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

Foreword xxv

Preface xxvii

Before You Begin xlvii

I Introduction to Computers, the Internet and Java 1
  1.1 Introduction 2
  1.2 Hardware and Software 4
    1.2.1 Moore’s Law 4
    1.2.2 Computer Organization 5
  1.3 Data Hierarchy 7
  1.4 Machine Languages, Assembly Languages and High-Level Languages 9
  1.5 Introduction to Object Technology 10
    1.5.1 Automobile as an Object 11
    1.5.2 Methods and Classes 11
    1.5.3 Instantiation 11
    1.5.4 Reuse 11
    1.5.5 Messages and Method Calls 12
    1.5.6 Attributes and Instance Variables 12
    1.5.7 Encapsulation and Information Hiding 12
    1.5.8 Inheritance 12
    1.5.9 Interfaces 13
    1.5.10 Object-Oriented Analysis and Design (OOAD) 13
    1.5.11 The UML (Unified Modeling Language) 13
  1.6 Operating Systems 14
    1.6.1 Windows—A Proprietary Operating System 14
    1.6.2 Linux—An Open-Source Operating System 14
    1.6.3 Apple’s macOS and Apple’s iOS for iPhone®, iPad® and iPod Touch® Devices 15
    1.6.4 Google’s Android 15
# Contents

1.7 Programming Languages 16
1.8 Java 18
1.9 A Typical Java Development Environment 19
1.10 Test-Driving a Java Application 22
1.11 Internet and World Wide Web 26
  1.11.1 Internet: A Network of Networks 27
  1.11.2 World Wide Web: Making the Internet User-Friendly 27
  1.11.3 Web Services and Mashups 27
  1.11.4 Internet of Things 28
1.12 Software Technologies 29
1.13 Getting Your Questions Answered 31

2 Introduction to Java Applications; Input/Output and Operators 35
  2.1 Introduction 36
  2.2 Your First Program in Java: Printing a Line of Text 36
    2.2.1 Compiling the Application 40
    2.2.2 Executing the Application 41
  2.3 Modifying Your First Java Program 42
  2.4 Displaying Text with `printf` 44
  2.5 Another Application: Adding Integers 45
    2.5.1 `import` Declarations 46
    2.5.2 Declaring and Creating a `Scanner` to Obtain User Input from the Keyboard 46
    2.5.3 Prompting the User for Input 47
    2.5.4 Declaring a Variable to Store an Integer and Obtaining an Integer from the Keyboard 47
    2.5.5 Obtaining a Second Integer 48
    2.5.6 Using Variables in a Calculation 48
    2.5.7 Displaying the Calculation Result 48
    2.5.8 Java API Documentation 49
    2.5.9 Declaring and Initializing Variables in Separate Statements 49
  2.6 Memory Concepts 49
  2.7 Arithmetic 50
  2.8 Decision Making: Equality and Relational Operators 54
  2.9 Wrap-Up 57

3 Introduction to Classes, Objects, Methods and Strings 68
  3.1 Introduction 69
  3.2 Instance Variables, `set` Methods and `get` Methods 70
    3.2.1 `Account` Class with an Instance Variable, and `set` and `get` Methods 70
    3.2.2 `AccountTest` Class That Creates and Uses an Object of Class `Account` 73
3.2.3 Compiling and Executing an App with Multiple Classes 76
3.2.4 Account UML Class Diagram 76
3.2.5 Additional Notes on Class AccountTest 78
3.2.6 Software Engineering with private Instance Variables and public set and get Methods 78
3.3 Account Class: Initializing Objects with Constructors 79
3.3.1 Declaring an Account Constructor for Custom Object Initialization 80
3.3.2 Class AccountTest: Initializing Account Objects When They’re Created 81
3.4 Account Class with a Balance; Floating-Point Numbers 82
3.4.1 Account Class with a balance Instance Variable of Type double 83
3.4.2 AccountTest Class to Use Class Account 85
3.5 Primitive Types vs. Reference Types 88
3.6 (Optional) GUI and Graphics Case Study: A Simple GUI 88
3.6.1 What Is a Graphical User Interface? 90
3.6.2 JavaFX Scene Builder and FXML 90
3.6.3 Welcome App—Displaying Text and an Image 90
3.6.4 Opening Scene Builder and Creating the File Welcome.fxml 90
3.6.5 Adding an Image to the Folder Containing Welcome.fxml 92
3.6.6 Creating a VBox Layout Container 92
3.6.7 Configuring the VBox 92
3.6.8 Adding and Configuring a Label 92
3.6.9 Adding and Configuring an ImageView 94
3.6.10 Previewing the Welcome GUI 95
3.7 Wrap-Up 96

4 Control Statements: Part 1; Assignment, ++ and -- Operators 104
4.1 Introduction 105
4.2 Algorithms 105
4.3 Pseudocode 106
4.4 Control Structures 106
4.4.1 Sequence Structure in Java 107
4.4.2 Selection Statements in Java 108
4.4.3 Iteration Statements in Java 108
4.4.4 Summary of Control Statements in Java 108
4.5 if Single-Selection Statement 109
4.6 if...else Double-Selection Statement 110
4.6.1 Nested if...else Statements 111
4.6.2 Dangling-else Problem 112
4.6.3 Blocks 112
4.6.4 Conditional Operator (?:) 113
4.7 Student Class: Nested if...else Statements 113
4.8 while Iteration Statement 116
4.9 Formulating Algorithms: Counter-Controlled Iteration 118

©Copyright 2018 by Pearson Education, Inc. All Rights Reserved.
## 4.10 Formulating Algorithms: Sentinel-Controlled Iteration 122

## 4.11 Formulating Algorithms: Nested Control Statements 129

## 4.12 Compound Assignment Operators 133

## 4.13 Increment and Decrement Operators 134

## 4.14 Primitive Types 137

## 4.15 (Optional) GUI and Graphics Case Study: Event Handling; Drawing Lines 138

### 4.15.1 Test-Driving the Completed Draw Lines App 138

### 4.15.2 Building the App’s GUI 139

### 4.15.3 Preparing to Interact with the GUI Programmatically 143

### 4.15.4 Class DrawLinesController 145

### 4.15.5 Class DrawLines — The Main Application Class 147

## 4.16 Wrap-Up 149

## 5 Control Statements: Part 2; Logical Operators 164

### 5.1 Introduction 165

### 5.2 Essentials of Counter-Controlled Iteration 165

### 5.3 for Iteration Statement 166

### 5.4 Examples Using the for Statement 170

#### 5.4.1 Application: Summing the Even Integers from 2 to 20 171

#### 5.4.2 Application: Compound-Interest Calculations 172

### 5.5 do…while Iteration Statement 175

### 5.6 switch Multiple-Selection Statement 176

### 5.7 Class AutoPolicy Case Study: Strings in switch Statements 182

### 5.8 break and continue Statements 185

#### 5.8.1 break Statement 185

#### 5.8.2 continue Statement 186

### 5.9 Logical Operators 187

#### 5.9.1 Conditional AND (&&) Operator 187

#### 5.9.2 Conditional OR (||) Operator 188

#### 5.9.3 Short-Circuit Evaluation of Complex Conditions 189

#### 5.9.4 Boolean Logical AND (&) and Boolean Logical Inclusive OR (|) Operators 189

#### 5.9.5 Boolean Logical Exclusive OR (^) 190

#### 5.9.6 Logical Negation (!) Operator 190

#### 5.9.7 Logical Operators Example 191

### 5.10 Structured-Programming Summary 193

### 5.11 (Optional) GUI and Graphics Case Study: Drawing Rectangles and Ovals 198

### 5.12 Wrap-Up 201

## 6 Methods: A Deeper Look 212

### 6.1 Introduction 213

### 6.2 Program Units in Java 213

### 6.3 static Methods, static Fields and Class Math 215

©Copyright 2018 by Pearson Education, Inc. All Rights Reserved.
6.4 Methods with Multiple Parameters 217
6.5 Notes on Declaring and Using Methods 221
6.6 Method-Call Stack and Activation Records 222
   6.6.1 Method-Call Stack 222
   6.6.2 Stack Frames 222
   6.6.3 Local Variables and Stack Frames 222
   6.6.4 Stack Overflow 223
6.7 Argument Promotion and Casting 223
6.8 Java API Packages 224
6.9 Case Study: Secure Random-Number Generation 226
6.10 Case Study: A Game of Chance; Introducing enum Types 231
6.11 Scope of Declarations 236
6.12 Method Overloading 238
   6.12.1 Declaring Overloaded Methods 238
   6.12.2 Distinguishing Between Overloaded Methods 239
   6.12.3 Return Types of Overloaded Methods 240
6.13 (Optional) GUI and Graphics Case Study: Colors and Filled Shapes 240
6.14 Wrap-Up 243

7 Arrays and ArrayLists 257
7.1 Introduction 258
7.2 Arrays 259
7.3 Declaring and Creating Arrays 260
7.4 Examples Using Arrays 262
   7.4.1 Creating and Initializing an Array 262
   7.4.2 Using an Array Initializer 263
   7.4.3 Calculating the Values to Store in an Array 264
   7.4.4 Summing the Elements of an Array 265
   7.4.5 Using Bar Charts to Display Array Data Graphically 265
   7.4.6 Using the Elements of an Array as Counters 267
   7.4.7 Using Arrays to Analyze Survey Results 268
7.5 Exception Handling: Processing the Incorrect Response 270
   7.5.1 The try Statement 270
   7.5.2 Executing the catch Block 270
   7.5.3 toString Method of the Exception Parameter 271
7.6 Case Study: Card Shuffling and Dealing Simulation 271
7.7 Enhanced for Statement 276
7.8 Passing Arrays to Methods 277
7.9 Pass-By-Value vs. Pass-By-Reference 279
7.10 Case Study: Class GradeBook Using an Array to Store Grades 280
7.11 Multidimensional Arrays 285
   7.11.1 Arrays of One-Dimensional Arrays 286
   7.11.2 Two-Dimensional Arrays with Rows of Different Lengths 286
   7.11.3 Creating Two-Dimensional Arrays with Array-Creation Expressions 287

©Copyright 2018 by Pearson Education, Inc. All Rights Reserved.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.11.4</td>
<td>Two-Dimensional Array Example: Displaying Element Values</td>
<td>287</td>
</tr>
<tr>
<td>7.11.5</td>
<td>Common Multidimensional-Array Manipulations Performed with for Statements</td>
<td>288</td>
</tr>
<tr>
<td>7.12</td>
<td>Case Study: Class GradeBook Using a Two-Dimensional Array</td>
<td>289</td>
</tr>
<tr>
<td>7.13</td>
<td>Variable-Length Argument Lists</td>
<td>295</td>
</tr>
<tr>
<td>7.14</td>
<td>Using Command-Line Arguments</td>
<td>296</td>
</tr>
<tr>
<td>7.15</td>
<td>Class Arrays</td>
<td>298</td>
</tr>
<tr>
<td>7.16</td>
<td>Introduction to Collections and Class ArrayList</td>
<td>301</td>
</tr>
<tr>
<td>7.17</td>
<td>(Optional) GUI and Graphics Case Study: Drawing Arcs</td>
<td>305</td>
</tr>
<tr>
<td>7.18</td>
<td>Wrap-Up</td>
<td>308</td>
</tr>
</tbody>
</table>

## 8 Classes and Objects: A Deeper Look

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>Introduction</td>
<td>330</td>
</tr>
<tr>
<td>8.2</td>
<td>Time Class Case Study</td>
<td>330</td>
</tr>
<tr>
<td>8.3</td>
<td>Controlling Access to Members</td>
<td>335</td>
</tr>
<tr>
<td>8.4</td>
<td>Referring to the Current Object’s Members with the this Reference</td>
<td>336</td>
</tr>
<tr>
<td>8.5</td>
<td>Time Class Case Study: Overloaded Constructors</td>
<td>338</td>
</tr>
<tr>
<td>8.6</td>
<td>Default and No-Argument Overloaded Constructors</td>
<td>343</td>
</tr>
<tr>
<td>8.7</td>
<td>Notes on Set and Get Methods</td>
<td>344</td>
</tr>
<tr>
<td>8.8</td>
<td>Composition</td>
<td>345</td>
</tr>
<tr>
<td>8.9</td>
<td>enum Types</td>
<td>348</td>
</tr>
<tr>
<td>8.10</td>
<td>Garbage Collection</td>
<td>351</td>
</tr>
<tr>
<td>8.11</td>
<td>static Class Members</td>
<td>351</td>
</tr>
<tr>
<td>8.12</td>
<td>static Import</td>
<td>355</td>
</tr>
<tr>
<td>8.13</td>
<td>final Instance Variables</td>
<td>356</td>
</tr>
<tr>
<td>8.14</td>
<td>Package Access</td>
<td>357</td>
</tr>
<tr>
<td>8.15</td>
<td>Using BigDecimal for Precise Monetary Calculations</td>
<td>358</td>
</tr>
<tr>
<td>8.16</td>
<td>(Optional) GUI and Graphics Case Study: Using Objects with Graphics</td>
<td>361</td>
</tr>
<tr>
<td>8.17</td>
<td>Wrap-Up</td>
<td>365</td>
</tr>
</tbody>
</table>

## 9 Object-Oriented Programming: Inheritance

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1</td>
<td>Introduction</td>
<td>374</td>
</tr>
<tr>
<td>9.2</td>
<td>Superclasses and Subclasses</td>
<td>375</td>
</tr>
<tr>
<td>9.3</td>
<td>protected Members</td>
<td>377</td>
</tr>
<tr>
<td>9.4</td>
<td>Relationship Between Superclasses and Subclasses</td>
<td>378</td>
</tr>
<tr>
<td>9.4.1</td>
<td>Creating and Using a CommissionEmployee Class</td>
<td>378</td>
</tr>
<tr>
<td>9.4.2</td>
<td>Creating and Using a BasePlusCommissionEmployee Class</td>
<td>383</td>
</tr>
<tr>
<td>9.4.3</td>
<td>Creating a CommissionEmployee–BasePlusCommissionEmployee Inheritance Hierarchy</td>
<td>388</td>
</tr>
<tr>
<td>9.4.4</td>
<td>CommissionEmployee–BasePlusCommissionEmployee Inheritance Hierarchy Using protected Instance Variables</td>
<td>391</td>
</tr>
<tr>
<td>9.4.5</td>
<td>CommissionEmployee–BasePlusCommissionEmployee Inheritance Hierarchy Using private Instance Variables</td>
<td>394</td>
</tr>
<tr>
<td>9.5</td>
<td>Constructors in Subclasses</td>
<td>398</td>
</tr>
</tbody>
</table>
9.6 Class Object
9.7 Designing with Composition vs. Inheritance
9.8 Wrap-Up

10 Object-Oriented Programming: Polymorphism and Interfaces

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

11 Exception Handling: A Deeper Look

11.1 Introduction
Contents

11.2 Example: Divide by Zero without Exception Handling 457
11.3 Example: Handling ArithmeticExceptions and InputMismatchExceptions 459
11.4 When to Use Exception Handling 465
11.5 Java Exception Hierarchy 465
11.6 finally Block 469
11.7 Stack Unwinding and Obtaining Information from an Exception 473
11.8 Chained Exceptions 476
11.9 Declaring New Exception Types 478
11.10 Preconditions and Postconditions 479
11.11 Assertions 479
11.12 try-with-Resources: Automatic Resource Deallocation 481
11.13 Wrap-Up 482

12 JavaFX Graphical User Interfaces: Part 1 488
12.1 Introduction 489
12.2 JavaFX Scene Builder 490
12.3 JavaFX App Window Structure 491
12.4 Welcome App—Displaying Text and an Image 492
   12.4.1 Opening Scene Builder and Creating the File Welcome.fxml 492
   12.4.2 Adding an Image to the Folder Containing Welcome.fxml 493
   12.4.3 Creating a VBox Layout Container 493
   12.4.4 Configuring the VBox Layout Container 494
   12.4.5 Adding and Configuring a Label 494
   12.4.6 Adding and Configuring an ImageView 495
   12.4.7 Previewing the Welcome GUI 497
12.5 Tip Calculator App—Introduction to Event Handling 497
   12.5.1 Test-Driving the Tip Calculator App 498
   12.5.2 Technologies Overview 499
   12.5.3 Building the App’s GUI 501
   12.5.4 TipCalculator Class 508
   12.5.5 TipCalculatorController Class 510
12.6 Features Covered in the Other JavaFX Chapters 515
12.7 Wrap-Up 515

13 JavaFX GUI: Part 2 523
13.1 Introduction 524
13.2 Laying Out Nodes in a Scene Graph 524
13.3 Painter App: RadioButtons, Mouse Events and Shapes 526
   13.3.1 Technologies Overview 526
   13.3.2 Creating the Painter.fxml File 528
   13.3.3 Building the GUI 528
   13.3.4 Painter Subclass of Application 531
   13.3.5 PainterController Class 532
## 13.4 Color Chooser App: Property Bindings and Property Listeners

13.4.1 Technologies Overview  
13.4.2 Building the GUI  
13.4.3 ColorChooser Subclass of Application  
13.4.4 ColorChooserController Class  

## 13.5 Cover Viewer App: Data-Driven GUIs with JavaFX Collections

13.5.1 Technologies Overview  
13.5.2 Adding Images to the App’s Folder  
13.5.3 Building the GUI  
13.5.4 CoverViewer Subclass of Application  
13.5.5 CoverViewerController Class  

## 13.6 Cover Viewer App: Customizing ListView Cells

13.6.1 Technologies Overview  
13.6.2 Copying the CoverViewer App  
13.6.3 ImageTextCell Custom Cell Factory Class  
13.6.4 CoverViewerController Class  

## 13.7 Additional JavaFX Capabilities  

## 13.8 JavaFX 9: Java SE 9 JavaFX Updates  

## 13.9 Wrap-Up  

## 14 Strings, Characters and Regular Expressions

14.1 Introduction  
14.2 Fundamentals of Characters and Strings  
14.3 Class String  
14.3.1 String Constructors  
14.3.2 String Methods length, charAt and getChars  
14.3.3 Comparing Strings  
14.3.4 Locating Characters and Substrings in Strings  
14.3.5 Extracting Substrings from Strings  
14.3.6 Concatenating Strings  
14.3.7 Miscellaneous String Methods  
14.3.8 String Method valueOf  
14.4 Class StringBuilder  
14.4.1 StringBuilder Constructors  
14.4.2 StringBuilder Methods length, capacity, setLength and ensureCapacity  
14.4.3 StringBuilder Methods charAt, setCharAt, getChars and reverse  
14.4.4 StringBuilder append Methods  
14.4.5 StringBuilder Insertion and Deletion Methods  
14.5 Class Character  
14.6 Tokenizing Strings  
14.7 Regular Expressions, Class Pattern and Class Matcher  
14.7.1 Replacing Substrings and Splitting Strings  
14.7.2 Classes Pattern and Matcher  
14.8 Wrap-Up  

©Copyright 2018 by Pearson Education, Inc. All Rights Reserved.
17 Lambdas and Streams  694
17.1 Introduction  695
17.2 Streams and Reduction  697
  17.2.1 Summing the Integers from 1 through 10 with a for Loop  697
  17.2.2 External Iteration with for Is Error Prone  698
  17.2.3 Summing with a Stream and Reduction  698
  17.2.4 Internal Iteration  699
17.3 Mapping and Lambdas  700
  17.3.1 Lambda Expressions  701
  17.3.2 Lambda Syntax  702
  17.3.3 Intermediate and Terminal Operations  703
17.4 Filtering  704
17.5 How Elements Move Through Stream Pipelines  706
17.6 Method References  707
  17.6.1 Creating an IntStream of Random Values  708
  17.6.2 Performing a Task on Each Stream Element with forEach and a Method Reference  708
  17.6.3 Mapping Integers to String Objects with mapToObj  709
  17.6.4 Concatenating Strings with collect  709
17.7 IntStream Operations  710
  17.7.1 Creating an IntStream and Displaying Its Values  711
  17.7.2 Terminal Operations count, min, max, sum and average  711
  17.7.3 Terminal Operation reduce  712
  17.7.4 Sorting IntStream Values  714
17.8 Functional Interfaces  715
17.9 Lambdas: A Deeper Look  716
17.10 Stream<Integer> Manipulations  717
  17.10.1 Creating a Stream<Integer>  718
  17.10.2 Sorting a Stream and Collecting the Results  719
  17.10.3 Filtering a Stream and Storing the Results for Later Use  719
  17.10.4 Filtering and Sorting a Stream and Collecting the Results  720
  17.10.5 Sorting Previously Collected Results  720
17.11 Stream<String> Manipulations  720
  17.11.1 Mapping Strings to Uppercase  721
  17.11.2 Filtering Strings Then Sorting Them in Case-Insensitive Ascending Order  722
  17.11.3 Filtering Strings Then Sorting Them in Case-Insensitive Descending Order  722
17.12 Stream<Employee> Manipulations  723
  17.12.1 Creating and Displaying a List<Employee>  724
  17.12.2 Filtering Employees with Salaries in a Specified Range  725
  17.12.3 Sorting Employees By Multiple Fields  728
  17.12.4 Mapping Employees to Unique-Last-Name Strings  730
  17.12.5 Grouping Employees By Department  731
  17.12.6 Counting the Number of Employees in Each Department  732
  17.12.7 Summing and Averaging Employee Salaries  733
## Contents

17.13 Creating a Stream<String> from a File  734  
17.14 Streams of Random Values  737  
17.15 Infinite Streams  739  
17.16 Lambda Event Handlers  741  
17.17 Additional Notes on Java SE 8 Interfaces  741  
17.18 Wrap-Up  742  

### 18 Recursion  756  
18.1 Introduction  757  
18.2 Recursion Concepts  758  
18.3 Example Using Recursion: Factorials  759  
18.4 Reimplementing Class FactorialCalculator Using BigInteger  761  
18.5 Example Using Recursion: Fibonacci Series  763  
18.6 Recursion and the Method-Call Stack  766  
18.7 Recursion vs. Iteration  767  
18.8 Towers of Hanoi  769  
18.9 Fractals  771  
18.9.1 Koch Curve Fractal  772  
18.9.2 (Optional) Case Study: Lo Feather Fractal  773  
18.9.3 (Optional) Fractal App GUI  775  
18.9.4 (Optional) FractalController Class  777  
18.10 Recursive Backtracking  782  
18.11 Wrap-Up  782  

### 19 Searching, Sorting and Big O  791  
19.1 Introduction  792  
19.2 Linear Search  793  
19.3 Big O Notation  796  
19.3.1 O(1) Algorithms  796  
19.3.2 O(n) Algorithms  796  
19.3.3 O(n²) Algorithms  796  
19.3.4 Big O of the Linear Search  797  
19.4 Binary Search  797  
19.4.1 Binary Search Implementation  798  
19.4.2 Efficiency of the Binary Search  801  
19.5 Sorting Algorithms  802  
19.6 Selection Sort  802  
19.6.1 Selection Sort Implementation  803  
19.6.2 Efficiency of the Selection Sort  805  
19.7 Insertion Sort  805  
19.7.1 Insertion Sort Implementation  806  
19.7.2 Efficiency of the Insertion Sort  808  
19.8 Merge Sort  809  
19.8.1 Merge Sort Implementation  809  
19.8.2 Efficiency of the Merge Sort  814  

©Copyright 2018 by Pearson Education, Inc. All Rights Reserved.
19.9 Big O Summary for This Chapter’s Searching and Sorting Algorithms 814
19.10 Massive Parallelism and Parallel Algorithms 815
19.11 Wrap-Up 815

20 Generic Classes and Methods: A Deeper Look 821
20.1 Introduction 822
20.2 Motivation for Generic Methods 822
20.3 Generic Methods: Implementation and Compile-Time Translation 824
20.4 Additional Compile-Time Translation Issues: Methods That Use a Type Parameter as the Return Type 827
20.5 Overloading Generic Methods 830
20.6 Generic Classes 831
20.7 Wildcards in Methods That Accept Type Parameters 838
20.8 Wrap-Up 842

21 Custom Generic Data Structures 846
21.1 Introduction 847
21.2 Self-Referential Classes 848
21.3 Dynamic Memory Allocation 848
21.4 Linked Lists 849
   21.4.1 Singly Linked Lists 849
   21.4.2 Implementing a Generic ListNode Class 850
   21.4.3 Generic Classes ListNode and List 853
   21.4.4 Class ListTest 853
   21.4.5 List Method insertAtFront 855
   21.4.6 List Method insertAtBack 856
   21.4.7 List Method removeFromFront 856
   21.4.8 List Method removeFromBack 857
   21.4.9 List Method print 858
   21.4.10 Creating Your Own Packages 858
21.5 Stacks 863
21.6 Queues 866
21.7 Trees 868
21.8 Wrap-Up 875

22 JavaFX Graphics and Multimedia 900
22.1 Introduction 901
22.2 Controlling Fonts with Cascading Style Sheets (CSS) 902
   22.2.1 CSS That Styles the GUI 902
   22.2.2 FXML That Defines the GUI—Introduction to XML Markup 905
   22.2.3 Referencing the CSS File from FXML 908
   22.2.4 Specifying the VBox’s Style Class 908
   22.2.5 Programatically Loading CSS 908

©Copyright 2018 by Pearson Education, Inc. All Rights Reserved.
Contents

22.3 Displaying Two-Dimensional Shapes 909
   22.3.1 Defining Two-Dimensional Shapes with FXML 909
   22.3.2 CSS That Styles the Two-Dimensional Shapes 912
22.4 Polyline, Polylines, Polygons and Paths
   22.4.1 GUI and CSS 915
22.5 Transforms 919
22.6 Playing Video with Media, MediaPlayer and MediaViewer
   22.6.1 VideoPlayer GUI 922
   22.6.2 VideoPlayerController Class 924
22.7 Transition Animations 928
   22.7.1 TransitionAnimations.fxml 928
22.8 Timeline Animations 934
22.9 Frame-by-Frame Animation with AnimationTimer 937
22.10 Drawing on a Canvas 939
22.11 Three-Dimensional Shapes 944
22.12 Wrap-Up 947

23 Concurrency 963
23.1 Introduction 964
23.2 Thread States and Life Cycle
   23.2.1 New and Runnable States 967
   23.2.2 Waiting State 967
   23.2.3 Timed Waiting State 967
   23.2.4 Blocked State 967
   23.2.5 Terminated State 967
   23.2.6 Operating-System View of the Runnable State 968
   23.2.7 Thread Priorities and Thread Scheduling 968
23.3 Creating and Executing Threads with the Executor Framework 969
23.4 Thread Synchronization
   23.4.1 Immutable Data 974
   23.4.2 Monitors 974
   23.4.3 Unsynchronized Mutable Data Sharing 975
   23.4.4 Synchronized Mutable Data Sharing—Making Operations Atomic 979
23.5 Producer/Consumer Relationship without Synchronization 982
23.6 Producer/Consumer Relationship: ArrayBlockingQueue 990
23.7 (Advanced) Producer/Consumer Relationship with synchronized, wait, notify and notifyAll 993
23.8 (Advanced) Producer/Consumer Relationship: Bounded Buffers 999
23.9 (Advanced) Producer/Consumer Relationship: The Lock and Condition Interfaces 1007
23.10 Concurrent Collections 1014
23.11 Multithreading in JavaFX 1016
## 24 Accessing Databases with JDBC

24.1 Introduction 1051
24.2 Relational Databases 1052
24.3 A books Database 1053
24.4 SQL
   24.4.1 Basic SELECT Query 1058
   24.4.2 WHERE Clause 1058
   24.4.3 ORDER BY Clause 1060
   24.4.4 Merging Data from Multiple Tables: INNER JOIN 1062
   24.4.5 INSERT Statement 1063
   24.4.6 UPDATE Statement 1064
   24.4.7 DELETE Statement 1065
24.5 Setting Up a Java DB Database 1066
   24.5.1 Creating the Chapter’s Databases on Windows 1067
   24.5.2 Creating the Chapter’s Databases on macOS 1068
   24.5.3 Creating the Chapter’s Databases on Linux 1068
24.6 Connecting to and Querying a Database 1068
   24.6.1 Automatic Driver Discovery 1070
   24.6.2 Connecting to the Database 1070
   24.6.3 Creating a Statement for Executing Queries 1071
   24.6.4 Executing a Query 1071
   24.6.5 Processing a Query’s ResultSet 1072
24.7 Querying the books Database 1073
   24.7.1 ResultSetTableModel Class 1073
   24.7.2 DisplayQueryResults App’s GUI 1080
   24.7.3 DisplayQueryResultsController Class 1080
24.8 RowSet Interface 1085
24.9 PreparedStatements
   24.9.1 AddressBook App That Uses PreparedStatements 1089
   24.9.2 Class Person 1089
   24.9.3 Class PersonQueries 1091
   24.9.4 AddressBook GUI 1094
   24.9.5 Class AddressBookController 1095
24.10 Stored Procedures 1100
24.11 Transaction Processing 1100
24.12 Wrap-Up 1101
# 25 Introduction to JShell: Java 9’s REPL

## 25.1 Introduction

## 25.2 Installing JDK 9

## 25.3 Introduction to JShell

- 25.3.1 Starting a JShell Session
- 25.3.2 Executing Statements
- 25.3.3 Declaring Variables Explicitly
- 25.3.4 Listing and Executing Prior Snippets
- 25.3.5 Evaluating Expressions and Declaring Variables Implicitly
- 25.3.6 Using Implicitly Declared Variables
- 25.3.7 Viewing a Variable’s Value
- 25.3.8 Resetting a JShell Session
- 25.3.9 Writing Multiline Statements
- 25.3.10 Editing Code Snippets
- 25.3.11 Exiting JShell

## 25.4 Command-Line Input in JShell

## 25.5 Declaring and Using Classes

- 25.5.1 Creating a Class in JShell
- 25.5.2 Explicitly Declaring Reference-Type Variables
- 25.5.3 Creating Objects
- 25.5.4 Manipulating Objects
- 25.5.5 Creating a Meaningful Variable Name for an Expression
- 25.5.6 Saving and Opening Code-Snippet Files

## 25.6 Discovery with JShell Auto-Completion

- 25.6.1 Auto-Completing Identifiers
- 25.6.2 Auto-Completing JShell Commands

## 25.7 Exploring a Class’s Members and Viewing Documentation

- 25.7.1 Listing Class Math’s static Members
- 25.7.2 Viewing a Method’s Parameters
- 25.7.3 Viewing a Method’s Documentation
- 25.7.4 Viewing a public Field’s Documentation
- 25.7.5 Viewing a Class’s Documentation
- 25.7.6 Viewing Method Overloads
- 25.7.7 Exploring Members of a Specific Object

## 25.8 Declaring Methods

- 25.8.1 Forward Referencing an Undeclared Method—Declaring Method displayCubes
- 25.8.2 Declaring a Previously Undeclared Method
- 25.8.3 Testing cube and Replacing Its Declaration
- 25.8.4 Testing Updated Method cube and Method displayCubes

## 25.9 Exceptions

## 25.10 Importing Classes and Adding Packages to the CLASSPATH

## 25.11 Using an External Editor

## 25.12 Summary of JShell Commands

- 25.12.1 Getting Help in JShell
- 25.12.2 /edit Command: Additional Features
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