

Contents



Foreword **xxvii**

Preface **xxix**

Before You Begin **xliv**

I Introduction and Test-Driving a Java Application **I**

1.1	Introduction	2
1.2	Object Technology Concepts	3
1.2.1	Automobile as an Object	4
1.2.2	Methods and Classes	4
1.2.3	Instantiation	4
1.2.4	Reuse	4
1.2.5	Messages and Method Calls	4
1.2.6	Attributes and Instance Variables	5
1.2.7	Encapsulation and Information Hiding	5
1.2.8	Inheritance	5
1.2.9	Interfaces	5
1.2.10	Object-Oriented Analysis and Design (OOAD)	6
1.2.11	The UML (Unified Modeling Language)	6
1.3	Java	6
1.4	A Typical Java Development Environment	8
1.5	Test-Driving a Java Application	11
1.6	Software Technologies	15
1.7	Getting Your Questions Answered	18

2 Introduction to Java Applications; Input/Output and Operators **19**

2.1	Introduction	20
2.2	Your First Program in Java: Printing a Line of Text	20
2.2.1	Compiling the Application	23
2.2.2	Executing the Application	24
2.3	Modifying Your First Java Program	24

2.4	Displaying Text with <code>printf</code>	26
2.5	Another Application: Adding Integers	27
2.5.1	<code>import</code> Declarations	28
2.5.2	Declaring and Creating a Scanner to Obtain User Input from the Keyboard	28
2.5.3	Prompting the User for Input	28
2.5.4	Declaring a Variable to Store an Integer and Obtaining an Integer from the Keyboard	29
2.5.5	Obtaining a Second Integer	29
2.5.6	Using Variables in a Calculation	29
2.5.7	Displaying the Calculation Result	29
2.5.8	Java API Documentation	30
2.5.9	Declaring and Initializing Variables in Separate Statements	30
2.6	Arithmetic	30
2.7	Decision Making: Equality and Relational Operators	31
2.8	Wrap-Up	34

3 Introduction to Classes, Objects, Methods and Strings 35

3.1	Introduction	36
3.2	Instance Variables, <i>set</i> Methods and <i>get</i> Methods	37
3.2.1	Account Class with an Instance Variable, and <i>set</i> and <i>get</i> Methods	37
3.2.2	AccountTest Class That Creates and Uses an Object of Class Account	40
3.2.3	Compiling and Executing an App with Multiple Classes	43
3.2.4	Account UML Class Diagram	43
3.2.5	Additional Notes on Class AccountTest	44
3.2.6	Software Engineering with <code>private</code> Instance Variables and <code>public set</code> and <code>get</code> Methods	45
3.3	Account Class: Initializing Objects with Constructors	46
3.3.1	Declaring an Account Constructor for Custom Object Initialization	47
3.3.2	Class AccountTest: Initializing Account Objects When They're Created	47
3.4	Account Class with a Balance; Floating-Point Numbers	49
3.4.1	Account Class with a <code>balance</code> Instance Variable of Type <code>double</code>	49
3.4.2	AccountTest Class to Use Class Account	51
3.5	Primitive Types vs. Reference Types	54
3.6	Wrap-Up	55

4 Control Statements: Part I; Assignment, ++ and -- Operators 56

4.1	Introduction	57
4.2	Control Structures	57
4.2.1	Sequence Structure in Java	57

4.2.2	Selection Statements in Java	58
4.2.3	Iteration Statements in Java	59
4.2.4	Summary of Control Statements in Java	59
4.3	if Single-Selection Statement	59
4.4	if...else Double-Selection Statement	60
4.4.1	Nested if...else Statements	61
4.4.2	Dangling-else Problem	62
4.4.3	Blocks	62
4.4.4	Conditional Operator (?:)	63
4.5	while Iteration Statement	63
4.6	Counter-Controlled Iteration	65
4.7	Sentinel-Controlled Iteration	68
4.8	Nesting Different Control Statements	72
4.9	Compound Assignment Operators	74
4.10	Increment and Decrement Operators	75
4.11	Primitive Types	78
4.12	Wrap-Up	78

5 Control Statements: Part 2; Logical Operators 79

5.1	Introduction	80
5.2	Essentials of Counter-Controlled Iteration	80
5.3	for Iteration Statement	81
5.4	Examples Using the for Statement	85
5.4.1	Application: Summing the Even Integers from 2 to 20	86
5.4.2	Application: Compound-Interest Calculations	87
5.5	do...while Iteration Statement	90
5.6	switch Multiple-Selection Statement	90
5.7	Class AutoPolicy: Strings in switch Statements	97
5.8	break and continue Statements	100
5.8.1	break Statement	100
5.8.2	continue Statement	101
5.9	Logical Operators	102
5.9.1	Conditional AND (&&) Operator	102
5.9.2	Conditional OR () Operator	103
5.9.3	Short-Circuit Evaluation of Complex Conditions	103
5.9.4	Boolean Logical AND (&) and Boolean Logical Inclusive OR () Operators	104
5.9.5	Boolean Logical Exclusive OR (^) Operator	104
5.9.6	Logical Negation (!) Operator	105
5.9.7	Logical Operators Example	105
5.10	Wrap-Up	108

6 Methods: A Deeper Look 109

6.1	Introduction	110
-----	--------------	-----

6.2	Program Units in Java	110
6.3	static Methods, static Fields and Class Math	111
6.4	Methods with Multiple Parameters	113
6.5	Notes on Declaring and Using Methods	116
6.6	Argument Promotion and Casting	117
6.7	Java API Packages	119
6.8	Case Study: Secure Random-Number Generation	120
6.9	Case Study: A Game of Chance; Introducing enum Types	125
6.10	Scope of Declarations	129
6.11	Method Overloading	132
	6.11.1 Declaring Overloaded Methods	132
	6.11.2 Distinguishing Between Overloaded Methods	133
	6.11.3 Return Types of Overloaded Methods	134
6.12	Wrap-Up	134

7 Arrays and ArrayLists **135**

7.1	Introduction	136
7.2	Arrays	137
7.3	Declaring and Creating Arrays	138
7.4	Examples Using Arrays	139
	7.4.1 Creating and Initializing an Array	140
	7.4.2 Using an Array Initializer	141
	7.4.3 Calculating the Values to Store in an Array	141
	7.4.4 Summing the Elements of an Array	143
	7.4.5 Using Bar Charts to Display Array Data Graphically	143
	7.4.6 Using the Elements of an Array as Counters	145
	7.4.7 Using Arrays to Analyze Survey Results	146
7.5	Exception Handling: Processing the Incorrect Response	148
	7.5.1 The try Statement	148
	7.5.2 Executing the catch Block	148
	7.5.3 toString Method of the Exception Parameter	149
7.6	Case Study: Card Shuffling and Dealing Simulation	149
7.7	Enhanced for Statement	153
7.8	Passing Arrays to Methods	155
7.9	Pass-By-Value vs. Pass-By-Reference	157
7.10	Case Study: Class GradeBook Using an Array to Store Grades	158
7.11	Multidimensional Arrays	163
	7.11.1 Arrays of One-Dimensional Arrays	164
	7.11.2 Two-Dimensional Arrays with Rows of Different Lengths	164
	7.11.3 Creating Two-Dimensional Arrays with Array-Creation Expressions	165
	7.11.4 Two-Dimensional Array Example: Displaying Element Values	165
	7.11.5 Common Multidimensional-Array Manipulations Performed with for Statements	166
7.12	Case Study: Class GradeBook Using a Two-Dimensional Array	167

7.13	Variable-Length Argument Lists	173
7.14	Using Command-Line Arguments	174
7.15	Class Arrays	176
7.16	Introduction to Collections and Class ArrayList	179
7.17	Wrap-Up	182

8 **Classes and Objects: A Deeper Look** **184**

8.1	Introduction	185
8.2	Time Class Case Study	185
8.3	Controlling Access to Members	190
8.4	Referring to the Current Object's Members with the <code>this</code> Reference	191
8.5	Time Class Case Study: Overloaded Constructors	193
8.6	Default and No-Argument Constructors	198
8.7	Notes on <i>Set</i> and <i>Get</i> Methods	199
8.8	Composition	200
8.9	enum Types	203
8.10	Garbage Collection	206
8.11	<code>static</code> Class Members	206
8.12	<code>static</code> Import	210
8.13	<code>final</code> Instance Variables	211
8.14	Package Access	212
8.15	Using <code>BigDecimal</code> for Precise Monetary Calculations	213
8.16	JavaMoney API	216
8.17	Time Class Case Study: Creating Packages	216
8.18	Wrap-Up	220

9 **Object-Oriented Programming: Inheritance** **221**

9.1	Introduction	222
9.2	Superclasses and Subclasses	223
9.3	<code>protected</code> Members	225
9.4	Relationship Between Superclasses and Subclasses	226
9.4.1	Creating and Using a <code>CommissionEmployee</code> Class	226
9.4.2	Creating and Using a <code>BasePlusCommissionEmployee</code> Class	231
9.4.3	Creating a <code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy	236
9.4.4	<code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy Using <code>protected</code> Instance Variables	239
9.4.5	<code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy Using <code>private</code> Instance Variables	242
9.5	Constructors in Subclasses	246
9.6	Class Object	247
9.7	Designing with Composition vs. Inheritance	248
9.8	Wrap-Up	249

10	Object-Oriented Programming: Polymorphism and Interfaces	251
10.1	Introduction	252
10.2	Polymorphism Examples	254
10.3	Demonstrating Polymorphic Behavior	255
10.4	Abstract Classes and Methods	257
10.5	Case Study: Payroll System Using Polymorphism	260
10.5.1	Abstract Superclass <code>Employee</code>	261
10.5.2	Concrete Subclass <code>SalariedEmployee</code>	263
10.5.3	Concrete Subclass <code>HourlyEmployee</code>	265
10.5.4	Concrete Subclass <code>CommissionEmployee</code>	266
10.5.5	Indirect Concrete Subclass <code>BasePlusCommissionEmployee</code>	268
10.5.6	Polymorphic Processing, Operator <code>instanceof</code> and Downcasting	269
10.6	Allowed Assignments Between Superclass and Subclass Variables	274
10.7	<code>final</code> Methods and Classes	274
10.8	A Deeper Explanation of Issues with Calling Methods from Constructors	275
10.9	Creating and Using Interfaces	276
10.9.1	Developing a Payable Hierarchy	278
10.9.2	Interface <code>Payable</code>	279
10.9.3	Class <code>Invoice</code>	279
10.9.4	Modifying Class <code>Employee</code> to Implement Interface <code>Payable</code>	281
10.9.5	Using Interface <code>Payable</code> to Process Invoices and Employees Polymorphically	283
10.9.6	Some Common Interfaces of the Java API	284
10.10	Java SE 8 Interface Enhancements	285
10.10.1	<code>default</code> Interface Methods	285
10.10.2	<code>static</code> Interface Methods	286
10.10.3	Functional Interfaces	286
10.11	Java SE 9 <code>private</code> Interface Methods	287
10.12	<code>private</code> Constructors	287
10.13	Program to an Interface, Not an Implementation	288
10.13.1	Implementation Inheritance Is Best for Small Numbers of Tightly Coupled Classes	288
10.13.2	Interface Inheritance Is Best for Flexibility	288
10.13.3	Rethinking the Employee Hierarchy	289
10.14	Wrap-Up	290
11	Exception Handling: A Deeper Look	291
11.1	Introduction	292
11.2	Example: Divide by Zero without Exception Handling	293
11.3	Example: Handling <code>ArithmeticExceptions</code> and <code>InputMismatchExceptions</code>	295
11.4	When to Use Exception Handling	300
11.5	Java Exception Hierarchy	301
11.6	<code>finally</code> Block	304

11.7	Stack Unwinding and Obtaining Information from an Exception	309
11.8	Chained Exceptions	311
11.9	Declaring New Exception Types	313
11.10	Preconditions and Postconditions	314
11.11	Assertions	315
11.12	try-with-Resources: Automatic Resource Deallocation	317
11.13	Wrap-Up	318

12 JavaFX Graphical User Interfaces: Part 1 319

12.1	Introduction	320
12.2	JavaFX Scene Builder	321
12.3	JavaFX App Window Structure	322
12.4	Welcome App —Displaying Text and an Image	323
12.4.1	Opening Scene Builder and Creating the File <code>Welcome.fxml</code>	323
12.4.2	Adding an Image to the Folder Containing <code>Welcome.fxml</code>	324
12.4.3	Creating a VBox Layout Container	324
12.4.4	Configuring the VBox Layout Container	325
12.4.5	Adding and Configuring a Label	325
12.4.6	Adding and Configuring an ImageView	326
12.4.7	Previewing the Welcome GUI	328
12.5	Tip Calculator App —Introduction to Event Handling	328
12.5.1	Test-Driving the Tip Calculator App	329
12.5.2	Technologies Overview	330
12.5.3	Building the App's GUI	332
12.5.4	TipCalculator Class	339
12.5.5	TipCalculatorController Class	341
12.6	Features Covered in the Other JavaFX Chapters	346
12.7	Wrap-Up	346

13 JavaFX GUI: Part 2 347

13.1	Introduction	348
13.2	Laying Out Nodes in a Scene Graph	348
13.3	Painter App : RadioButtons, Mouse Events and Shapes	350
13.3.1	Technologies Overview	350
13.3.2	Creating the Painter.fxml File	352
13.3.3	Building the GUI	352
13.3.4	Painter Subclass of Application	355
13.3.5	PainterController Class	356
13.4	Color Chooser App : Property Bindings and Property Listeners	360
13.4.1	Technologies Overview	360
13.4.2	Building the GUI	361
13.4.3	ColorChooser Subclass of Application	363
13.4.4	ColorChooserController Class	364
13.5	Cover Viewer App : Data-Driven GUIs with JavaFX Collections	366
13.5.1	Technologies Overview	367

13.5.2	Adding Images to the App's Folder	367
13.5.3	Building the GUI	367
13.5.4	CoverViewer Subclass of Application	369
13.5.5	CoverViewerController Class	369
13.6	Cover Viewer App: Customizing ListView Cells	371
13.6.1	Technologies Overview	372
13.6.2	Copying the CoverViewer App	372
13.6.3	ImageTextCell Custom Cell Factory Class	373
13.6.4	CoverViewerController Class	374
13.7	Additional JavaFX Capabilities	375
13.8	JavaFX 9: Java SE 9 JavaFX Updates	377
13.9	Wrap-Up	379

14 **Strings, Characters and Regular Expressions** **380**

14.1	Introduction	381
14.2	Fundamentals of Characters and Strings	381
14.3	Class String	382
14.3.1	String Constructors	382
14.3.2	String Methods length, charAt and getChars	383
14.3.3	Comparing Strings	385
14.3.4	Locating Characters and Substrings in Strings	389
14.3.5	Extracting Substrings from Strings	391
14.3.6	Concatenating Strings	392
14.3.7	Miscellaneous String Methods	392
14.3.8	String Method valueOf	394
14.4	Class StringBuilder	395
14.4.1	StringBuilder Constructors	396
14.4.2	StringBuilder Methods length, capacity, setLength and ensureCapacity	396
14.4.3	StringBuilder Methods charAt, setCharAt, getChars and reverse	398
14.4.4	StringBuilder append Methods	399
14.4.5	StringBuilder Insertion and Deletion Methods	401
14.5	Class Character	402
14.6	Tokenizing Strings	407
14.7	Regular Expressions, Class Pattern and Class Matcher	408
14.7.1	Replacing Substrings and Splitting Strings	413
14.7.2	Classes Pattern and Matcher	415
14.8	Wrap-Up	417

15 **Files, Input/Output Streams, NIO and XML Serialization** **418**

15.1	Introduction	419
15.2	Files and Streams	419

15.3	Using NIO Classes and Interfaces to Get File and Directory Information	421
15.4	Sequential Text Files	425
15.4.1	Creating a Sequential Text File	425
15.4.2	Reading Data from a Sequential Text File	428
15.4.3	Case Study: A Credit-Inquiry Program	429
15.4.4	Updating Sequential Files	434
15.5	XML Serialization	434
15.5.1	Creating a Sequential File Using XML Serialization	434
15.5.2	Reading and Deserializing Data from a Sequential File	440
15.6	FileChooser and DirectoryChooser Dialogs	441
15.7	(Optional) Additional java.io Classes	447
15.7.1	Interfaces and Classes for Byte-Based Input and Output	447
15.7.2	Interfaces and Classes for Character-Based Input and Output	449
15.8	Wrap-Up	450

16 Generic Collections **451**

16.1	Introduction	452
16.2	Collections Overview	452
16.3	Type-Wrapper Classes	454
16.4	Autoboxing and Auto-Unboxing	454
16.5	Interface Collection and Class Collections	454
16.6	Lists	455
16.6.1	ArrayList and Iterator	456
16.6.2	LinkedList	458
16.7	Collections Methods	463
16.7.1	Method sort	463
16.7.2	Method shuffle	467
16.7.3	Methods reverse, fill, copy, max and min	469
16.7.4	Method binarySearch	471
16.7.5	Methods addAll, frequency and disjoint	472
16.8	Class PriorityQueue and Interface Queue	474
16.9	Sets	475
16.10	Maps	478
16.11	Synchronized Collections	482
16.12	Unmodifiable Collections	482
16.13	Abstract Implementations	483
16.14	Java SE 9: Convenience Factory Methods for Immutable Collections	483
16.15	Wrap-Up	487

17 Lambdas and Streams **488**

17.1	Introduction	489
17.2	Streams and Reduction	491
17.2.1	Summing the Integers from 1 through 10 with a for Loop	491
17.2.2	External Iteration with for Is Error Prone	492
17.2.3	Summing with a Stream and Reduction	492

17.2.4	Internal Iteration	493
17.3	Mapping and Lambdas	494
17.3.1	Lambda Expressions	495
17.3.2	Lambda Syntax	496
17.3.3	Intermediate and Terminal Operations	497
17.4	Filtering	498
17.5	How Elements Move Through Stream Pipelines	500
17.6	Method References	501
17.6.1	Creating an IntStream of Random Values	502
17.6.2	Performing a Task on Each Stream Element with forEach and a Method Reference	502
17.6.3	Mapping Integers to String Objects with mapToObj	503
17.6.4	Concatenating Strings with collect	503
17.7	IntStream Operations	504
17.7.1	Creating an IntStream and Displaying Its Values	505
17.7.2	Terminal Operations count, min, max, sum and average	505
17.7.3	Terminal Operation reduce	506
17.7.4	Sorting IntStream Values	508
17.8	Functional Interfaces	509
17.9	Lambdas: A Deeper Look	510
17.10	Stream<Integer> Manipulations	511
17.10.1	Creating a Stream<Integer>	512
17.10.2	Sorting a Stream and Collecting the Results	513
17.10.3	Filtering a Stream and Storing the Results for Later Use	513
17.10.4	Filtering and Sorting a Stream and Collecting the Results	514
17.10.5	Sorting Previously Collected Results	514
17.11	Stream<String> Manipulations	514
17.11.1	Mapping Strings to Uppercase	515
17.11.2	Filtering Strings Then Sorting Them in Case-Insensitive Ascending Order	516
17.11.3	Filtering Strings Then Sorting Them in Case-Insensitive Descending Order	516
17.12	Stream<Employee> Manipulations	517
17.12.1	Creating and Displaying a List<Employee>	518
17.12.2	Filtering Employees with Salaries in a Specified Range	519
17.12.3	Sorting Employees By Multiple Fields	522
17.12.4	Mapping Employees to Unique-Last-Name Strings	524
17.12.5	Grouping Employees By Department	525
17.12.6	Counting the Number of Employees in Each Department	526
17.12.7	Summing and Averaging Employee Salaries	527
17.13	Creating a Stream<String> from a File	528
17.14	Streams of Random Values	531
17.15	Infinite Streams	533
17.16	Lambda Event Handlers	535
17.17	Additional Notes on Java SE 8 Interfaces	535
17.18	Wrap-Up	536

18	Recursion	537
18.1	Introduction	538
18.2	Recursion Concepts	538
18.3	Example Using Recursion: Factorials	539
18.4	Reimplementing Class <code>FactorialCalculator</code> Using <code>BigInteger</code>	541
18.5	Example Using Recursion: Fibonacci Series	543
18.6	Recursion and the Method-Call Stack	546
18.7	Recursion vs. Iteration	547
18.8	Towers of Hanoi	549
18.9	Fractals	551
18.9.1	Koch Curve Fractal	551
18.9.2	(Optional) Case Study: Lo Feather Fractal	552
18.9.3	(Optional) <code>FractalApp</code> GUI	555
18.9.4	(Optional) <code>FractalController</code> Class	557
18.10	Recursive Backtracking	561
18.11	Wrap-Up	562
19	Generic Classes and Methods: A Deeper Look	563
19.1	Introduction	564
19.2	Motivation for Generic Methods	564
19.3	Generic Methods: Implementation and Compile-Time Translation	566
19.4	Additional Compile-Time Translation Issues: Methods That Use a Type Parameter as the Return Type	569
19.5	Overloading Generic Methods	572
19.6	Generic Classes	573
19.7	Wildcards in Methods That Accept Type Parameters	580
19.8	Wrap-Up	584
20	JavaFX Graphics, Animation and Video	585
20.1	Introduction	586
20.2	Controlling Fonts with Cascading Style Sheets (CSS)	587
20.2.1	CSS That Styles the GUI	587
20.2.2	FXML That Defines the GUI—Introduction to XML Markup	590
20.2.3	Referencing the CSS File from FXML	593
20.2.4	Specifying the <code>VBox</code> 's Style Class	593
20.2.5	Programmatically Loading CSS	593
20.3	Displaying Two-Dimensional Shapes	594
20.3.1	Defining Two-Dimensional Shapes with FXML	594
20.3.2	CSS That Styles the Two-Dimensional Shapes	596
20.4	<code>PolyLines</code> , <code>Polygons</code> and <code>Paths</code>	599
20.4.1	GUI and CSS	599
20.4.2	<code>PolyShapesController</code> Class	601
20.5	Transforms	604
20.6	Playing Video with <code>Media</code> , <code>MediaPlayer</code> and <code>MediaViewer</code>	606

20.6.1	VideoPlayer GUI	607
20.6.2	VideoPlayerController Class	609
20.7	Transition Animations	612
20.7.1	TransitionAnimations.fxml	612
20.7.2	TransitionAnimationsController Class	615
20.8	Timeline Animations	618
20.9	Frame-by-Frame Animation with AnimationTimer	621
20.10	Drawing on a Canvas	624
20.11	Three-Dimensional Shapes	628
20.12	Wrap-Up	632

21 Concurrency and Multi-Core Performance **634**

21.1	Introduction	635
21.2	Thread States and Life Cycle	637
21.2.1	<i>New</i> and <i>Runnable</i> States	638
21.2.2	<i>Waiting</i> State	638
21.2.3	<i>Timed Waiting</i> State	638
21.2.4	<i>Blocked</i> State	638
21.2.5	<i>Terminated</i> State	638
21.2.6	Operating-System View of the <i>Runnable</i> State	639
21.2.7	Thread Priorities and Thread Scheduling	639
21.2.8	Indefinite Postponement and Deadlock	640
21.3	Creating and Executing Threads with the Executor Framework	640
21.4	Thread Synchronization	644
21.4.1	Immutable Data	645
21.4.2	Monitors	645
21.4.3	Unsynchronized Mutable Data Sharing	646
21.4.4	Synchronized Mutable Data Sharing—Making Operations Atomic	650
21.5	Producer/Consumer Relationship without Synchronization	653
21.6	Producer/Consumer Relationship: ArrayBlockingQueue	661
21.7	(Advanced) Producer/Consumer Relationship with synchronized, wait, notify and notifyAll	664
21.8	(Advanced) Producer/Consumer Relationship: Bounded Buffers	670
21.9	(Advanced) Producer/Consumer Relationship: The Lock and Condition Interfaces	678
21.10	Concurrent Collections	685
21.11	Multithreading in JavaFX	687
21.11.1	Performing Computations in a Worker Thread: Fibonacci Numbers	688
21.11.2	Processing Intermediate Results: Sieve of Eratosthenes	693
21.12	sort/parallelSort Timings with the Java SE 8 Date/Time API	699
21.13	Java SE 8: Sequential vs. Parallel Streams	702
21.14	(Advanced) Interfaces Callable and Future	704
21.15	(Advanced) Fork/Join Framework	709
21.16	Wrap-Up	709

22 Accessing Databases with JDBC 711

22.1	Introduction	712
22.2	Relational Databases	713
22.3	A books Database	714
22.4	SQL	718
22.4.1	Basic SELECT Query	719
22.4.2	WHERE Clause	719
22.4.3	ORDER BY Clause	721
22.4.4	Merging Data from Multiple Tables: INNER JOIN	723
22.4.5	INSERT Statement	724
22.4.6	UPDATE Statement	725
22.4.7	DELETE Statement	726
22.5	Setting Up a Java DB Database	727
22.5.1	Creating the Chapter's Databases on Windows	728
22.5.2	Creating the Chapter's Databases on macOS	729
22.5.3	Creating the Chapter's Databases on Linux	729
22.6	Connecting to and Querying a Database	729
22.6.1	Automatic Driver Discovery	731
22.6.2	Connecting to the Database	731
22.6.3	Creating a Statement for Executing Queries	732
22.6.4	Executing a Query	732
22.6.5	Processing a Query's ResultSet	732
22.7	Querying the books Database	734
22.7.1	ResultSetTableModel Class	734
22.7.2	DisplayQueryResults App's GUI	741
22.7.3	DisplayQueryResultsController Class	741
22.8	RowSet Interface	746
22.9	PreparedStatement	749
22.9.1	AddressBook App That Uses PreparedStatement	750
22.9.2	Class Person	750
22.9.3	Class PersonQueries	752
22.9.4	AddressBook GUI	755
22.9.5	Class AddressBookController	756
22.10	Stored Procedures	761
22.11	Transaction Processing	761
22.12	Wrap-Up	762

23 Introduction to JShell: Java 9's REPL for Interactive Java 763

23.1	Introduction	764
23.2	Installing JDK 9	766
23.3	Introduction to JShell	766
23.3.1	Starting a JShell Session	767

23.3.2	Executing Statements	767
23.3.3	Declaring Variables Explicitly	768
23.3.4	Listing and Executing Prior Snippets	770
23.3.5	Evaluating Expressions and Declaring Variables Implicitly	772
23.3.6	Using Implicitly Declared Variables	772
23.3.7	Viewing a Variable’s Value	773
23.3.8	Resetting a JShell Session	773
23.3.9	Writing Multiline Statements	773
23.3.10	Editing Code Snippets	774
23.3.11	Exiting JShell	777
23.4	Command-Line Input in JShell	777
23.5	Declaring and Using Classes	778
23.5.1	Creating a Class in JShell	779
23.5.2	Explicitly Declaring Reference-Type Variables	779
23.5.3	Creating Objects	780
23.5.4	Manipulating Objects	780
23.5.5	Creating a Meaningful Variable Name for an Expression	781
23.5.6	Saving and Opening Code-Snippet Files	782
23.6	Discovery with JShell Auto-Completion	782
23.6.1	Auto-Completing Identifiers	783
23.6.2	Auto-Completing JShell Commands	784
23.7	Exploring a Class’s Members and Viewing Documentation	784
23.7.1	Listing Class <code>Math</code> ’s <code>static</code> Members	785
23.7.2	Viewing a Method’s Parameters	785
23.7.3	Viewing a Method’s Documentation	786
23.7.4	Viewing a <code>public</code> Field’s Documentation	786
23.7.5	Viewing a Class’s Documentation	787
23.7.6	Viewing Method Overloads	787
23.7.7	Exploring Members of a Specific Object	788
23.8	Declaring Methods	790
23.8.1	Forward Referencing an Undeclared Method—Declaring Method <code>displayCubes</code>	790
23.8.2	Declaring a Previously Undeclared Method	790
23.8.3	Testing <code>cube</code> and Replacing Its Declaration	791
23.8.4	Testing Updated Method <code>cube</code> and Method <code>displayCubes</code>	791
23.9	Exceptions	792
23.10	Importing Classes and Adding Packages to the <code>CLASSPATH</code>	793
23.11	Using an External Editor	795
23.12	Summary of JShell Commands	797
23.12.1	Getting Help in JShell	798
23.12.2	<code>/edit</code> Command: Additional Features	799
23.12.3	<code>/reload</code> Command	799
23.12.4	<code>/drop</code> Command	800
23.12.5	Feedback Modes	800
23.12.6	Other JShell Features Configurable with <code>/set</code>	802
23.13	Keyboard Shortcuts for Snippet Editing	803

23.14	How JShell Reinterprets Java for Interactive Use	803
23.15	IDE JShell Support	804
23.16	Wrap-Up	804

24 Java Persistence API (JPA) 820

24.1	Introduction	821
24.2	JPA Technology Overview	822
24.2.1	Generated Entity Classes	822
24.2.2	Relationships Between Tables in the Entity Classes	822
24.2.3	The <code>javax.persistence</code> Package	823
24.3	Querying a Database with JPA	823
24.3.1	Creating the Java DB Database	824
24.3.2	Populate the books Database with Sample Data	825
24.3.3	Creating the Java Project	825
24.3.4	Adding the JPA and Java DB Libraries	826
24.3.5	Creating the Persistence Unit for the books Database	826
24.3.6	Querying the Authors Table	827
24.3.7	JPA Features of Autogenerated Class Authors	829
24.4	Named Queries; Accessing Data from Multiple Tables	830
24.4.1	Using a Named Query to Get the List of Authors, then Display the Authors with Their Titles	830
24.4.2	Using a Named Query to Get the List of Titles, then Display Each with Its Authors	833
24.5	Address Book: Using JPA and Transactions to Modify a Database	835
24.5.1	Transaction Processing	835
24.5.2	Creating the AddressBook Database, Project and Persistence Unit	836
24.5.3	Addresses Entity Class	837
24.5.4	AddressBookController Class	837
24.5.5	Other JPA Operations	843
24.6	Web Resources	843
24.7	Wrap-Up	844

25 ATM Case Study, Part I: Object-Oriented Design with the UML 845

25.1	Case Study Introduction	846
25.2	Examining the Requirements Document	846
25.3	Identifying the Classes in a Requirements Document	854
25.4	Identifying Class Attributes	860
25.5	Identifying Objects' States and Activities	865
25.6	Identifying Class Operations	868
25.7	Indicating Collaboration Among Objects	875
25.8	Wrap-Up	882

26	ATM Case Study Part 2: Implementing an Object-Oriented Design	886
26.1	Introduction	887
26.2	Starting to Program the Classes of the ATM System	887
26.3	Incorporating Inheritance and Polymorphism into the ATM System	892
26.4	ATM Case Study Implementation	898
26.4.1	Class ATM	898
26.4.2	Class Screen	904
26.4.3	Class Keypad	905
26.4.4	Class CashDispenser	905
26.4.5	Class DepositSlot	907
26.4.6	Class Account	907
26.4.7	Class BankDatabase	909
26.4.8	Class Transaction	911
26.4.9	Class BalanceInquiry	913
26.4.10	Class Withdrawal	914
26.4.11	Class Deposit	918
26.4.12	Class ATMCaseStudy	920
26.5	Wrap-Up	921
27	Java Platform Module System	923
27.1	Introduction	924
27.2	Module Declarations	929
27.2.1	requires	930
27.2.2	requires transitive—Implied Readability	930
27.2.3	exports and exports...to	930
27.2.4	uses	931
27.2.5	provides...with	931
27.2.6	open, opens and opens...to	931
27.2.7	Restricted Keywords	932
27.3	Modularized Welcome App	932
27.3.1	Welcome App's Structure	933
27.3.2	Class Welcome	936
27.3.3	module-info.java	936
27.3.4	Module-Dependency Graph	937
27.3.5	Compiling a Module	938
27.3.6	Running an App from a Module's Exploded Folders	940
27.3.7	Packaging a Module into a Modular JAR File	940
27.3.8	Running the Welcome App from a Modular JAR File	941
27.3.9	Aside: Classpath vs. Module Path	941
27.4	Creating and Using a Custom Module	942
27.4.1	Exporting a Package for Use in Other Modules	942
27.4.2	Using a Class from a Package in Another Module	943
27.4.3	Compiling and Running the Example	945

27.4.4	Packaging the App into Modular JAR Files	946
27.4.5	Strong Encapsulation and Accessibility	947
27.5	Module-Dependency Graphs: A Deeper Look	948
27.5.1	java.sql	948
27.5.2	java.se	948
27.5.3	Browsing the JDK Module Graph	950
27.5.4	Error: Module Graph with a Cycle	950
27.6	Migrating Code to Java 9	951
27.6.1	Unnamed Module	952
27.6.2	Automatic Modules	952
27.6.3	jdeps: Java Dependency Analysis	953
27.7	Resources in Modules; Using an Automatic Module	955
27.7.1	Automatic Modules	956
27.7.2	Requiring Multiple Modules	957
27.7.3	Opening a Module for Reflection	957
27.7.4	Module-Dependency Graph	958
27.7.5	Compiling the Module	958
27.7.6	Running a Modularized App	959
27.8	Creating Custom Runtimes with <code>jlLink</code>	959
27.8.1	Listing the JRE's Modules	960
27.8.2	Custom Runtime Containing Only <code>java.base</code>	961
27.8.3	Creating a Custom Runtime for the <code>Welcome</code> App	962
27.8.4	Executing the <code>Welcome</code> App Using a Custom Runtime	962
27.8.5	Using the Module Resolver on a Custom Runtime	963
27.9	Services and <code>ServiceLoader</code>	963
27.9.1	Service-Provider Interface	965
27.9.2	Loading and Consuming Service Providers	966
27.9.3	<code>uses</code> Module Directive and Service Consumers	969
27.9.4	Running the App with No Service Providers	969
27.9.5	Implementing a Service Provider	970
27.9.6	<code>provides...with</code> Module Directive and Declaring a Service Provider	971
27.9.7	Running the App with One Service Provider	971
27.9.8	Implementing a Second Service Provider	972
27.9.9	Running the App with Two Service Providers	973
27.10	Wrap-Up	973

28 Additional Java 9 Topics

975

28.1	Introduction	976
28.2	Recap: Java 9 Features Covered in Earlier Chapters	977
28.3	New Version String Format	977
28.4	Regular Expressions: New <code>Matcher</code> Class Methods	978
28.4.1	Methods <code>appendReplacement</code> and <code>appendTail</code>	979
28.4.2	Methods <code>replaceFirst</code> and <code>replaceAll</code>	980
28.4.3	Method <code>results</code>	980

28.5	New Stream Interface Methods	980
28.5.1	Stream Methods <code>takeWhile</code> and <code>dropWhile</code>	982
28.5.2	Stream Method <code>iterate</code>	982
28.5.3	Stream Method of <code>Nullable</code>	983
28.6	Modules in JShell	983
28.7	JavaFX 9 Skin APIs	984
28.8	Other GUI and Graphics Enhancements	985
28.8.1	Multi-Resolution Images	985
28.8.2	TIFF Image I/O	985
28.8.3	Platform-Specific Desktop Features	986
28.9	Security Related Java 9 Topics	986
28.9.1	Filter Incoming Serialization Data	986
28.9.2	Create PKCS12 Keystores by Default	986
28.9.3	Datagram Transport Layer Security (DTLS)	987
28.9.4	OCSP Stapling for TLS	987
28.9.5	TLS Application-Layer Protocol Negotiation Extension	987
28.10	Other Java 9 Topics	987
28.10.1	Indify String Concatenation	987
28.10.2	Platform Logging API and Service	987
28.10.3	Process API Updates	988
28.10.4	Spin-Wait Hints	988
28.10.5	UTF-8 Property Resource Bundles	988
28.10.6	Use CLDR Locale Data by Default	988
28.10.7	Elide Deprecation Warnings on Import Statements	989
28.10.8	Multi-Release JAR Files	989
28.10.9	Unicode 8	989
28.10.10	Concurrency Enhancements	989
28.11	Items Removed from the JDK and Java 9	990
28.12	Items Proposed for Removal from Future Java Versions	991
28.12.1	Enhanced Deprecation	991
28.12.2	Items Likely to Be Removed in Future Java Versions	991
28.12.3	Finding Deprecated Features	992
28.12.4	Java Applets	992
28.13	Wrap-Up	992

A Operator Precedence Chart 994

B ASCII Character Set 996

C Keywords and Reserved Words 997

D Primitive Types 998

E	Bit Manipulation	999
E.1	Introduction	999
E.2	Bit Manipulation and the Bitwise Operators	999
E.3	BitSet Class	1009
F	Labeled break and continue Statements	1012
F.1	Introduction	1012
F.2	Labeled break Statement	1012
F.3	Labeled continue Statement	1013
	Index	1015

