

Java 9 Recipes

A Problem-Solution Approach

Third Edition



Josh Juneau

Apress®

Java 9 Recipes: A Problem-Solution Approach

Josh Juneau
Hinckley, Illinois, USA

ISBN-13 (pbk): 978-1-4842-1975-1
DOI 10.1007/978-1-4842-1976-8

ISBN-13 (electronic): 978-1-4842-1976-8

Library of Congress Control Number: 2017943502

Copyright © 2017 by Josh Juneau

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr
Editorial Director: Todd Green
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Technical Reviewer: Vinay Kumar
Coordinating Editor: Jill Balzano
Copy Editor: Brendan Frost
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484219751. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

This book is dedicated to my wife and children.

Contents at a Glance

About the Author	xliii
About the Technical Reviewer	xliv
Acknowledgments	xlvii
Introduction	xlix
■ Chapter 1: Getting Started with Java 9	1
■ Chapter 2: Java 9 Enhancements	35
■ Chapter 3: Strings.....	47
■ Chapter 4: Numbers and Dates.....	65
■ Chapter 5: Object-Oriented Java	99
■ Chapter 6: Lambda Expressions	137
■ Chapter 7: Data Sources and Collections	159
■ Chapter 8: Input and Output	195
■ Chapter 9: Exceptions and Logging	221
■ Chapter 10: Concurrency	239
■ Chapter 11: Debugging and Unit Testing	267
■ Chapter 12: Unicode, Internationalization, and Currency Codes	285
■ Chapter 13: Working with Databases	305
■ Chapter 14: JavaFX Fundamentals	355
■ Chapter 15: Graphics with JavaFX	429
■ Chapter 16: Media with JavaFX.....	471

■ **Chapter 17: Java Web Applications with JavaServer Faces..... 499**

■ **Chapter 18: Nashorn and Scripting 529**

■ **Chapter 19: E-mail..... 553**

■ **Chapter 20: JSON and XML Processing 563**

■ **Chapter 21: Networking 585**

■ **Chapter 22: Java Modularity 605**

Index..... 615

Contents

About the Author xliii

About the Technical Reviewerxlv

Acknowledgmentsxlvii

Introductionxlix

■ Chapter 1: Getting Started with Java 9 1

 1-1. Creating a Development Environment 1

 Problem 1

 Solution..... 1

 How It Works..... 2

 1-2. Getting to “Hello, World” 3

 Problem 3

 Solution..... 3

 How It Works..... 7

 1-3. Configuring the CLASSPATH..... 9

 Problem 9

 Solution..... 9

 How It Works..... 10

 1-4. Organizing Code with Packages 11

 Problem 11

 Solution..... 11

 How It Works..... 12

 1-5. Declaring Variables and Access Modifiers 14

 Problem 14

 Solution..... 14

 How It Works..... 16

- 1-6. Compiling and Executing from the Command-Line or Terminal Interpreter 17
 - Problem 17
 - Solution..... 17
 - How It Works..... 18
- 1-7. Developing Within the Interactive jShell 19
 - Problem 19
 - Solution..... 20
 - How It Works..... 22
- 1-8. Converting to and from a String..... 24
 - Problem 24
 - Solution..... 24
 - How It Works..... 24
- 1-9. Passing Arguments via Command-Line Execution 25
 - Problem 25
 - Solution..... 25
 - How It Works..... 26
- 1-10. Executing a Script via the jShell 27
 - Problem 27
 - Solution..... 27
 - How It Works..... 27
- 1-11. Accepting Input from the Keyboard 28
 - Problem 28
 - Solution..... 28
 - How It Works..... 29
- 1-12. Documenting Your Code..... 30
 - Problem 30
 - Solution..... 30
 - How It Works..... 31

1-13. Reading Environment Variables	33
Problem	33
Solution.....	33
How It Works.....	34
Summary.....	34
■ Chapter 2: Java 9 Enhancements	35
2-1. Avoiding Redundancy in Interface Code	35
Problem	35
Solution.....	35
How It Works.....	36
2-2. Creating Modules for Simplifying and Code Reuse.....	36
Problem	36
Solution.....	36
How It Works.....	38
2-3. Easily Retrieving Information on OS Processes	38
Problem	38
Solution.....	38
How It Works.....	40
2-4. Handling Errors with Ease	40
Problem	40
Solution.....	41
How It Works.....	41
2-5. Filtering Data Before and After a Condition with Streams	42
Problem	42
Solution.....	42
How It Works.....	42
2-6. Developing a Concise HTTP Client	43
Problem	43
Solution.....	43
How It Works.....	44

2-7. Redirecting Platform Logs	44
Problem	44
Solution.....	44
How It Works.....	45
2-8. Utilizing Factory Methods to Create Immutable Collections	45
Problem	45
Solution.....	45
How It Works.....	46
Summary	46
■ Chapter 3: Strings.....	47
Compact Strings: Java 9 String Enhancements	47
3-1. Obtaining a Subsection of a String	47
Problem	47
Solution.....	47
How It Works.....	48
3-2. Comparing Strings	48
Problem	48
Solution.....	48
How It Works.....	50
3-3. Trimming Whitespace.....	51
Problem	51
Solution.....	51
How It Works.....	52
3-4. Changing the Case of a String	52
Problem	52
Solution.....	52
How It Works.....	52
3-5. Concatenating Strings	53
Problem	53
Solution 1.....	53

Solution 2.....	53
Solution 3.....	54
How It Works.....	54
3-6. Converting Strings to Numeric Values	55
Problem	55
Solution 1.....	55
Solution 2.....	55
How It Works.....	55
3-7. Iterating Over the Characters of a String	56
Problem	56
Solution.....	56
How It Works.....	57
3-8. Finding Text Matches	58
Problem	58
Solution 1.....	58
Solution 2.....	59
How It Works.....	59
3-9. Replacing All Text Matches	60
Problem	60
Solution.....	61
How It Works.....	61
3-10. Determining Whether a File Suffix Matches a Given String	61
Problem	61
Solution.....	62
How It Works.....	62
3-11. Making a String That Can Contain Dynamic Information	62
Problem	62
Solution 1.....	62
Solution 2.....	63
How It Works.....	63
Summary	64

■ **Chapter 4: Numbers and Dates..... 65**

4-1. Rounding Float and Double Values to Integers 66

 Problem 66

 Solution..... 66

 How It Works..... 66

4-2. Formatting Double and Long Decimal Values 67

 Problem 67

 Solution..... 67

 How It Works..... 68

4-3. Comparing int Values 69

 Problem 69

 Solution 1..... 69

 Solution 2..... 69

 How It Works..... 70

4-4. Comparing Floating-Point Numbers..... 70

 Problem 70

 Solution 1..... 70

 Solution 2..... 71

 How It Works..... 71

4-5. Calculating Monetary Values 71

 Problem 71

 Solution 1..... 71

 Solution 2..... 72

 How It Works..... 73

4-6. Randomly Generating Values 74

 Problem 74

 Solution 1..... 74

 Solution 2..... 75

 How It Works..... 75

4-7. Obtaining the Current Date Without Time	76
Problem	76
Solution.....	76
How It Works.....	76
4-8. Obtaining a Date Object Given Date Criteria	76
Problem	76
Solution.....	76
How It Works.....	77
4-9. Obtaining a Year-Month-Day Date Combination.....	77
Problem	77
Solution 1.....	77
Solution 2.....	77
How It Works.....	78
4-10. Obtaining and Calculating Times Based on the Current Time	78
Problem	78
Solution.....	78
How It Works.....	79
4-11. Obtaining and Using the Date and Time Together	80
Problem	80
Solution 1.....	80
Solution 2.....	82
Solution 3.....	82
How It Works.....	83
4-12. Obtaining a Machine Timestamp	84
Problem	84
Solution.....	84
How It Works.....	85
4-13. Converting Dates and Times Based on the Time Zone.....	85
Problem	85
Solution.....	85
How It Works.....	87

4-14. Comparing Two Dates 88
 Problem 88
 Solution..... 88
 How It Works..... 89

4-15. Finding the Interval Between Dates and Times 89
 Problem 89
 Solution 1..... 89
 Solution 2..... 90
 Solution 3..... 91
 How It Works..... 91

4-16. Obtaining Date-Time from a Specified String 92
 Problem 92
 Solution..... 92
 How It Works..... 93

4-17. Formatting Dates for Display 94
 Problem 94
 Solution 1..... 94
 Solution 2..... 94
 How It Works..... 95

4-18. Writing Readable Numeric Literals 97
 Problem 97
 Solution..... 97
 How It Works..... 97

4-19. Declaring Binary Literals..... 97
 Problem 97
 Solution..... 98
 How It Works..... 98

Summary 98

■ Chapter 5: Object-Oriented Java	99
5-1. Controlling Access to Members of a Class.....	99
Problem	99
Solution.....	99
How It Works.....	100
5-2. Making Private Fields Accessible to Other Classes	100
Problem	100
Solution.....	100
How It Works.....	100
5-3. Creating a Class with a Single Instance	101
Problem	101
Solution 1.....	101
Solution 2.....	103
How It Works.....	103
5-4. Generating Instances of a Class	104
Problem	104
Solution.....	104
How It Works.....	106
5-5. Creating Reusable Objects.....	106
Problem	106
Solution.....	106
How It Works.....	107
5-6. Defining an Interface for a Class	108
Problem	108
Solution.....	108
How It Works.....	108
5-7. Modifying Interfaces Without Breaking Existing Code	110
Problem	110
Solution.....	110
How It Works.....	111

- 5-8. Constructing Instances of the Same Class with Different Values 112
 - Problem 112
 - Solution..... 112
 - How It Works..... 115
- 5-9. Interacting with a Class via Interfaces..... 116
 - Problem 116
 - Solution..... 116
 - How It Works..... 116
- 5-10. Making a Class Cloneable..... 117
 - Problem 117
 - Solution..... 117
 - How It Works..... 121
- 5-11. Comparing Objects 121
 - Problem 121
 - Solution 1..... 121
 - Solution 2..... 122
 - How It Works..... 125
- 5-12. Extending the Functionality of a Class..... 126
 - Problem 126
 - Solution..... 126
 - How It Works..... 129
- 5-13. Defining a Template for Classes to Extend..... 129
 - Problem 129
 - Solution..... 129
 - How It Works..... 130
- 5-14. Increasing Class Encapsulation 131
 - Problem 131
 - Solution..... 131
 - How It Works..... 134
- Summary..... 136

■ Chapter 6: Lambda Expressions	137
6-1. Writing a Simple Lambda Expression	137
Problem	137
Solution.....	137
How It Works.....	138
6-2. Enabling the Use of Lambda Expressions.....	140
Problem	140
Solution 1.....	140
Solution 2.....	140
How It Works.....	141
6-3. Invoking Existing Methods by Name.....	143
Problem	143
Solution.....	144
How It Works.....	145
6-4. Sorting with Fewer Lines of Code.....	147
Problem	147
Solution 1.....	148
Solution 2.....	148
How It Works.....	149
6-5. Filtering a Collection of Data.....	149
Problem	149
Solution.....	150
How It Works.....	150
6-6. Implementing Runnable.....	150
Problem	150
Solution.....	150
How It Works.....	151
6-7. Replacing Anonymous Inner Classes	151
Problem	151
Solution.....	151
How It Works.....	152

6-8. Accessing Class Variables from a Lambda Expression	153
Problem	153
Solution.....	153
How It Works.....	154
6-9. Passing Lambda Expressions to Methods	155
Problem	155
Solution.....	155
How It Works.....	157
Summary	157
■ Chapter 7: Data Sources and Collections	159
7-1. Defining a Fixed Set of Related Constants.....	159
Problem	159
Solution.....	159
How It Works.....	161
7-2. Designing Intelligent Constants	162
Problem	162
Solution.....	162
How It Works.....	165
7-3. Executing Code Based on a Specified Value	166
Problem	166
Solution.....	166
How It Works.....	169
7-4. Working with Fix-Sized Arrays	170
Problem	170
Solution.....	170
How It Works.....	172
7-5. Safely Enabling Types or Methods to Operate on Objects of Various Types	174
Problem	174
Solution.....	175
How It Works.....	176

7-6. Working with Dynamic Arrays	180
Problem	180
Solution.....	180
How It Works.....	182
7-7. Making Your Objects Iterable	183
Problem	183
Solution.....	183
How It Works.....	185
7-8. Iterating Over Collections.....	186
Problem	186
Solution.....	186
How It Works.....	188
7-9. Iterating Over a Map	189
Problem	189
Solution.....	189
How It Works.....	190
7-10. Executing Streams in Parallel	192
Problem	192
Solution.....	192
How It Works.....	193
Summary	193
■ Chapter 8: Input and Output	195
8-1. Serializing Java Objects.....	196
Problem	196
Solution.....	196
How It Works.....	197
8-2. Serializing Java Objects More Efficiently.....	199
Problem	199
Solution.....	199
How It Works.....	199

8-3. Serializing Java Objects as XML	200
Problem	200
Solution.....	200
How It Works.....	201
8-4. Creating a Socket Connection and Sending Serializable Objects Across the Wire	202
Problem	202
Solution.....	202
How It Works.....	204
8-5. Obtaining the Java Execution Path	205
Problem	205
Solution.....	205
How It Works.....	205
8-6. Copying a File	205
Problem	205
Solution.....	206
How It Works.....	206
8-7. Moving a File.....	206
Problem	206
Solution.....	206
How It Works.....	207
8-8. Creating a Directory.....	207
Problem	207
Solution 1.....	207
Solution 2.....	207
How It Works.....	208
8-9. Iterating Over Files in a Directory	208
Problem	208
Solution.....	208
How It Works.....	209

8-10. Querying (and Setting) File Metadata	209
Problem	209
Solution.....	209
How It Works.....	210
8-11. Monitoring a Directory for Content Changes.....	211
Problem	211
Solution.....	211
How It Works.....	212
8-12. Reading Property Files.....	212
Problem	212
Solution.....	213
How It Works.....	214
8-13. Uncompressing Files	214
Problem	214
Solution.....	214
How It Works.....	215
8-14. Managing Operating System Processes	216
Problem	216
Solution.....	216
How It Works.....	217
Summary	219
■ Chapter 9: Exceptions and Logging	221
9-1. Catching Exceptions	222
Problem	222
Solution.....	222
How It Works.....	223
9-2. Guaranteeing a Block of Code Is Executed	223
Problem	223
Solution.....	224
How It Works.....	224

9-3. Throwing Exceptions..... 224
 Problem 224
 Solution..... 225
 How It Works..... 225

9-4. Catching Multiple Exceptions..... 225
 Problem 225
 Solution 1..... 225
 Solution 2..... 226
 How It Works..... 226

9-5. Catching the Uncaught Exceptions 227
 Problem 227
 Solution 1..... 227
 Solution 2..... 227
 How It Works..... 228

9-6. Managing Resources with try/catch Blocks 228
 Problem 228
 Solution..... 228
 How It Works..... 229

9-7. Creating an Exception Class 229
 Problem 229
 Solution 1..... 229
 Solution 2..... 230
 How It Works..... 230

9-8. Rethrowing the Caught Exception..... 230
 Problem 230
 Solution..... 230
 How It Works..... 231

9-9. Logging Events Within Your Application 231
 Problem 231
 Solution..... 231
 How It Works..... 232

9-10. Rotating and Purging Logs.....	233
Problem	233
Solution.....	233
How It Works.....	233
9-11. Logging Exceptions.....	234
Problem	234
Solution.....	234
How It Works.....	235
9-12. Logging with the Unified JVM Logger	235
Problem	235
Solution.....	235
How It Works.....	236
Summary	237
■ Chapter 10: Concurrency	239
10-1. Starting a Background Task.....	239
Problem	239
Solution.....	239
How It Works.....	240
10-2. Updating (and Iterating) a Map	240
Problem	240
Solution.....	241
How It Works.....	241
10-3. Inserting a Key into a Map Only If the Key Is Not Already Present.....	242
Problem	242
Solution.....	242
How It Works.....	243
10-4. Iterating Through a Changing Collection.....	244
Problem	244
Solution 1.....	244
Solution 2.....	244
How It Works.....	244

- 10-5. Coordinating Different Collections 245**
 - Problem 245
 - Solution 1 245
 - Solution 2..... 246
 - How It Works..... 247
- 10-6. Splitting Work into Separate Threads 249**
 - Problem 249
 - Solution..... 249
 - How It Works..... 249
- 10-7. Coordinating Threads..... 250**
 - Problem 250
 - Solution 1 250
 - Solution 2..... 251
 - Solution 3..... 252
 - How It Works..... 253
- 10-8. Creating Thread-Safe Objects..... 255**
 - Problem 255
 - Solution 1 255
 - Solution 2..... 256
 - How It Works..... 256
- 10-9. Implementing Thread-Safe Counters 257**
 - Problem 257
 - Solution..... 257
 - How It Works..... 257
- 10-10. Breaking Down Tasks into Discrete Units of Work 258**
 - Problem 258
 - Solution..... 258
 - How It Works..... 260

10-11. Updating a Common Value Across Multiple Threads.....	261
Problem	261
Solution.....	261
How It Works.....	263
10-12. Executing Multiple Tasks Asynchronously	264
Problem	264
Solution.....	264
How It Works.....	265
Summary	266
■ Chapter 11: Debugging and Unit Testing	267
11-1. Understanding Exceptions	267
Problem	267
Solution.....	267
How It Works.....	268
11-2. Locking Down Behavior of Your Classes	268
Problem	268
Solution.....	269
How It Works.....	270
11-3. Scripting Your Unit Tests	270
Problem	270
Solution.....	271
How It Works.....	273
11-4. Finding Bugs Early	273
Problem	273
Solution.....	274
How It Works.....	275
11-5. Monitoring Garbage Collection in Your Application	276
Problem	276
Solution 1.....	276
Solution 2.....	276
How It Works.....	278

11-6. Obtaining a Thread Dump	278
Problem	278
Solution.....	278
How It Works.....	284
■ Chapter 12: Unicode, Internationalization, and Currency Codes	285
12-1. Converting Unicode Characters to Digits	285
Problem	285
Solution.....	285
How It Works.....	287
12-2. Creating and Working with Locales	287
Problem	287
Solution.....	287
How It Works.....	290
12-3. Setting the Default Locale.....	291
Problem	291
Solution.....	291
How It Works.....	292
12-4. Matching and Filtering Locales.....	293
Problem	293
Solution.....	293
How It Works.....	295
12-5. Searching Unicode with Regular Expressions	295
Problem	295
Solution 1.....	295
Solution 2.....	297
How It Works.....	297
12-6. Overriding the Default Currency	298
Problem	298
Solution.....	298
How It Works.....	299

12-7. Converting Byte Arrays to and from Strings.....	300
Problem	300
Solution.....	300
How It Works.....	301
12-8. Converting Character Streams and Buffers	302
Problem	302
Solution 1.....	302
Solution 2.....	302
How It Works.....	303
12-9. Setting the Search Order of Locale-Sensitive Services	304
Problem	304
Solution.....	304
How It Works.....	304
Summary	304
■ Chapter 13: Working with Databases	305
13-1. Connecting to a Database.....	305
Problem	305
Solution 1.....	305
Solution 2.....	306
How It Works.....	306
13-2. Handling Connection and SQL Exceptions	308
Problem	308
Solution.....	308
How It Works.....	309
13-3. Querying a Database and Retrieving Results.....	310
Problem	310
Solution.....	310
How It Works.....	310

13-4. Performing CRUD Operations.....	312
Problem	312
Solution.....	312
How It Works.....	314
13-5. Simplifying Connection Management.....	315
Problem	315
Solution.....	315
How It Works.....	318
13-6. Guarding Against SQL Injection.....	318
Problem	318
Solution.....	319
How It Works.....	320
13-7. Performing Transactions	322
Problem	322
Solution.....	322
How It Works.....	324
13-8. Creating a Scrollable ResultSet	326
Problem	326
Solution.....	326
How It Works.....	327
13-9. Creating an Updatable ResultSet	328
Problem	328
Solution.....	328
How It Works.....	329
13-10. Caching Data for Use When Disconnected.....	330
Problem	330
Solution.....	330
How It Works.....	333

13-11. Joining RowSet Objects When Not Connected to the Data Source	335
Problem	335
Solution.....	335
How It Works.....	340
13-12. Filtering Data in a RowSet	341
Problem	341
Solution.....	341
How It Works.....	345
13-13. Querying and Storing Large Objects	346
Problem	346
Solution.....	347
How It Works.....	348
13-14. Invoking Stored Procedures.....	349
Problem	349
Solution.....	350
How It Works.....	350
13-15. Obtaining Dates for Database Use	351
Problem	351
Solution.....	351
How It Works.....	352
13-16. Closing Resources Automatically.....	352
Problem	352
Solution.....	352
How It Works.....	353
Summary	353
■ Chapter 14: JavaFX Fundamentals	355
14-1. Creating a Simple User Interface	356
Problem	356
Solution 1.....	356
Solution 2.....	357
How It Works.....	359

- 14-2. Drawing Text 360**
 - Problem 360
 - Solution..... 360
 - How It Works..... 361
- 14-3. Changing Text Fonts..... 362**
 - Problem 362
 - Solution 1..... 362
 - Solution 2..... 363
 - How It Works..... 365
- 14-4. Creating Shapes..... 367**
 - Problem 367
 - Solution..... 367
 - How It Works..... 370
- 14-5. Assigning Colors to Objects 372**
 - Problem 372
 - Solution..... 372
 - How It Works..... 375
- 14-6. Creating Menus..... 376**
 - Problem 376
 - Solution..... 376
 - How It Works..... 378
- 14-7. Adding Components to a Layout 380**
 - Problem 380
 - Solution..... 380
 - How It Works..... 381
- 14-8. Generating Borders 382**
 - Problem 382
 - Solution..... 382
 - How It Works..... 384

14-9. Binding Expressions	384
Problem	384
Solution.....	384
How It Works.....	388
14-10. Creating and Working with Observable Lists	389
Problem	389
Solution.....	389
How It Works.....	391
14-11. Generating a Background Process.....	392
Problem	392
Solution.....	392
How It Works.....	395
14-12. Associating Keyboard Sequences with Applications	397
Problem	397
Solution.....	397
How It Works.....	399
14-13. Creating and Working with Tables	399
Problem	399
Solution.....	399
How It Works.....	404
14-14. Organizing the UI with Split Views	405
Problem	405
Solution.....	405
How It Works.....	407
14-15. Adding Tabs to the UI	408
Problem	408
Solution.....	408
How It Works.....	410

14-16. Developing a Dialog Box	411
Problem	411
Solution.....	411
How It Works.....	415
14-17. Printing with JavaFX.....	416
Problem	416
Solution.....	416
How It Works.....	422
14-18. Embedding Swing Content in JavaFX	423
Problem	423
Solution.....	423
How It Works.....	427
Summary	427
■ Chapter 15: Graphics with JavaFX	429
15-1. Creating Images.....	430
Problem	430
Solution.....	430
How It Works.....	435
15-2. Generating an Animation	438
Problem	438
Solution.....	438
How It Works.....	442
15-3. Animating Shapes Along a Path	446
Problem	446
Solution.....	447
How It Works.....	450
15-4. Manipulating Layout via Grids	452
Problem	452
Solution.....	452
How It Works.....	460

15-5. Enhancing the Interface with CSS	463
Problem	463
Solution.....	463
How It Works.....	467
Summary.....	469
■ Chapter 16: Media with JavaFX.....	471
16-1. Playing Audio	471
Problem	471
Solution.....	472
How It Works.....	478
16-2. Playing Video	481
Problem	481
Solution.....	482
How It Works.....	486
16-3. Controlling Media Actions and Events	490
Problem	490
Solution.....	490
How It Works.....	491
16-4. Marking a Position in a Video.....	492
Problem	492
Solution.....	492
How It Works.....	495
16-5. Synchronizing Animation and Media	495
Problem	495
Solution.....	496
How It Works.....	497
Summary.....	497

- **Chapter 17: Java Web Applications with JavaServer Faces..... 499**
 - 17-1. Creating and Configure a Web Project 499
 - Problem 499
 - Solution..... 500
 - How It Works..... 503
 - 17-2. Developing a JSF Application 506
 - Problem 506
 - Solution..... 506
 - How It Works..... 509
 - 17-3. Developing a Model for Data..... 511
 - Problem 511
 - Solution..... 511
 - How It Works..... 513
 - 17-4. Writing View Controllers..... 517
 - Problem 517
 - Solution..... 517
 - How It Works..... 519
 - 17-5. Developing Asynchronous Views 519
 - Problem 519
 - Solution..... 519
 - How It Works..... 521
 - 17-6. Applying the Correct Scope..... 522
 - Problem 522
 - Solution..... 522
 - How It Works..... 522
 - 17-7. Generating and Applying a Template 523
 - Problem 523
 - Solution..... 523
 - How It Works..... 526
 - Summary 527

■ Chapter 18: Nashorn and Scripting	529
18-1. Loading and Executing JavaScript from Java.....	529
Problem	529
Solution.....	529
How It Works.....	530
18-2. Executing JavaScript via the Command Line.....	531
Problem	531
Solution 1.....	531
Solution 2.....	532
How It Works.....	533
18-3. Embedding Expressions in Strings	533
Problem	533
Solution.....	533
How It Works.....	534
18-4. Passing Java Parameters	534
Problem	534
Solution.....	534
How It Works.....	535
18-5. Passing Return Values from JavaScript to Java	535
Problem	535
Solution.....	535
How It Works.....	536
18-6. Using Java Classes and Libraries	537
Problem	537
Solution.....	537
How It Works.....	539
18-7. Accessing Java Arrays and Collections in Nashorn	540
Problem	540
Solution.....	540
How It Works.....	540

18-8. Implementing Java Interfaces	541
Problem	541
Solution.....	541
How It Works.....	542
18-9. Extending Java Classes	542
Problem	542
Solution.....	542
How It Works.....	543
18-10. Creating Executable Scripts in Unix.....	544
Problem	544
Solution.....	544
How It Works.....	544
18-11. Implementing JavaFX with Nashorn	545
Problem	545
Solution 1.....	545
Solution 2.....	547
How It Works.....	549
18-12. Utilizing ECMAScript6 Features	550
Problem	550
Solution.....	550
How It Works.....	551
Summary	551
■ Chapter 19: E-mail.....	553
19-1. Installing JavaMail.....	553
Problem	553
Solution.....	553
How It Works.....	553
19-2. Sending an E-Mail.....	554
Problem	554
Solution.....	554
How It Works.....	554

19-3. Attaching Files to an E-Mail Message.....	555
Problem	555
Solution.....	555
How It Works.....	556
19-4. Sending an HTML E-Mail	556
Problem	556
Solution.....	556
How It Works.....	557
19-5. Sending E-Mail to a Group of Recipients	557
Problem	557
Solution.....	557
How It Works.....	558
19-6. Checking E-Mail.....	558
Problem	558
Solution.....	558
How It Works.....	559
19-7. Monitoring an E-Mail Account.....	559
Problem	559
Solution.....	559
How It Works.....	561
Summary	561
■ Chapter 20: JSON and XML Processing	563
20-1. Writing an XML File	563
Problem	563
Solution.....	563
How It Works.....	565
20-2. Reading an XML File	566
Problem	566
Solution 1.....	566
Solution 2.....	567
How It Works.....	569

20-3. Transforming XML	569
Problem	569
Solution.....	569
How It Works.....	570
20-4. Validating XML	572
Problem	572
Solution.....	573
How It Works.....	573
20-5. Creating Java Bindings for an XML Schema	574
Problem	574
Solution.....	574
How It Works.....	575
20-6. Unmarshalling XML to a Java Object	575
Problem	575
Solution.....	575
How It Works.....	576
20-7. Building an XML Document with JAXB	576
Problem	576
Solution.....	577
How It Works.....	577
20-8. Parsing an XML Catalog.....	578
Problem	578
Solution.....	578
How It Works.....	578
20-9. Working with JSON	579
Problem	579
Solution.....	579
How It Works.....	580

20-10. Building a JSON Object.....	580
Problem	580
Solution.....	580
How It Works.....	580
20-11. Writing a JSON Object to File.....	581
Problem	581
Solution.....	581
How It Works.....	581
20-12. Parsing a JSON Object.....	582
Problem	582
Solution.....	582
How It Works.....	583
Summary	584
■ Chapter 21: Networking	585
21-1. Listening for Connections on the Server.....	585
Problem	585
Solution.....	585
How it Works.....	587
21-2. Defining a Network Connection to a Server.....	588
Problem	588
Solution.....	588
How it Works.....	590
21-3. Bypassing TCP for InfiniBand to Gain Performance Boosts	591
Problem	591
Solution.....	591
How it Works.....	592
21-4. Broadcasting to a Group of Recipients	593
Problem	593
Solution.....	593
How it Works.....	595

21-5. Generating and Reading from URLs	598
Problem	598
Solution.....	598
How it Works.....	599
21-6. Parsing a URL	600
Problem	600
Solution.....	600
How it Works.....	601
21-7. Making HTTP Requests and Working with HTTP Responses	602
Problem	602
Solution.....	602
How it Works.....	603
Summary	604
■ Chapter 22: Java Modularity	605
22-1. Constructing a Module.....	605
Problem	605
Solution.....	605
How It Works.....	606
22-2. Compiling and Executing a Module	606
Problem	606
Solution.....	606
How It Works.....	607
22-3. Creating a Module Dependency	607
Problem	607
Solution.....	607
How It Works.....	608
22-4. Packaging a Module	609
Problem	609
Solution.....	609
How It Works.....	609

22-5. Listing Dependencies or Determining JDK-Internal API Use.....	610
Problem	610
Solution.....	611
How It Works.....	612
22-6. Providing Loose Coupling Between Modules.....	612
Problem	612
Solution.....	612
How It Works.....	613
22-7. Linking Modules.....	613
Problem	613
Solution.....	613
How It Works.....	614
Summary	614
Index.....	615

About the Author



Josh Juneau has been developing software and enterprise applications since the early days of Java EE. Application and database development have been his focus since the start of his career. He became an Oracle database administrator and adopted the PL/SQL language for performing administrative tasks and developing applications for the Oracle database. In an effort to build more complex solutions, he began to incorporate Java into his PL/SQL applications and later developed stand-alone and web applications with Java. Josh wrote his early Java web applications utilizing JDBC and servlets or JSP to work with back-end databases. Later, he began to incorporate frameworks into his enterprise solutions, such as Java EE and JBoss Seam. Today, he primarily develops enterprise web solutions utilizing Java EE and other technologies. He also includes the use of alternative languages, such as Jython and Groovy, for some of his projects.

Over the years, Josh has dabbled in many different programming languages, including alternative languages for the JVM, in particular. In 2006, Josh began devoting time to the Jython Project as editor and publisher of the *Jython Monthly* newsletter. In late 2008, he began a podcast dedicated to the Jython programming language. Josh was the lead author for *The Definitive Guide to Jython*, *Oracle PL/SQL Recipes*, and *Java 7 Recipes*, and a solo author of *Java EE 7 Recipes* and *Introducing Java EE 7*, which were all published by Apress. He works as an application developer and system analyst at Fermi National Accelerator Laboratory, and he also writes technical articles for Oracle and OTN. He was a member of the JSR 372 and JSR 378 expert groups, and is an active member of the Java Community, helping to lead the Chicago Java User Group's Adopt-a-JSR effort.

When not coding or writing, Josh enjoys spending time with his wonderful wife and five children, especially swimming, fishing, playing ball, and watching movies. To hear more from Josh, follow his blog at <http://jj-blogger.blogspot.com>. You can also follow him on Twitter at @javajuneau.

About the Technical Reviewer



Vinay Kumar is an Oracle ACE and technology evangelist. He has extensive experience in designing and implementing large-scale projects in Oracle Enterprise Technologies. He has implemented multiple Enterprise Portal on web and intranet using Oracle WebCenter Portal/ADF and open source. He is a technology advisor, trainer, and architect. He loves exploring emerging solutions and applications mainly related to Oracle Middleware and open source. He loves spending his time in mentoring, writing technical blogs (www.techartifact.com), publishing white papers & maintaining dedicated education channel at youtube for Java, ADF/ WebCenter. He also authored a book on Enterprise Portal (WebCenter 12c) of Apress publication.

Find his views at [@vinaykuma201](https://twitter.com/vinaykuma201).

Acknowledgments

To my wife Angela: As the years pass, I will continue to be amazed by you. I want to thank for being a great wife, mother, and sidekick, always giving your all to me and the children. You've helped me make it through this book and my other projects, and your inspiration always keeps me moving forward. Thanks for always supporting the work I do. It was great to finally have you at JavaOne in 2016, and it was great to spend time away. I love you very much.

To my children, Kaitlyn, Jacob, Matthew, Zachary, and Lucas: I love you all so much and I cherish every moment we have together. You all continue to make me so proud through your schoolwork, Scouting, sports, and the myriad of other things that you achieve. I hate to see you growing up so quickly...sometimes I wish that I could pause time. I hope that you will understand why I've worked so hard at times on the nights and weekends when you read this book some day.

I want to thank my original coauthors of the first edition of this book: Carl Dea, Freddy Guime, John O'Conner, and Mark Beaty. You guys are Java gurus, and your expertise is engrained within the pages of this second edition just as much as the first. I had a great time working with you on the original and hope to do so again in the future.

To the folks at Apress: I thank you for providing me with the chance to share my knowledge with others. I especially thank Jonathan Gennick for the continued support of my work and for providing the continued guidance to produce useful content for our readers. I also thank Jill Balzano for doing a great job coordinating this project. The technical reviewer, Vinay Kumar, has done an excellent job of solidifying the book content. I really thank you for your hard work and expertise. Lastly, I'd like to thank everyone else at Apress who had a hand in this book.

To the Java community: Thanks for continuing to make the Java platform such an innovative and effective platform for application development. We all have the privilege of working with a mature and robust platform, and it would not be successful today if it weren't for everyone's continued contributions to the technology. I thank all of the Oracle and community Java experts. Once again, the roadmap for the future is continuing to look great. I am looking forward to using Java technology for many years to come.

Introduction

The Java programming language was introduced in 1995 by Sun Microsystems. Derived from languages such as C and C++, Java was designed to be more intuitive and easier to use than older languages, specifically due to its simplistic object model and automated facilities such as memory management. At the time, Java drew the interest of developers because of its object-oriented, concurrent architecture; its excellent security and scalability; and because applications developed in the Java language could run on any operating system that contained a Java Virtual Machine (JVM). Since its inception, Java has been described as a language that allows developers to “write once, run everywhere” as code is compiled into class files that contain bytecode, and the resulting class files can run on any compliant JVM. This concept made Java an immediate success for desktop development, which later branched off into different technological solutions over the years, including development of web-based applications and rich Internet applications (RIAs). Today, Java is deployed on a broad range of devices, including mobile phones, printers, medical devices, Blu-ray players, and so on.

The Java platform consists of a hierarchy of components, starting with the Java Development Kit (JDK), which is composed of the Java Runtime Environment (JRE), the Java programming language, and platform tools that are necessary to develop and run Java applications. The JRE contains the JVM, plus the Java application programming interfaces (APIs) and libraries that assist in the development of Java applications. The JVM is the base upon which compiled Java class files run and is responsible for interpreting compiled Java classes and executing the code. Every operating system that is capable of running Java code has its own version of the JVM. To that end, the JRE must be installed on any system that will be running local Java desktop or stand-alone Java applications. Oracle provides JRE implementations for most of the major operating systems. Each operating system can have its own flavor of the JRE. For instance, mobile devices can run a scaled-down version of the full JRE that is optimized to run Java Mobile Edition (ME) and Java SE embedded applications. The Java platform APIs and libraries are a collection of predefined classes that are used by all Java applications. Any application that runs on the JVM makes use of the Java platform APIs and libraries. This allows applications to use the functionality that has been predefined and loaded into the JVM and leaves developers with more time to worry about the details of their specific application. The classes that comprise the Java platform APIs and libraries allow Java applications to use one set of classes in order to communicate with the underlying operating system. As such, the Java platform takes care of interpreting the set of instructions provided by a Java application into operating system commands that are required for the machine on which the application is being executed. This creates a facade for Java developers to write code against so that they can develop applications that can be written once and run on every machine that contains a relevant JVM.

The JVM and the Java platform APIs and libraries play key roles in the life cycle of every Java application. Entire books have been written to explore the platform and JVM. This book focuses on the Java language itself, which is used to develop Java applications, although the JVM and Java platform APIs and libraries are referenced as needed. The Java language is a robust, secure, and modern object-oriented language that can be used to develop applications to run on the JVM. The Java programming language has been refined over several iterations and it becomes more powerful, secure, and modern with each new release. This book covers many features of the Java programming language from those that were introduced in Java 1.0 through those that made their way into the language in Java 9. In 2014, Oracle Corporation released Java 8, which was another milestone release for the Java ecosystem. Not only was Java already the most modern, statically

typed, object-oriented language available for development, but Java 8 added important new enhancements to the language, such as lambda expressions, streams processing, and default methods. JavaFX 8 was also released at the same time, advancing desktop Java applications more than ever. JavaFX 8 can be used for developing rich desktop and Internet applications using the Java language, or any other language that runs on the JVM. It provides a rich set of graphical and media user interfaces to develop extraordinary visual applications. This release is another nice update to the JavaFX platform, adding in features such as the Swing node and the Print API. In 2017, Java 9 is released, enhancing the platform with features such as modularity, an updated Process API, and jShell. This book covers the fundamentals of Java development, such as installing the JDK, writing classes, and running applications. It delves into essential topics such as the development of object-oriented constructs, exception handling, unit testing, and localization. The book also provides solutions for desktop application development using the JavaFX, and some web-based and database solutions. It covers JavaFX in depth and is an essential guide for developers beginning to use JavaFX 8+. This book can be used as a guide for solving problems that ordinary Java developers may encounter at some point. A broad range of topics is discussed, and the solutions to the problems that are covered in this book are concise and to the point. If you are a novice Java developer, we hope that this book will help you get started on your journey to working with one of the most advanced and widely used programming languages available today. For those of you who have used the Java language for some time, we hope that this book will provide you with updated material that is new to Java 9, JavaFX, and even some Java web development so that you can further refine your Java development skills. I ensure that advanced Java application developers will also learn a thing or two regarding the new features of the language and perhaps even stumble upon some techniques that were not used in the past. Whatever your skill level, this book is good to have close at hand as a reference for solutions to those problems that you encounter in your daily programming.

Who This Book Is For

This book is intended for all those who are interested in learning the Java programming language and/or already know the language but would like some information regarding the new features included in Java SE 9 and JavaFX. Those who have not yet programmed in the Java language can read this book, and it will allow them to start from scratch to get up and running quickly. Intermediate and advanced Java developers who are looking to update their arsenal with the latest features that Java SE 9 makes available to them can also read the book to quickly update and refresh their skill set. Java desktop programmers will find this book useful for its content on developing desktop applications using the JavaFX API. There is, of course, a myriad of other essential topics that will be useful to Java developers of any type.

How This Book Is Structured

This book is structured such that it does not have to be read from cover to cover. In fact, it is structured so that developers can choose which topics they wish to read about and jump right to them. Each recipe contains a problem to solve, one or more solutions to solve that problem, and a detailed explanation of how the solution works. Although some recipes may build upon concepts that have been discussed in other recipes, they contain the appropriate references so that the developer can find other related recipes that are beneficial to the solution. The book is designed to allow developers to get up and running quickly with a solution so that they can be home in time for dinner.