

Contents

The online chapters and appendices listed at the end of this Table of Contents are located on the book's Companion Website (<http://www.pearsonglobaleditions.com/deitel>)—see the inside front cover of your book for details.

Foreword	25
Preface	27
Before You Begin	47
I Introduction to Computers, the Internet and Java	53
1.1 Introduction	54
1.2 Hardware and Software	56
1.2.1 Moore's Law	56
1.2.2 Computer Organization	57
1.3 Data Hierarchy	59
1.4 Machine Languages, Assembly Languages and High-Level Languages	61
1.5 Introduction to Object Technology	62
1.5.1 Automobile as an Object	63
1.5.2 Methods and Classes	63
1.5.3 Instantiation	63
1.5.4 Reuse	63
1.5.5 Messages and Method Calls	64
1.5.6 Attributes and Instance Variables	64
1.5.7 Encapsulation and Information Hiding	64
1.5.8 Inheritance	64
1.5.9 Interfaces	65
1.5.10 Object-Oriented Analysis and Design (OOAD)	65
1.5.11 The UML (Unified Modeling Language)	65
1.6 Operating Systems	66
1.6.1 Windows—A Proprietary Operating System	66
1.6.2 Linux—An Open-Source Operating System	66
1.6.3 Apple's macOS and Apple's iOS for iPhone®, iPad® and iPod Touch® Devices	67
1.6.4 Google's Android	67

1.7	Programming Languages	68
1.8	Java	70
1.9	A Typical Java Development Environment	71
1.10	Test-Driving a Java Application	74
1.11	Internet and World Wide Web	78
1.11.1	Internet: A Network of Networks	79
1.11.2	World Wide Web: Making the Internet User-Friendly	79
1.11.3	Web Services and Mashups	79
1.11.4	Internet of Things	80
1.12	Software Technologies	81
1.13	Getting Your Questions Answered	83

2 Introduction to Java Applications; Input/Output and Operators 87

2.1	Introduction	88
2.2	Your First Program in Java: Printing a Line of Text	88
2.2.1	Compiling the Application	92
2.2.2	Executing the Application	93
2.3	Modifying Your First Java Program	94
2.4	Displaying Text with <code>printf</code>	96
2.5	Another Application: Adding Integers	97
2.5.1	<code>import</code> Declarations	98
2.5.2	Declaring and Creating a <code>Scanner</code> to Obtain User Input from the Keyboard	98
2.5.3	Prompting the User for Input	99
2.5.4	Declaring a Variable to Store an Integer and Obtaining an Integer from the Keyboard	99
2.5.5	Obtaining a Second Integer	100
2.5.6	Using Variables in a Calculation	100
2.5.7	Displaying the Calculation Result	100
2.5.8	Java API Documentation	101
2.5.9	Declaring and Initializing Variables in Separate Statements	101
2.6	Memory Concepts	101
2.7	Arithmetic	102
2.8	Decision Making: Equality and Relational Operators	106
2.9	Wrap-Up	109

3 Introduction to Classes, Objects, Methods and Strings 120

3.1	Introduction	121
3.2	Instance Variables, <code>set</code> Methods and <code>get</code> Methods	122
3.2.1	Account Class with an Instance Variable, and <code>set</code> and <code>get</code> Methods	122
3.2.2	AccountTest Class That Creates and Uses an Object of Class Account	125

3.2.3	Compiling and Executing an App with Multiple Classes	128
3.2.4	Account UML Class Diagram	128
3.2.5	Additional Notes on Class <code>AccountTest</code>	130
3.2.6	Software Engineering with <code>private</code> Instance Variables and <code>public set</code> and <code>get</code> Methods	130
3.3	Account Class: Initializing Objects with Constructors	131
3.3.1	Declaring an Account Constructor for Custom Object Initialization	132
3.3.2	Class <code>AccountTest</code> : Initializing Account Objects When They're Created	133
3.4	Account Class with a Balance; Floating-Point Numbers	134
3.4.1	Account Class with a <code>balance</code> Instance Variable of Type <code>double</code>	135
3.4.2	<code>AccountTest</code> Class to Use Class <code>Account</code>	137
3.5	Primitive Types vs. Reference Types	140
3.6	(Optional) GUI and Graphics Case Study: A Simple GUI	140
3.6.1	What Is a Graphical User Interface?	142
3.6.2	JavaFX Scene Builder and FXML	142
3.6.3	Welcome App—Displaying Text and an Image	142
3.6.4	Opening Scene Builder and Creating the File <code>Welcome.fxml</code>	142
3.6.5	Adding an Image to the Folder Containing <code>Welcome.fxml</code>	144
3.6.6	Creating a <code>VBox</code> Layout Container	144
3.6.7	Configuring the <code>VBox</code>	144
3.6.8	Adding and Configuring a <code>Label</code>	144
3.6.9	Adding and Configuring an <code>ImageView</code>	146
3.6.10	Previewing the Welcome GUI	147
3.7	Wrap-Up	148

4 Control Statements: Part I; Assignment, ++ and -- Operators 156

4.1	Introduction	157
4.2	Algorithms	157
4.3	Pseudocode	158
4.4	Control Structures	158
4.4.1	Sequence Structure in Java	159
4.4.2	Selection Statements in Java	160
4.4.3	Iteration Statements in Java	160
4.4.4	Summary of Control Statements in Java	160
4.5	<code>if</code> Single-Selection Statement	161
4.6	<code>if...else</code> Double-Selection Statement	162
4.6.1	Nested <code>if...else</code> Statements	163
4.6.2	Dangling- <code>else</code> Problem	164
4.6.3	Blocks	164
4.6.4	Conditional Operator (<code>?:</code>)	165
4.7	<code>Student</code> Class: Nested <code>if...else</code> Statements	165
4.8	<code>while</code> Iteration Statement	168
4.9	Formulating Algorithms: Counter-Controlled Iteration	170

4.10	Formulating Algorithms: Sentinel-Controlled Iteration	174
4.11	Formulating Algorithms: Nested Control Statements	181
4.12	Compound Assignment Operators	185
4.13	Increment and Decrement Operators	186
4.14	Primitive Types	189
4.15	(Optional) GUI and Graphics Case Study: Event Handling; Drawing Lines	190
4.15.1	Test-Driving the Completed Draw Lines App	190
4.15.2	Building the App's GUI	191
4.15.3	Preparing to Interact with the GUI Programmatically	195
4.15.4	Class DrawLinesController	197
4.15.5	Class DrawLines—The Main Application Class	199
4.16	Wrap-Up	201

5 Control Statements: Part 2; Logical Operators 216

5.1	Introduction	217
5.2	Essentials of Counter-Controlled Iteration	217
5.3	for Iteration Statement	218
5.4	Examples Using the for Statement	222
5.4.1	Application: Summing the Even Integers from 2 to 20	223
5.4.2	Application: Compound-Interest Calculations	224
5.5	do...while Iteration Statement	227
5.6	switch Multiple-Selection Statement	228
5.7	Class AutoPolicy Case Study: Strings in switch Statements	234
5.8	break and continue Statements	237
5.8.1	break Statement	237
5.8.2	continue Statement	238
5.9	Logical Operators	239
5.9.1	Conditional AND (&&) Operator	239
5.9.2	Conditional OR () Operator	240
5.9.3	Short-Circuit Evaluation of Complex Conditions	241
5.9.4	Boolean Logical AND (&) and Boolean Logical Inclusive OR () Operators	241
5.9.5	Boolean Logical Exclusive OR (^)	242
5.9.6	Logical Negation (!) Operator	242
5.9.7	Logical Operators Example	243
5.10	Structured-Programming Summary	245
5.11	(Optional) GUI and Graphics Case Study: Drawing Rectangles and Ovals	250
5.12	Wrap-Up	253

6 Methods: A Deeper Look 264

6.1	Introduction	265
6.2	Program Units in Java	265
6.3	static Methods, static Fields and Class Math	267

6.4	Methods with Multiple Parameters	269
6.5	Notes on Declaring and Using Methods	273
6.6	Method-Call Stack and Activation Records	274
	6.6.1 Method-Call Stack	274
	6.6.2 Stack Frames	274
	6.6.3 Local Variables and Stack Frames	274
	6.6.4 Stack Overflow	275
6.7	Argument Promotion and Casting	275
6.8	Java API Packages	276
6.9	Case Study: Secure Random-Number Generation	278
6.10	Case Study: A Game of Chance; Introducing enum Types	283
6.11	Scope of Declarations	288
6.12	Method Overloading	290
	6.12.1 Declaring Overloaded Methods	290
	6.12.2 Distinguishing Between Overloaded Methods	291
	6.12.3 Return Types of Overloaded Methods	292
6.13	(Optional) GUI and Graphics Case Study: Colors and Filled Shapes	292
6.14	Wrap-Up	295

7 Arrays and ArrayLists 309

7.1	Introduction	310
7.2	Arrays	311
7.3	Declaring and Creating Arrays	312
7.4	Examples Using Arrays	314
	7.4.1 Creating and Initializing an Array	314
	7.4.2 Using an Array Initializer	315
	7.4.3 Calculating the Values to Store in an Array	316
	7.4.4 Summing the Elements of an Array	317
	7.4.5 Using Bar Charts to Display Array Data Graphically	317
	7.4.6 Using the Elements of an Array as Counters	319
	7.4.7 Using Arrays to Analyze Survey Results	320
7.5	Exception Handling: Processing the Incorrect Response	322
	7.5.1 The try Statement	322
	7.5.2 Executing the catch Block	322
	7.5.3 toString Method of the Exception Parameter	323
7.6	Case Study: Card Shuffling and Dealing Simulation	323
7.7	Enhanced for Statement	328
7.8	Passing Arrays to Methods	329
7.9	Pass-By-Value vs. Pass-By-Reference	331
7.10	Case Study: Class GradeBook Using an Array to Store Grades	332
7.11	Multidimensional Arrays	337
	7.11.1 Arrays of One-Dimensional Arrays	338
	7.11.2 Two-Dimensional Arrays with Rows of Different Lengths	338
	7.11.3 Creating Two-Dimensional Arrays with Array-Creation Expressions	339

12 Contents

7.11.4	Two-Dimensional Array Example: Displaying Element Values	339
7.11.5	Common Multidimensional-Array Manipulations Performed with <code>for</code> Statements	340
7.12	Case Study: Class <code>GradeBook</code> Using a Two-Dimensional Array	341
7.13	Variable-Length Argument Lists	347
7.14	Using Command-Line Arguments	348
7.15	Class Arrays	350
7.16	Introduction to Collections and Class <code>ArrayList</code>	353
7.17	(Optional) GUI and Graphics Case Study: Drawing Arcs	357
7.18	Wrap-Up	360

8 Classes and Objects: A Deeper Look 381

8.1	Introduction	382
8.2	Time Class Case Study	382
8.3	Controlling Access to Members	387
8.4	Referring to the Current Object's Members with the <code>this</code> Reference	388
8.5	Time Class Case Study: Overloaded Constructors	390
8.6	Default and No-Argument Constructors	395
8.7	Notes on <i>Set</i> and <i>Get</i> Methods	396
8.8	Composition	397
8.9	<code>enum</code> Types	400
8.10	Garbage Collection	403
8.11	<code>static</code> Class Members	403
8.12	<code>static</code> Import	407
8.13	<code>final</code> Instance Variables	408
8.14	Package Access	409
8.15	Using <code>BigDecimal</code> for Precise Monetary Calculations	410
8.16	(Optional) GUI and Graphics Case Study: Using Objects with Graphics	413
8.17	Wrap-Up	417

9 Object-Oriented Programming: Inheritance 425

9.1	Introduction	426
9.2	Superclasses and Subclasses	427
9.3	<code>protected</code> Members	429
9.4	Relationship Between Superclasses and Subclasses	430
9.4.1	Creating and Using a <code>CommissionEmployee</code> Class	430
9.4.2	Creating and Using a <code>BasePlusCommissionEmployee</code> Class	435
9.4.3	Creating a <code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy	440
9.4.4	<code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy Using <code>protected</code> Instance Variables	443
9.4.5	<code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy Using <code>private</code> Instance Variables	446
9.5	Constructors in Subclasses	450

9.6	Class Object	451
9.7	Designing with Composition vs. Inheritance	452
9.8	Wrap-Up	454

10 Object-Oriented Programming: Polymorphism and Interfaces **459**

10.1	Introduction	460
10.2	Polymorphism Examples	462
10.3	Demonstrating Polymorphic Behavior	463
10.4	Abstract Classes and Methods	465
10.5	Case Study: Payroll System Using Polymorphism	468
10.5.1	Abstract Superclass Employee	469
10.5.2	Concrete Subclass SalariedEmployee	471
10.5.3	Concrete Subclass HourlyEmployee	473
10.5.4	Concrete Subclass CommissionEmployee	474
10.5.5	Indirect Concrete Subclass BasePlusCommissionEmployee	476
10.5.6	Polymorphic Processing, Operator instanceof and Downcasting	477
10.6	Allowed Assignments Between Superclass and Subclass Variables	482
10.7	final Methods and Classes	482
10.8	A Deeper Explanation of Issues with Calling Methods from Constructors	483
10.9	Creating and Using Interfaces	484
10.9.1	Developing a Payable Hierarchy	486
10.9.2	Interface Payable	487
10.9.3	Class Invoice	487
10.9.4	Modifying Class Employee to Implement Interface Payable	489
10.9.5	Using Interface Payable to Process Invoices and Employees Polymorphically	491
10.9.6	Some Common Interfaces of the Java API	492
10.10	Java SE 8 Interface Enhancements	493
10.10.1	default Interface Methods	493
10.10.2	static Interface Methods	494
10.10.3	Functional Interfaces	494
10.11	Java SE 9 private Interface Methods	495
10.12	private Constructors	495
10.13	Program to an Interface, Not an Implementation	496
10.13.1	Implementation Inheritance Is Best for Small Numbers of Tightly Coupled Classes	496
10.13.2	Interface Inheritance Is Best for Flexibility	496
10.13.3	Rethinking the Employee Hierarchy	497
10.14	(Optional) GUI and Graphics Case Study: Drawing with Polymorphism	498
10.15	Wrap-Up	500

11 Exception Handling: A Deeper Look **507**

11.1	Introduction	508
------	--------------	-----

11.2	Example: Divide by Zero without Exception Handling	509
11.3	Example: Handling <code>ArithmeticExceptions</code> and <code>InputMismatchExceptions</code>	511
11.4	When to Use Exception Handling	517
11.5	Java Exception Hierarchy	517
11.6	<code>finally</code> Block	521
11.7	Stack Unwinding and Obtaining Information from an Exception	525
11.8	Chained Exceptions	528
11.9	Declaring New Exception Types	530
11.10	Preconditions and Postconditions	531
11.11	Assertions	531
11.12	<code>try-with-Resources</code> : Automatic Resource Deallocation	533
11.13	Wrap-Up	534

12 JavaFX Graphical User Interfaces: Part 1 **540**

12.1	Introduction	541
12.2	JavaFX Scene Builder	542
12.3	JavaFX App Window Structure	543
12.4	Welcome App—Displaying Text and an Image	544
12.4.1	Opening Scene Builder and Creating the File <code>Welcome.fxml</code>	544
12.4.2	Adding an Image to the Folder Containing <code>Welcome.fxml</code>	545
12.4.3	Creating a <code>VBox</code> Layout Container	545
12.4.4	Configuring the <code>VBox</code> Layout Container	546
12.4.5	Adding and Configuring a <code>Label</code>	546
12.4.6	Adding and Configuring an <code>ImageView</code>	547
12.4.7	Previewing the Welcome GUI	549
12.5	Tip Calculator App—Introduction to Event Handling	549
12.5.1	Test-Driving the Tip Calculator App	550
12.5.2	Technologies Overview	551
12.5.3	Building the App's GUI	553
12.5.4	<code>TipCalculator</code> Class	560
12.5.5	<code>TipCalculatorController</code> Class	562
12.6	Features Covered in the Other JavaFX Chapters	567
12.7	Wrap-Up	567

13 JavaFX GUI: Part 2 **575**

13.1	Introduction	576
13.2	Laying Out Nodes in a Scene Graph	576
13.3	Painter App: <code>RadioButtons</code> , Mouse Events and Shapes	578
13.3.1	Technologies Overview	578
13.3.2	Creating the <code>Painter.fxml</code> File	580
13.3.3	Building the GUI	580
13.3.4	<code>Painter</code> Subclass of <code>Application</code>	583
13.3.5	<code>PainterController</code> Class	584

13.4	Color Chooser App: Property Bindings and Property Listeners	588
13.4.1	Technologies Overview	588
13.4.2	Building the GUI	589
13.4.3	ColorChooser Subclass of Application	591
13.4.4	ColorChooserController Class	592
13.5	Cover Viewer App: Data-Driven GUIs with JavaFX Collections	594
13.5.1	Technologies Overview	595
13.5.2	Adding Images to the App's Folder	595
13.5.3	Building the GUI	595
13.5.4	CoverViewer Subclass of Application	597
13.5.5	CoverViewerController Class	597
13.6	Cover Viewer App: Customizing ListView Cells	599
13.6.1	Technologies Overview	600
13.6.2	Copying the CoverViewer App	600
13.6.3	ImageTextCell Custom Cell Factory Class	601
13.6.4	CoverViewerController Class	602
13.7	Additional JavaFX Capabilities	603
13.8	JavaFX 9: Java SE 9 JavaFX Updates	605
13.9	Wrap-Up	607

14 Strings, Characters and Regular Expressions 616

14.1	Introduction	617
14.2	Fundamentals of Characters and Strings	617
14.3	Class String	618
14.3.1	String Constructors	618
14.3.2	String Methods length, charAt and getChars	619
14.3.3	Comparing Strings	621
14.3.4	Locating Characters and Substrings in Strings	625
14.3.5	Extracting Substrings from Strings	627
14.3.6	Concatenating Strings	628
14.3.7	Miscellaneous String Methods	629
14.3.8	String Method valueOf	630
14.4	Class StringBuilder	631
14.4.1	StringBuilder Constructors	632
14.4.2	StringBuilder Methods length, capacity, setLength and ensureCapacity	633
14.4.3	StringBuilder Methods charAt, setCharAt, getChars and reverse	634
14.4.4	StringBuilder append Methods	635
14.4.5	StringBuilder Insertion and Deletion Methods	637
14.5	Class Character	638
14.6	Tokenizing Strings	643
14.7	Regular Expressions, Class Pattern and Class Matcher	644
14.7.1	Replacing Substrings and Splitting Strings	649
14.7.2	Classes Pattern and Matcher	651
14.8	Wrap-Up	653

15	Files, Input/Output Streams, NIO and XML Serialization	664
15.1	Introduction	665
15.2	Files and Streams	665
15.3	Using NIO Classes and Interfaces to Get File and Directory Information	667
15.4	Sequential Text Files	671
15.4.1	Creating a Sequential Text File	671
15.4.2	Reading Data from a Sequential Text File	674
15.4.3	Case Study: A Credit-Inquiry Program	675
15.4.4	Updating Sequential Files	680
15.5	XML Serialization	680
15.5.1	Creating a Sequential File Using XML Serialization	680
15.5.2	Reading and Deserializing Data from a Sequential File	686
15.6	FileChooser and DirectoryChooser Dialogs	687
15.7	(Optional) Additional java.io Classes	693
15.7.1	Interfaces and Classes for Byte-Based Input and Output	693
15.7.2	Interfaces and Classes for Character-Based Input and Output	695
15.8	Wrap-Up	696
16	Generic Collections	704
16.1	Introduction	705
16.2	Collections Overview	705
16.3	Type-Wrapper Classes	707
16.4	Autoboxing and Auto-Unboxing	707
16.5	Interface Collection and Class Collections	707
16.6	Lists	708
16.6.1	ArrayList and Iterator	709
16.6.2	LinkedList	711
16.7	Collections Methods	716
16.7.1	Method sort	716
16.7.2	Method shuffle	720
16.7.3	Methods reverse, fill, copy, max and min	722
16.7.4	Method binarySearch	724
16.7.5	Methods addAll, frequency and disjoint	725
16.8	Class PriorityQueue and Interface Queue	727
16.9	Sets	728
16.10	Maps	731
16.11	Synchronized Collections	735
16.12	Unmodifiable Collections	735
16.13	Abstract Implementations	736
16.14	Java SE 9: Convenience Factory Methods for Immutable Collections	736
16.15	Wrap-Up	740

17	Lambdas and Streams	746
17.1	Introduction	747
17.2	Streams and Reduction	749
17.2.1	Summing the Integers from 1 through 10 with a for Loop	749
17.2.2	External Iteration with for Is Error Prone	750
17.2.3	Summing with a Stream and Reduction	750
17.2.4	Internal Iteration	751
17.3	Mapping and Lambdas	752
17.3.1	Lambda Expressions	753
17.3.2	Lambda Syntax	754
17.3.3	Intermediate and Terminal Operations	755
17.4	Filtering	756
17.5	How Elements Move Through Stream Pipelines	758
17.6	Method References	759
17.6.1	Creating an IntStream of Random Values	760
17.6.2	Performing a Task on Each Stream Element with forEach and a Method Reference	760
17.6.3	Mapping Integers to String Objects with mapToObj	761
17.6.4	Concatenating Strings with collect	761
17.7	IntStream Operations	762
17.7.1	Creating an IntStream and Displaying Its Values	763
17.7.2	Terminal Operations count, min, max, sum and average	763
17.7.3	Terminal Operation reduce	764
17.7.4	Sorting IntStream Values	766
17.8	Functional Interfaces	767
17.9	Lambdas: A Deeper Look	768
17.10	Stream<Integer> Manipulations	769
17.10.1	Creating a Stream<Integer>	770
17.10.2	Sorting a Stream and Collecting the Results	771
17.10.3	Filtering a Stream and Storing the Results for Later Use	771
17.10.4	Filtering and Sorting a Stream and Collecting the Results	772
17.10.5	Sorting Previously Collected Results	772
17.11	Stream<String> Manipulations	772
17.11.1	Mapping Strings to Uppercase	773
17.11.2	Filtering Strings Then Sorting Them in Case-Insensitive Ascending Order	774
17.11.3	Filtering Strings Then Sorting Them in Case-Insensitive Descending Order	774
17.12	Stream<Employee> Manipulations	775
17.12.1	Creating and Displaying a List<Employee>	776
17.12.2	Filtering Employees with Salaries in a Specified Range	777
17.12.3	Sorting Employees By Multiple Fields	780
17.12.4	Mapping Employees to Unique-Last-Name Strings	782
17.12.5	Grouping Employees By Department	783
17.12.6	Counting the Number of Employees in Each Department	784
17.12.7	Summing and Averaging Employee Salaries	785

17.13	Creating a <code>Stream<String></code> from a File	786
17.14	Streams of Random Values	789
17.15	Infinite Streams	791
17.16	Lambda Event Handlers	793
17.17	Additional Notes on Java SE 8 Interfaces	793
17.18	Wrap-Up	794

18 Recursion 808

18.1	Introduction	809
18.2	Recursion Concepts	810
18.3	Example Using Recursion: Factorials	811
18.4	Reimplementing Class <code>FactorialCalculator</code> Using <code>BigInteger</code>	813
18.5	Example Using Recursion: Fibonacci Series	815
18.6	Recursion and the Method-Call Stack	818
18.7	Recursion vs. Iteration	819
18.8	Towers of Hanoi	821
18.9	Fractals	823
18.9.1	Koch Curve Fractal	824
18.9.2	(Optional) Case Study: Lo Feather Fractal	825
18.9.3	(Optional) Fractal App GUI	827
18.9.4	(Optional) <code>FractalController</code> Class	829
18.10	Recursive Backtracking	834
18.11	Wrap-Up	834

19 Searching, Sorting and Big O 843

19.1	Introduction	844
19.2	Linear Search	845
19.3	Big O Notation	848
19.3.1	$O(1)$ Algorithms	848
19.3.2	$O(n)$ Algorithms	848
19.3.3	$O(n^2)$ Algorithms	848
19.3.4	Big O of the Linear Search	849
19.4	Binary Search	849
19.4.1	Binary Search Implementation	850
19.4.2	Efficiency of the Binary Search	853
19.5	Sorting Algorithms	854
19.6	Selection Sort	854
19.6.1	Selection Sort Implementation	855
19.6.2	Efficiency of the Selection Sort	857
19.7	Insertion Sort	857
19.7.1	Insertion Sort Implementation	858
19.7.2	Efficiency of the Insertion Sort	860
19.8	Merge Sort	861
19.8.1	Merge Sort Implementation	861
19.8.2	Efficiency of the Merge Sort	866

19.9	Big O Summary for This Chapter's Searching and Sorting Algorithms	866
19.10	Massive Parallelism and Parallel Algorithms	867
19.11	Wrap-Up	867
20	Generic Classes and Methods: A Deeper Look	873
20.1	Introduction	874
20.2	Motivation for Generic Methods	874
20.3	Generic Methods: Implementation and Compile-Time Translation	876
20.4	Additional Compile-Time Translation Issues: Methods That Use a Type Parameter as the Return Type	879
20.5	Overloading Generic Methods	882
20.6	Generic Classes	883
20.7	Wildcards in Methods That Accept Type Parameters	890
20.8	Wrap-Up	894
21	Custom Generic Data Structures	898
21.1	Introduction	899
21.2	Self-Referential Classes	900
21.3	Dynamic Memory Allocation	900
21.4	Linked Lists	901
21.4.1	Singly Linked Lists	901
21.4.2	Implementing a Generic List Class	902
21.4.3	Generic Classes ListNode and List	905
21.4.4	Class ListTest	905
21.4.5	List Method insertAtFront	907
21.4.6	List Method insertAtBack	908
21.4.7	List Method removeFromFront	908
21.4.8	List Method removeFromBack	909
21.4.9	List Method print	910
21.4.10	Creating Your Own Packages	910
21.5	Stacks	915
21.6	Queues	918
21.7	Trees	920
21.8	Wrap-Up	927
22	JavaFX Graphics and Multimedia	952
22.1	Introduction	953
22.2	Controlling Fonts with Cascading Style Sheets (CSS)	954
22.2.1	CSS That Styles the GUI	954
22.2.2	FXML That Defines the GUI—Introduction to XML Markup	957
22.2.3	Referencing the CSS File from FXML	960
22.2.4	Specifying the VBox's Style Class	960
22.2.5	Programmatically Loading CSS	960

22.3	Displaying Two-Dimensional Shapes	961
22.3.1	Defining Two-Dimensional Shapes with FXML	961
22.3.2	CSS That Styles the Two-Dimensional Shapes	964
22.4	Polylines, Polygons and Paths	966
22.4.1	GUI and CSS	967
22.4.2	PolyShapesController Class	968
22.5	Transforms	971
22.6	Playing Video with Media, MediaPlayer and MediaPlayer	973
22.6.1	VideoPlayer GUI	974
22.6.2	VideoPlayerController Class	976
22.7	Transition Animations	980
22.7.1	TransitionAnimations.fxml	980
22.7.2	TransitionAnimationsController Class	982
22.8	Timeline Animations	986
22.9	Frame-by-Frame Animation with AnimationTimer	989
22.10	Drawing on a Canvas	991
22.11	Three-Dimensional Shapes	996
22.12	Wrap-Up	999

23 Concurrency

1015

23.1	Introduction	1016
23.2	Thread States and Life Cycle	1018
23.2.1	<i>New</i> and <i>Runnable</i> States	1019
23.2.2	<i>Waiting</i> State	1019
23.2.3	<i>Timed Waiting</i> State	1019
23.2.4	<i>Blocked</i> State	1019
23.2.5	<i>Terminated</i> State	1019
23.2.6	Operating-System View of the <i>Runnable</i> State	1020
23.2.7	Thread Priorities and Thread Scheduling	1020
23.2.8	Indefinite Postponement and Deadlock	1021
23.3	Creating and Executing Threads with the Executor Framework	1021
23.4	Thread Synchronization	1025
23.4.1	Immutable Data	1026
23.4.2	Monitors	1026
23.4.3	Unsynchronized Mutable Data Sharing	1027
23.4.4	Synchronized Mutable Data Sharing—Making Operations Atomic	1031
23.5	Producer/Consumer Relationship without Synchronization	1034
23.6	Producer/Consumer Relationship: ArrayBlockingQueue	1042
23.7	(Advanced) Producer/Consumer Relationship with synchronized, wait, notify and notifyAll	1045
23.8	(Advanced) Producer/Consumer Relationship: Bounded Buffers	1051
23.9	(Advanced) Producer/Consumer Relationship: The Lock and Condition Interfaces	1059
23.10	Concurrent Collections	1066
23.11	Multithreading in JavaFX	1068

23.11.1	Performing Computations in a Worker Thread: Fibonacci Numbers	1069
23.11.2	Processing Intermediate Results: Sieve of Eratosthenes	1074
23.12	sort/parallelSort Timings with the Java SE 8 Date/Time API	1080
23.13	Java SE 8: Sequential vs. Parallel Streams	1083
23.14	(Advanced) Interfaces Callable and Future	1085
23.15	(Advanced) Fork/Join Framework	1090
23.16	Wrap-Up	1090

24 Accessing Databases with JDBC **1102**

24.1	Introduction	1103
24.2	Relational Databases	1104
24.3	A books Database	1105
24.4	SQL	1109
24.4.1	Basic SELECT Query	1110
24.4.2	WHERE Clause	1110
24.4.3	ORDER BY Clause	1112
24.4.4	Merging Data from Multiple Tables: INNER JOIN	1114
24.4.5	INSERT Statement	1115
24.4.6	UPDATE Statement	1116
24.4.7	DELETE Statement	1117
24.5	Setting Up a Java DB Database	1118
24.5.1	Creating the Chapter's Databases on Windows	1119
24.5.2	Creating the Chapter's Databases on macOS	1120
24.5.3	Creating the Chapter's Databases on Linux	1120
24.6	Connecting to and Querying a Database	1120
24.6.1	Automatic Driver Discovery	1122
24.6.2	Connecting to the Database	1122
24.6.3	Creating a Statement for Executing Queries	1123
24.6.4	Executing a Query	1123
24.6.5	Processing a Query's ResultSet	1124
24.7	Querying the books Database	1125
24.7.1	ResultSetTableModel Class	1125
24.7.2	DisplayQueryResults App's GUI	1132
24.7.3	DisplayQueryResultsController Class	1132
24.8	RowSet Interface	1137
24.9	PreparedStatement	1140
24.9.1	AddressBook App That Uses PreparedStatement	1141
24.9.2	Class Person	1141
24.9.3	Class PersonQueries	1143
24.9.4	AddressBook GUI	1146
24.9.5	Class AddressBookController	1147
24.10	Stored Procedures	1152
24.11	Transaction Processing	1152
24.12	Wrap-Up	1153

25	Introduction to JShell: Java 9's REPL	1161
25.1	Introduction	1162
25.2	Installing JDK 9	1164
25.3	Introduction to JShell	1164
25.3.1	Starting a JShell Session	1165
25.3.2	Executing Statements	1165
25.3.3	Declaring Variables Explicitly	1166
25.3.4	Listing and Executing Prior Snippets	1168
25.3.5	Evaluating Expressions and Declaring Variables Implicitly	1170
25.3.6	Using Implicitly Declared Variables	1170
25.3.7	Viewing a Variable's Value	1171
25.3.8	Resetting a JShell Session	1171
25.3.9	Writing Multiline Statements	1171
25.3.10	Editing Code Snippets	1172
25.3.11	Exiting JShell	1175
25.4	Command-Line Input in JShell	1175
25.5	Declaring and Using Classes	1176
25.5.1	Creating a Class in JShell	1177
25.5.2	Explicitly Declaring Reference-Type Variables	1177
25.5.3	Creating Objects	1178
25.5.4	Manipulating Objects	1178
25.5.5	Creating a Meaningful Variable Name for an Expression	1179
25.5.6	Saving and Opening Code-Snippet Files	1180
25.6	Discovery with JShell Auto-Completion	1180
25.6.1	Auto-Completing Identifiers	1181
25.6.2	Auto-Completing JShell Commands	1182
25.7	Exploring a Class's Members and Viewing Documentation	1182
25.7.1	Listing Class <code>Math</code> 's <code>static</code> Members	1183
25.7.2	Viewing a Method's Parameters	1183
25.7.3	Viewing a Method's Documentation	1184
25.7.4	Viewing a <code>public</code> Field's Documentation	1184
25.7.5	Viewing a Class's Documentation	1185
25.7.6	Viewing Method Overloads	1185
25.7.7	Exploring Members of a Specific Object	1186
25.8	Declaring Methods	1188
25.8.1	Forward Referencing an Undeclared Method—Declaring Method <code>displayCubes</code>	1188
25.8.2	Declaring a Previously Undeclared Method	1188
25.8.3	Testing <code>cube</code> and Replacing Its Declaration	1189
25.8.4	Testing Updated Method <code>cube</code> and Method <code>displayCubes</code>	1189
25.9	Exceptions	1190
25.10	Importing Classes and Adding Packages to the <code>CLASSPATH</code>	1191
25.11	Using an External Editor	1193
25.12	Summary of JShell Commands	1195
25.12.1	Getting Help in JShell	1196
25.12.2	<code>/edit</code> Command: Additional Features	1197

25.12.3	/reload Command	1197
25.12.4	/drop Command	1198
25.12.5	Feedback Modes	1198
25.12.6	Other JShell Features Configurable with /set	1200
25.13	Keyboard Shortcuts for Snippet Editing	1201
25.14	How JShell Reinterprets Java for Interactive Use	1201
25.15	IDE JShell Support	1202
25.16	Wrap-Up	1202

Chapters on the Web **1218**

A Operator Precedence Chart **1219**

B ASCII Character Set **1221**

C Keywords and Reserved Words **1222**

D Primitive Types **1223**

E Using the Debugger **1224**

E.1	Introduction	1225
E.2	Breakpoints and the run, stop, cont and print Commands	1225
E.3	The print and set Commands	1229
E.4	Controlling Execution Using the step, step up and next Commands	1231
E.5	The watch Command	1233
E.6	The clear Command	1235
E.7	Wrap-Up	1238

Appendices on the Web **1239**

Index **1241**

Online Chapters and Appendices

The online chapters and appendices are located on the book's Companion Website. See the book's inside front cover for details.

26 Swing GUI Components: Part I

27 Graphics and Java 2D

- 28 Networking**
- 29 Java Persistence API (JPA)**
- 30 JavaServer™ Faces Web Apps: Part 1**
- 31 JavaServer™ Faces Web Apps: Part 2**
- 32 REST-Based Web Services**
- 33 (Optional) ATM Case Study, Part 1:
Object-Oriented Design with the UML**
- 34 (Optional) ATM Case Study, Part 2:
Implementing an Object-Oriented Design**
- 35 Swing GUI Components: Part 2**
- 36 Java Module System and Other Java 9 Features**
- F Using the Java API Documentation**
- G Creating Documentation with javadoc**
- H Unicode®**
- I Formatted Output**
- J Number Systems**
- K Bit Manipulation**
- L Labeled break and continue Statements**
- M UML 2: Additional Diagram Types**
- N Design Patterns**