: Swing, Part 1	565
Swing	566
Getting Started	567
Swing Application Design	571
Panels	572
Refactoring	576

More Widgets	580
Refactoring	583
Button Clicks and ActionListeners	586
List Models	588
The Application	591
Layout	594
GridBagLayout	602
Moving Forward	606
Additional Lesson II: Swing, Part 2	607
Miscellaneous Aesthetics	608
Feel	612
Tables	635
Feedback	640
Responsiveness	647
Remaining Tasks	650
Final Notes	651
Additional Lesson III: Java Miscellany	653
JARs	653
Regular Expressions	656
Cloning and Covariance	662
JDBC	664
Internationalization	673
Call by Reference versus Call by Value	681
Java Periphery	683
What Else Is There?	693
Appendix A: An Agile Java Glossary	701
Appendix B: Java Operator Precedence Rules	715
Appendix C: Getting Started with IDEA	717
IDEA	717
The Hello Project	718
Running Tests	724
Taking Advantage of IDEA	729
Agile Java References	733
Index	735