CONTENTS

List of Programs xxv
List of Figures xxix
List of Tables xxxiii
Preface to second version xxxv

PART 1 Structured Programming 1

1 Getting started 3

1.1 Programming 3
1.2 Editing source code 5
Source code file naming rules 5
1.3 Development tools for Java 6
1.4 Compiling Java programs 6
1.5 Running Java programs 7
1.6 The components of a program 8
Operations 8
Programming with objects 8
Object-based programming 9
1.7 The Java programming language 9
Classes and methods 9
Comments and indentation 10
Program entry point 11
Method calls 12
Variables 12
1.8 The Java Virtual Machine 13
1.9 Review questions 15
1.10 Programming exercises 16

2 Basic programming elements 17

2.1 Printing to the terminal window 17
The `print()` and `println()` methods 17
Creating program output using strings 19
2.2 Local variables 20
Declaring variables 20
Assigning variables 21
Logical errors 22
Literals and constants 24
Choosing names 24

2.3 Numerical data types 25
Primitive data type `int` 26
Primitive data type `double` 26
2.4 Arithmetic expressions and operators 26
Arithmetic expression evaluation rules 26
Conversion between primitive data types 27
Precedence and associativity rules 28
Integer and floating-point division 29
2.5 Formatted output 31
Format string 31
Sample format specifications 31
Printing with fixed field widths 31
2.6 Reading numbers from the keyboard 34
The `Scanner` class 34
Reading integers 35
Reading floating-point numbers 36
Error handling 37
Reading multiple values per line 37
Skipping the rest of the line when reading values from the keyboard 38
2.7 Review questions 39
2.8 Programming exercises 41

3 Program control flow 45
3.1 Boolean expressions 46
Boolean primitive data type 46
Relational operators 46
Understanding relational operators 47
Logical operators 48
Short-circuit evaluation 49
De Morgan's laws 50
Using Boolean expressions to control flow of execution 50
3.2 Control flow: selection statements 51
Simple selection statement: `if` 51
Blocks of statements: `{ ... }` 52
Local variables in a block 53
Selection statement with two choices: `if-else` 54
Nested selection statements 56
3.3 Control flow: loops 59
Loop test before loop body: `while` 59
Loop test after loop body: `do-while` 60
Infinite loops 61
Using loops to implement user dialogue 61
Choosing the right loop 62
Nested loops 63

3.4 Assertions 64
Making assertions 64
Assertions as a testing technique 65
3.5 Review questions 66
3.6 Programming exercises 68

PART 2 Object-Based Programming 73

4 Using objects 75
4.1 Introduction to the object model 75
Abstractions, classes and objects 76
Objects, reference values and reference variables 77
The `new` operator 78
Using objects 78
Object state 80
4.2 Strings 81
Characters and strings 81
String concatenation 83
Creating string objects 85
String comparison 85
Converting primitive values to strings 86
Other useful methods for strings 88
4.3 Manipulating references 89
Reference types and variables 89
Assignment to reference variables 89
Aliases: one object, several references 89
The `null` literal 90
Comparing objects 90
4.4 Primitive values as objects 92
Auto-boxing 92
Auto-unboxing 92
Explicit boxing and unboxing 93
Useful methods in the wrapper classes 93
4.5 Review questions 95
4.6 Programming exercises 97

5 More on control structures 99
5.1 The extended assignment operators 100
5.2 The increment and decrement operators 101

5.3 Counter-controlled loops 102
Local variables in the `for(;;)` loop 105
Other increments in `for(;;)` loops 106
Counting backwards with `for(;;)` loops 106
Nested `for(;;)` loops 106
5.4 Changing control flow in loops 108
The `break` statement 108
The `continue` statement 108
5.5 Common pitfalls in loops 110
Infinite loop: `for(;;)` 110
One-off errors 110
Errors in initialization 110
Errors in the loop condition 111
Optimizing loops 111
5.6 Multiple-selection statements 112
The `default` label 113
The `case` label values 113
Falling through case labels 115
5.7 Review questions 117
5.8 Programming exercises 120

6 Arrays 123
6.1 Arrays as data structures 124
6.2 Creating and using arrays 125
Declaring array reference variables 125
Creating arrays 125
Default initialization 125
Arrays of objects 126
The `length` field 127
Accessing an array element 127
Array bounds 128
Array aliases 128
Alternate notation for array declaration 128
6.3 Initializing arrays 129
6.4 Iterating over an array 131
Comparing all values in an array with a given value 132
Finding the lowest value in an array 132
Finding the highest value in an array 133
Adding the values in an array 133
Iterating over an array of objects 135
6.5 Multidimensional arrays 136
Creation and initialization of multidimensional arrays 136
Printing a two-dimensional array in tabular form 139
Iterating over a specific row in a two-dimensional array 139
Iterating over a specific column in a two-dimensional array 139

Iterating over all the columns in a two-dimensional array 139
Ragged arrays 141
Arrays with more than two dimensions 143

6.6 More on iteration: enhanced `for` loop 143

6.7 More miscellaneous operations on arrays 145

Copying arrays 145
Comparing arrays 146
Working with partially-filled arrays 147

6.8 Pseudo-random number generator 149

The `Random` class 149
Determining the range 149
Simulating a dice roll 150
Generating the same sequence of pseudo-random numbers 151

6.9 Review questions 152

6.10 Programming exercises 156

7 Defining classes 161

7.1 Class members 162

7.2 Defining object properties using field variables 163

Field declarations 163
Initializing fields 163

7.3 Defining behaviour using instance methods 164

Method declaration and formal parameters 164
Method calls and actual parameter expressions 167
Parameter passing: call-by-value 168
The current object: `this` 173
Method execution and the `return` statement 174
Passing information using arrays 177
Automatic garbage collection 179

7.4 Static members of a class 179

Accessing static members 182
Initializing static variables 183
The `main()` method and program arguments 184

7.5 Initializing object state 186

Default constructors: implicit or explicit 186
Constructors with parameters 187
Overloading constructors 189

7.6 Enumerated types 192

Simple form of enumerated types 192
Selected methods for enumerated types 192
General form of enumerated types 194
Declaring enumerated types inside a class 196

7.7 Review questions 197

7.8 Programming exercises 204

8 Object communication 207

8.1 Responsibilities and roles 207

A naive solution to a given problem 208
Combining properties and behaviour in classes 209
Using references as parameters 211
Advantages of good abstractions 212
Rethinking responsibility 212

8.2 Communication and cooperation 213

Dividing and assigning responsibility 213
Clarifying the responsibilities of classes in the program 216
Communication between objects at runtime 217
Assigning the responsibility for calculating salaries 218

8.3 Relationships between objects 218

Fields with reference values 218
Objects that communicate 219
Object ownership 220
One-to-one and one-to-many associations 221
Suggestions for further development 223

8.4 Method overloading 223

8.5 Documenting source code 224

Multi-line comments 225
Documenting classes and members 225
Hiding internal methods and fields 226
A fully-documented Java class 226
How to document programs 228

8.6 Review questions 230

8.7 Programming exercises 231

PART 3 Useful Techniques for Building Programs 233

9 Sorting and searching arrays 235

9.1 Natural order 235

Relational operators for primitive data types 236
Understanding relational operators 236

9.2 Selection sort 238

Sorting an array of integers 238
Pseudocode for the selection sort 239
Coding the selection sort 240
Analysing the amount of work required by a selection sort 242

9.3 Insertion sort 242

Sorting an array of integers using an insertion sort 242
Pseudocode for the insertion sort 242
Coding the insertion sort 243
Analysing the amount of work required by an insertion sort 244

9.4 Sorting arrays of objects 245
Comparing objects: the `Comparable` interface 245
Implementing the natural order for time objects 246
Sorting arrays of comparable objects 249
9.5 Linear search 251
9.6 Binary search 253
9.7 Sorting and searching a CD collection 256
9.8 Sorting and searching using the Java standard library 259
9.9 Review questions 261
9.10 Programming exercises 263

10 Exception handling 269

10.1 What is an exception? 269
10.2 Method execution and exception propagation 270
Method execution 270
Stack trace 272
Exception propagation 273
10.3 Exception handling 274
`try`–`catch` scenario 1: no exception 276
`try`–`catch` scenario 2: exception handling 277
`try`–`catch` scenario 3: exception propagation 278
10.4 Checked exceptions 281
Dealing with checked exceptions using the `throws` clause 282
Programming with checked exceptions 282
10.5 Unchecked exceptions 284
10.6 Review questions 286
10.7 Programming exercises 287

11 Text file I/O and simple GUI dialogs 289

11.1 File handling 290
Data records 291
11.2 Text files 294
Writing to text files 294
Reading from text files 300
11.3 Simple GUI dialog design 304
Overview of the `JOptionPane` class 304
Message dialogs – presenting information to the user 307
Input dialogs – reading data from the user 308
Confirmation dialogs – getting confirmation from the user 309
11.4 Review questions 312
11.5 Programming exercises 316

PART 4 Object-Oriented Programming (OOP) 321

12 Inheritance 323

12.1 Main concepts 324

Superclasses and subclasses 324
Specialization and generalization 324
Inheritance in Java 325

12.2 Extending properties and behaviour 326

Initializing object state 328
Extending behaviour 329

12.3 Using inherited members 332

Accessing inherited members in the subclass 332
Accessing inherited members using subclass references 333
Using superclass references to refer to subclass objects 334
Conversion between reference types 335
Using superclass and subclass references 336

12.4 Extending behaviour in the subclass 338

Overriding instance methods in the subclass 338
Using an overridden method from the superclass 340
Using a shadowed field 342

12.5 Final classes and methods 343

Final classes 343
Final methods 343
Overloading methods 344

12.6 Review questions 344

12.7 Programming exercises 348

13 Polymorphism and interfaces 353

13.1 Programming with inheritance 354

A superclass with multiple subclasses 354
Polymorphism and polymorphic references 357
Arrays of polymorphic references 359
Consequences of polymorphism 361
Inheritance is transitive 361

13.2 Abstract classes 362

Abstract and concrete classes 362
Implementing abstract methods in subclasses 363
Abstract classes in the Java standard library 365

13.3 Aggregation 366

13.4 Interfaces in Java 367

The concept of contracts in programs 367
Declaring an interface 368
Implementing an interface 369

Implementing multiple interfaces 370
Inheritance between interfaces 371
Polymorphic use of interface references 371
13.5 Review questions 374
13.6 Programming exercises 376

PART 5 Applying OOP 385

14 Test-driven program development 387
14.1 Developing large programs 387
14.2 Simple testing with `assert` 388
The game board 389
Game board state 391
The state of several cells 393
Stacking pieces 396
Better error messages when tests fail 397
14.3 Testing framework 398
JUnit framework 399
Writing test methods 399
Running tests with JUnit 403
Fixing test failures with JUnit 404
Full columns 405
14.4 Printing the game board 407
Simulated game play 407
Method for printing game board 410
14.5 Refactoring program code 412
14.6 Interactive Four-in-a-Row game 414
Functional decomposition 414
Dropping the next piece (steps 3a and 3b) 416
A simplified end of game test (step 2) 416
14.7 Ending a game 419
No winners 419
Four pieces in a row 420
14.8 Machine-controlled player 425
The interface `IPlayer` and implementations 425
A game between arbitrary players 427
14.9 Class libraries 428
Packages and the import statement 429
Static import 429
Declaring packages 430
Designing frameworks 431

14.10 Encapsulating implementation details 433
Access modifiers for package members 434
Access modifiers for class members 435
Application programming interface (API) 439
Limiting access to fields 440
Tell – don't ask 440
14.11 Base classes 441
Base classes in the `game` package 441
The package `game.terminal` 443
14.12 Compiling and running code in packages 445
The package `game.test` 447
14.13 Review questions 451
14.14 Programming exercises 452

15 Using dynamic data structures 455
15.1 Overview 456
Abstract data types 456
Organizing and manipulating data 456
15.2 Character sequences: `StringBuilder` 457
Creating a character sequence using a `StringBuilder` 457
Modifying the contents of a `StringBuilder` 458
Other classes for handling character sequences 459
15.3 Introduction to generic types 461
Declaring generic classes 462
Using generic classes 463
Generic interfaces 465
Handling of generic types at compile time 466
15.4 Collections 467
Superinterface `Collection<E>` 467
Traversing a collection using an iterator 469
Traversing a collection using a `for(:)` loop 470
Default string representation of a collection 471
15.5 Lists: `ArrayList<E>` 471
Subinterface `List<E>` 471
Using lists 472
15.6 Sets: `HashSet<E>` 474
Interface `Set<E>` 474
Using sets 474
15.7 Maps: `HashMap<K, V>` 477
Hashing 478
The `Map<K, V>` interface 481
Map views 483
Using maps 484

15.8 More on generic types 486
Specifying subtypes: `<? extends T>` 487
Specifying supertypes: `<? super T>` 487
Specifying any type: `<?>` 488
Generic methods 488
15.9 Sorting and searching methods from the Java APIs 489
15.10 Review questions 492
15.11 Programming exercises 495

16 Implementing dynamic data structures 499
16.1 Simple linked lists 499
Manipulating references in a linked list 501
Operations on linked lists 502
Inserting at the head of a linked list 508
Inserting at the tail of a linked list 509
Removing from the head of a linked list 510
Removing from the tail of a linked list 512
Removing a node from inside a linked list 514
Iterator for a linked list 516
Converting to an array 517
Remarks on linked lists 517
16.2 Stacks and queues 517
Implementing a stack 517
Using stacks 520
Queues 523
Using queues 524
16.3 Review questions 526
16.4 Programming exercises 526

17 Recursion 531
17.1 Recursion and iteration 532
Factorials: an example from mathematics 532
Using recursive method calls 533
17.2 Designing recursive algorithms 534
17.3 Infinite recursion 535
17.4 Recursive binary search 536
17.5 Towers of Hanoi 539
Recursive solution 540
Iterative solution 543
17.6 Quicksort: a recursive sorting algorithm 544
17.7 Fibonacci series: an example of iterative solution 548
17.8 Review questions 551
17.9 Programming exercises 552

18 More on exception handling 555

18.1 Exception classes 555

The `Exception` class 557
The `RuntimeException` class 557
The `Error` class 557

18.2 Throwing an exception programmatically 557

18.3 Handling several types of exceptions 559

Typical programming errors in exception handling 561

18.4 Defining new exceptions 562

18.5 Using the `finally` block 564

18.6 Review questions 565

18.7 Programming exercises 567

19 Files and streams 569

19.1 Streams 570

Overview of byte streams 570
Overview of character streams 571

19.2 Terminal window I/O 572

Writing to the terminal window 573
Reading from the keyboard 573

19.3 Binary files 576

File path 576
Writing to binary files 577
Reading binary values 582

19.4 Object serialization 586

Writing objects 587
Reading objects 594
Effective storing of objects 595

19.5 Random access files 598

Overview of the class `RandomAccessFile` 598
Setting up random access 600
The file pointer 600
Reading and writing random access files 601

19.6 Review questions 610

19.7 Programming exercises 613

20 Graphical user interfaces 615

20.1 Building GUIs 616

Designing GUI-based dialogues 616
Delegating events 616
GUI building packages 617
Basics of GUI development 617

20.2 Components and containers 617
Components 618
Containers 620
Windows and frames 621
Panels 622
Steps in developing a GUI 622
GUI and the `main()` method 625
20.3 GUI control components 626
Text fields 626
Buttons 627
GUI design with panels 628
Using different types of buttons 631
20.4 Designing layout 635
`FlowLayout` 635
`BorderLayout` 636
`GridLayout` 639
20.5 Event-driven programming 644
Events 644
Event delegation model 644
Classes `ActionEvent` and `WindowEvent` 646
Programming paradigms for event handling 647
20.6 Dialog boxes 661
Application using dialog boxes 662
Implementing a dialog box 666
20.7 Anonymous classes as listeners 669
Creating and registering anonymous listener objects 669
Using anonymous listener objects 670
Some remarks on anonymous classes 672
20.8 Programming model for GUI-based applications 673
20.9 GUI for the Four-in-a-Row game 674
20.10 Review questions 680
20.11 Programming exercises 688

PART 6 Appendices 691

A Answers to review questions 693
A.1 Getting started 693
A.2 Basic programming elements 693
A.3 Program control flow 696
A.4 Using objects 697
A.5 More on control structures 699
A.6 Arrays 700

A.7 Defining classes 702
A.8 Object communication 706
A.9 Sorting and searching arrays 707
A.10 Exception handling 708
A.11 Text file I/O and simple GUI dialogs 709
A.12 Inheritance 710
A.13 Polymorphism and interfaces 713
A.14 Test-driven program development 714
A.15 Using dynamic data structures 715
A.16 Implementing dynamic data structures 716
A.17 Recursion 717
A.18 More on exception handling 717
A.19 Files and streams 718
A.20 Graphical user interfaces 720

B Language reference 725

B.1 Reserved words 725
B.2 Operators 726
B.3 Primitive data types 728
B.4 Java modifiers 729

C Formatted values 731

C.1 Syntax for format strings 731
C.2 Conversion codes and types 732
C.3 Examples 735

D The Unicode character set 737

D.1 Excerpt from the Unicode character set 737
D.2 Lexicographical order and alphabetical order 739

E Console I/O and simple GUI dialog boxes 741

E.1 Support for console I/O 741
E.2 Support for simple GUI dialog boxes 744

F Numeral systems and representation 749

F.1 Numeral systems 749
Decimal numeral system 749
Binary numeral system 750
Octal numeral system 751
Hexadecimal numeral system 751

F.2 Conversions between numeral systems 752
F.3 Conversions from decimal numeral system 753
F.4 Integer representation 754
F.5 String representation of integers 756

G Programming tools in the JDK 759

G.1 Programming tools 759
G.2 Commands 759
Compiling source code: `javac` 759
Running the program: `java` 760
Generating documentation: `javadoc` 761
G.3 Configuring the `CLASSPATH` 761

H Introduction to UML 763

H.1 Class diagram 763
H.2 Object diagrams 765
H.3 Sequence diagrams 766
H.4 Activity diagrams 766

Index 769