Contents

Preface xi

Notational Conventions xii

1 Running Java: Compilation, Loading, and Execution 2

2 Names and Reserved Names 2

3 Java Naming Conventions 2

4 Comments and Program Layout 2

5 Types 4
5.1 Primitive Types . . . 4
5.2 Reference Types . . . 4
5.3 Array Types . . . 4
5.4 Boxing: Wrapping Primitive Types as Reference Types . . . 4
5.5 Subtypes and Compatibility . . . 6
5.6 Signatures and Subsumption . . . 6
5.7 Type Conversion . . . 6

6 Variables, Parameters, Fields, and Scope 8
6.1 Values Bound to Variables, Parameters, or Fields . . . 8
6.2 Variable Declarations . . . 8
6.3 Scope of Variables, Parameters, and Fields . . . 8

7 Strings 10
7.1 String Formatting . . . 12

8 Arrays 16
8.1 Array Creation and Access . . . 16
8.2 Array Initializers . . . 16
8.3 Multidimensional Arrays . . . 18
8.4 The Utility Class Arrays . . . 18

9 Classes 22
9.1 Class Declarations and Class Bodies . . . 22
9.2 Top-Level Classes, Nested Classes, Member Classes, and Local Classes . . . 22
9.3 Class Modifiers . . . 22
9.4 The Class Modifiers `public`, `final`, `abstract` . . . 24
9.5 Subclasses, Superclasses, Class Hierarchy, Inheritance, and Overriding . . . 24
9.6 Field Declarations in Classes . . . 26
9.7 The Member Access Modifiers `private`, `protected`, `public` . . . 26
9.8 Method Declarations . . . 28
9.9 Parameter Arrays and Variable-Arity Methods . . . 30

9.10 Constructor Declarations 30
9.11 Nested Classes, Member Classes, Local Classes, and Inner Classes 32
9.12 Anonymous Classes 32
9.13 Initializer Blocks, Field Initializers, and Initializers 32

10 Classes and Objects in the Computer **34**
10.1 What Is a Class? 34
10.2 What Is an Object? 34
10.3 Inner Objects 34

11 Expressions **36**
11.1 Table of Expression Forms 36
11.2 Arithmetic Operators 38
11.3 Logical Operators 38
11.4 Bitwise Operators and Shift Operators 38
11.5 Assignment Expressions 40
11.6 Conditional Expressions 40
11.7 Object Creation Expressions 40
11.8 Instance Test Expressions 40
11.9 Field Access Expressions 42
11.10 The Current Object Reference `this` 42
11.11 Type Cast Expression 42
11.12 Method Call Expressions 44
11.13 Lambda Expressions (Java 8.0) 48
11.14 Method Reference Expressions (Java 8.0) 48

12 Statements **52**
12.1 Expression Statements 52
12.2 Block Statements 52
12.3 The Empty Statement 52
12.4 Choice Statements 54
12.5 Loop Statements 56
12.6 Returns, Labeled Statements, Exits, and Exceptions 60
12.7 The Try-with-Resources Statement 64
12.8 The `assert` Statement 64

13 Interfaces **66**
13.1 Interface Declarations 66
13.2 Classes Implementing Interfaces 66
13.3 Default and Static Methods on Interfaces (Java 8.0) 68
13.4 Annotation Type Declarations 68

14 Enum Types **70**

15 Exceptions, Checked and Unchecked **72**

16 Compilation, Source Files, Class Names, and Class Files **74**

17 Packages and Jar Files **74**

18 Mathematical Functions **76**

19 String Builders and String Buffers **78**

20 Threads, Concurrent Execution, and Synchronization **80**
20.1 Threads and Concurrent Execution 80
20.2 Locks and the `synchronized` Statement 82
20.3 Operations on Threads 84
20.4 Operations on Locked Objects 84
20.5 The Java Memory Model and Visibility Across Threads 86

21 Generic Types and Methods **88**
21.1 Generics: Safety, Generality, and Efficiency 88
21.2 Generic Types, Type Parameters, and Type Instances 88
21.3 How Can Type Instances Be Used? 88
21.4 Generic Classes 90
21.5 Constraints on Type Parameters 92
21.6 How Can Type Parameters Be Used? 92
21.7 Generic Interfaces 94
21.8 Generic Methods 96
21.9 Wildcard Type Arguments 98
21.10 The Raw Type 100
21.11 The Implementation of Generic Types and Methods 100

22 Generic Collections and Maps **102**
22.1 Interface Collection<T> 104
22.2 Interface List<T> and Its Implementations LinkedList<T> and ArrayList<T> 105
22.3 Interface Set<T> and Its Implementations HashSet<T> and LinkedHashSet<T> 106
22.4 Interface SortedSet<T> and Implementation TreeSet<T> 106
22.5 Interface Map<K,V> and Implementation HashMap<K,V> 108
22.6 Interface SortedMap<K,V> and Implementation TreeMap<K,V> 110
22.7 Going through a Collection: Interfaces Iterator<T> and Iterable<T> 112
22.8 Interface ListIterator<T> 114
22.9 Equality, Hash Codes, and Comparison 114
22.10 The Comparator<T> Interface 116
22.11 The Utility Class Collections 118
22.12 Choosing the Right Collection Class or Map Class 120

23 Functional Interfaces (Java 8.0) **122**
23.1 Functional Programming 122
23.2 Generic Functional Interfaces 124
23.3 Primitive-Type Specialized Functional Interfaces 124
23.4 Covariance and Contravariance in Functional Interfaces 126
23.5 Interface Function<T,R> 126
23.6 Interface UnaryOperator<T> 126

23.7 Interfaces Predicate<T> and BiPredicate<T,U> 128
23.8 Interfaces Consumer<T> and BiConsumer<T,U> 128
23.9 Interface Supplier<T> 128
23.10 Interface BiFunction<T,U,R> 130
23.11 Interface BinaryOperator<T> 130

24 Streams for Bulk Data (Java 8.0) **132**
24.1 Creating Streams 134
24.2 Stream Builders 134
24.3 Methods on Streams 136
24.4 Numeric Streams: DoubleStream, IntStream, and LongStream 140
24.5 Summary Statistics for Numeric Streams 140
24.6 Collectors on Streams 142

25 Class Optional<T> (Java 8.0) **146**

26 Input and Output **148**
26.1 Creating an IO Stream from Another One 149
26.2 Kinds of Input and Output Methods 150
26.3 Imports, Exceptions, Thread Safety 150
26.4 Sequential Character Input: Readers 152
26.5 Sequential Character Output: Writers 153
26.6 Printing Primitive Data to a Character Stream: PrintWriter 154
26.7 The Appendable Interface and the CharSequence Interface 154
26.8 Reading Primitive Data from a Character Stream: StreamTokenizer 156
26.9 Sequential Byte Input: InputStream 158
26.10 Sequential Byte Output: OutputStream 159
26.11 Binary Input-Output of Primitive Data: DataInput and DataOutput 160
26.12 Serialization of Objects: ObjectInput and ObjectOutput 162
26.13 Buffered Input and Output 164
26.14 Random Access Files: RandomAccessFile 166
26.15 Files, Directories, and File Descriptors 168
26.16 Thread Communication: PipedInputStream and PipedOutputStream 168
26.17 Socket Communication 170

27 Reflection **172**
27.1 Reflective Use of Types: The Class<T> Class 172
27.2 Reflection: The Field Class 174
27.3 Reflection: The Method Class and the Constructor<T> Class 174
27.4 Exceptions Thrown When Using Reflection 174

28 Metadata Annotations **176**

29 What Is New in Java 8.0 **178**

References **180**

Index **181**