

---

# Developing Java Software

---

Russel Winder

*King's College London, UK*

and

Graham Roberts

*University College London, UK*

**JOHN WILEY & SONS**

Chichester • New York • Weinheim • Brisbane • Singapore • Toronto

---

# Contents

---

<b>Contents</b>	<b>v</b>
<b>Preface</b>	<b>xvii</b>
<b>Part 1 Programming with Objects and Classes</b>	
<b>1 Introduction</b>	<b>3</b>
1.1 The Start	3
1.2 A (Very!) Short History of Java	3
1.3 Being at the Right Place at the Right Time	5
1.4 What is Java?	5
1.5 The JDK	7
1.6 Applications and Applets	8
1.7 Java is Architecture Neutral	8
1.8 Java is No Joke	10
1.9 Summary	10
<b>2 Programming Fundamentals</b>	<b>11</b>
2.1 Introduction	11
2.2 Imperative Programming	12
2.2.1 Controlling Execution	15
2.2.2 State and Variables	15
2.2.3 Assignment	17
2.2.4 Variable Types and Type Checking	17
2.3 Moving On — Giving a Program Structure	18
2.3.1 Abstraction & Encapsulation	20
2.3.2 Procedural Abstraction	20
2.3.3 Procedure Arguments	22
2.3.4 Procedural Decomposition and Structured Programming	23
2.4 Organizing Variables	24

2.4.1	Scope and Lifetime	25
2.4.2	Names and Scope	26
2.4.3	Initialization	26
2.5	Comments and Documentation	27
2.6	Summary	28
<b>3</b>	<b>Introducing Objects and Classes</b>	<b>29</b>
3.1	Introduction	29
3.2	A Stack Object	30
3.3	A Stack Class	31
3.4	Classes and Types	32
3.5	Using a Stack Object	33
3.6	Message Passing	34
3.7	Classes and Encapsulation	35
3.8	Programming with Classes	35
3.9	Summary	36
<b>4</b>	<b>Class Relationships</b>	<b>37</b>
4.1	Introduction	37
4.2	Association	37
4.3	Inheritance	38
4.4	An Inheritance Example	39
4.5	Choosing Between Inheritance and Association	40
4.6	Inheritance Hierarchies and Abstract Classes	42
4.7	Using an Abstract Class	43
4.8	Interfaces	45
4.9	Reusability and Components	47
4.10	A Small Example of Using Inheritance	48
4.11	Summary	49
<b>5</b>	<b>Java, Classes and Objects</b>	<b>51</b>
5.1	Introduction	51
5.2	Objects and Variables	51
5.3	Objects and Method Parameters	52
5.4	Overloading Methods	54
5.5	Creating and Initializing New Objects	55
5.5.1	Constructors	55
5.5.2	Instance Variable Initialization	57
5.5.3	Instance Initializers	57
5.5.4	Choosing Between the Alternatives	57
5.6	Removing Old Objects	58
5.7	The Java Class Libraries	59
5.8	The Library Class Object	61
5.9	Copying and Comparing Objects	61

5.10	Arrays	62
5.11	Strings	63
5.12	Container Classes	64
5.13	Static Variables and Methods	65
5.14	Files and Packages	68
5.15	Summary	68
<b>6</b>	<b>Introducing Threads and Exceptions</b>	<b>71</b>
6.1	Threads — Doing More Than One Thing At Once	71
6.1.1	Using Threads	73
6.1.2	Thread Synchronization	73
6.1.3	Thread Management	74
6.2	Exceptions	75
6.2.1	Catching Exceptions	76
6.2.2	Throwing an Exception	77
6.2.3	Plan to Use Exceptions	78
6.3	Summary	78
<b>7</b>	<b>A Complete Program</b>	<b>81</b>
7.1	A Complete Java Program	81
7.2	Some Thoughts on this Program	83
7.3	Summary	84
<b>8</b>	<b>The Programming Process</b>	<b>85</b>
8.1	Introduction	85
8.2	Object-Oriented Software Engineering	85
8.3	Why Object-Oriented?	86
8.3.1	Route Planning	87
8.3.2	Problem Domains	87
8.4	Writing Programs	88
8.4.1	An Overview	89
8.4.2	User Requirements Gathering	90
8.4.3	Analysis	92
8.4.4	Design, Implementation and Testing	97
8.4.5	Review and Iterate	98
8.4.6	Program Testing and Delivery	99
8.4.7	Debugging	100
8.5	Maintenance	101
8.6	Design Patterns	102
8.7	Summary	103
<b>9</b>	<b>A Program Design Example</b>	<b>105</b>
9.1	Introduction	105
9.2	Requirements and Scenarios	105

9.2.1 Scenario 1 — Opening a New Account	106
9.2.2 Scenario 2 — Closing an Account	107
9.2.3 Scenario 3 — Check Balance of Account	107
9.2.4 Scenario 4 — Deposit Money in an Account	107
9.2.5 Scenario 5 — Withdraw Money from an Account	108
9.2.6 Scenario Results	108
9.3 Object-Oriented Analysis (OOA)	108
9.3.1 Looking for Classes	109
9.3.2 Revised Classes	112
9.3.3 Using CRC	114
9.4 Designing the Program — First Iteration	115
9.4.1 Class Bank	115
9.4.2 Class Account	115
9.4.3 Account Subclasses	116
9.4.4 Class Customer	117
9.4.5 First Review	117
9.5 Designing the Program — Second Iteration	118
9.5.1 Second Design of the Bank Class	118
9.5.2 Second Design of the Account Classes	121
9.6 The Main Method	121
9.7 Testing	122
9.8 A Critique	122
9.9 Summary	123

## **Part 2 The Java Programming Language in Detail**

<b>10 A Java Language Reference</b>	<b>127</b>
10.1 Introduction	127
10.2 Syntax and Semantics	127
10.3 The Presentation	128
10.4 The Example Programs	129
10.5 Running the Example Programs	130
10.6 Summary	131
<b>11 Variables, Types and Expressions</b>	<b>133</b>
11.1 Introduction	133
11.2 Comments	133
11.3 Identifiers	138
11.4 Unicode Escapes	140
11.5 Literals	140
11.6 Types	142
11.6.1 Primitive Types	142
11.6.2 Reference Types	144

11.6.3 Automatic Type Conversion	144
11.7 Scope	145
11.8 Variables	147
11.8.1 Declaring Variables	147
11.8.2 Variable Initialization	149
11.8.3 Array Variables	151
11.9 Expressions and Operators	155
11.9.1 Primary Expressions	155
11.9.2 Operators	157
11.10 Source Files	170
11.11 Summary	170
<b>12 Flow Control</b>	<b>171</b>
12.1 Introduction	171
12.2 Selection	171
12.2.1 The If Statement	172
12.2.2 Switch Statement	175
12.3 Iteration	180
12.3.1 The While Loop	181
12.3.2 The Do Loop	184
12.3.3 The For Loop	186
12.3.4 Checking Loop Behaviour	189
12.4 Transfer Statements	191
12.4.1 Break	191
12.4.2 Continue	195
12.4.3 Return	197
12.5 Recursion	198
12.6 Summary	200
<b>13 Classes and Packages</b>	<b>201</b>
13.1 Introduction	201
13.2 Classes	201
13.2.1 The Class Declaration	201
13.2.2 Public, Private and Protected	204
13.2.3 Instance Variables	205
13.2.4 Static or Class Variables	206
13.2.5 Methods	208
13.2.6 Static or Class Methods	213
13.2.7 Constructors	215
13.2.8 Static Initializers	218
13.2.9 This	220
13.3 Top-Level Classes	223
13.4 Nested Classes	223
13.4.1 Nested Top-Level Classes	223

13.4.2	Member Classes	226
13.4.3	Local Classes	231
13.4.4	Anonymous Classes	235
13.5	Packages	238
13.6	Summary	241
<b>14</b>	<b>Inheritance and Interfaces</b>	<b>243</b>
14.1	Introduction	243
14.2	Inheritance	243
14.2.1	Basic Inheritance	244
14.2.2	Private and Protected Keywords and Inheritance	246
14.2.3	Final Classes	249
14.2.4	Method Overriding	249
14.2.5	Final Methods	253
14.2.6	Method Call Expressions	254
14.2.7	Modified Name Lookup for Member Classes	262
14.2.8	Constructors and Inheritance	264
14.2.9	Super and Hidden Names	266
14.2.10	Abstract Classes	268
14.2.11	Abstract Methods	270
14.2.12	Methods Inherited from Class Object	272
14.3	Interfaces	277
14.3.1	Interface Declaration	279
14.3.2	Implements	282
14.4	Summary	285
<b>15</b>	<b>Exception Handling</b>	<b>287</b>
15.1	Introduction	287
15.1.1	Exception Classes	287
15.1.2	Try, Catch and Finally	290
15.1.3	Exception Propagation	295
15.1.4	Throws Declarations	295
15.1.5	Throw	299
15.2	Summary	301
<b>16</b>	<b>Threads and Concurrency</b>	<b>303</b>
16.1	Introduction	303
16.2	Class Thread	305
16.3	Synchronized Methods	312
16.4	Synchronized Statement	318
16.5	Summary	319

## Part 3 Building Class Libraries

<b>17 Building Libraries</b>	<b>323</b>
17.1 Introduction	323
17.2 Introducing The Package Architecture	326
17.3 Avoiding Assumptions	328
17.4 What's in a Name?	329
17.5 How to Iterate?	330
17.6 Making Assumptions	330
17.7 Order Relations	332
17.7.1 The First Implementation Approach	333
17.7.2 An Alternative Implementation Approach	339
17.7.3 A Third Variant Implementation	343
17.7.4 Which Design is Best?	347
17.8 The Biggest Question of All	349
17.9 More Exceptional Material	350
17.10 Documentation and Style	352
17.11 Summary	353
<b>18 List and Sequence</b>	<b>355</b>
18.1 Introduction	355
18.2 Linked Lists	356
18.3 A List ADT	357
18.4 Abstracting Storage Strategies	371
18.5 But Which to Use?	375
18.6 Summary	377
<b>19 Queue</b>	<b>379</b>
19.1 Introduction	379
19.2 Using the Abstraction by Conformance Architecture	380
19.2.1 Vector Implementation	381
19.2.2 List Implementation	385
19.2.3 Circular Array Implementation	388
19.2.4 Linked List Implementation	395
19.2.5 Circular List Implementation	400
19.2.6 An Example	406
19.3 Using the Abstraction by Inheritance Architecture	408
19.4 Using the Abstraction by Association Architecture	410
19.5 Performance Criteria	414
19.6 A Typesafe Queue	414
19.7 Summary	416
<b>20 Dequeue</b>	<b>419</b>
20.1 Introduction	419
20.2 A Small Digression	420
20.3 A Dequeue by Inheritance from Queue	421



20.4	A Doubly Linked List Open Container	424
20.5	Another (Better!) Doubly Linked List	431
20.6	Summary	441
<b>21</b>	<b>Stack</b>	<b>443</b>
21.1	Introduction	443
21.2	Vector Implementation of Stack	447
21.3	An Example of Using a Stack	451
21.4	Summary	454
<b>22</b>	<b>Tree</b>	<b>455</b>
22.1	Introduction	455
22.2	Manipulating Trees	458
22.3	Ordered Binary Tree	481
22.4	AVL Tree	491
22.5	Summary	504
<b>23</b>	<b>Heap</b>	<b>505</b>
23.1	Introduction	505
23.2	Inserting Data	508
23.3	Removing Data	509
23.4	The Declaration of the Class	510
23.5	A Note on Using Heaps	519
23.6	Summary	521
<b>24</b>	<b>Priority Queue</b>	<b>523</b>
24.1	Introduction	523
24.2	Priority Queue as Array of Queues	523
24.3	Priority Queue as a Single List	529
24.4	Priority Queue as Heap	535
24.5	A Note on the Implementation	539
24.6	Summary	539
<b>25</b>	<b>Mapping and Hash Table</b>	<b>541</b>
25.1	Introduction	541
25.2	Mappings	542
25.3	Representing Mappings	544
25.4	Sparse Mappings	547
25.5	An OrderedMapping Class	548
25.6	Associative Arrays	565
25.7	Hashing	565
25.8	Open Hash Table	568
25.9	Chaining Hash Table	577
25.10	Summary	586

<b>26 Searching</b>	<b>587</b>
26.1 Introduction	587
26.2 Some Constraints	587
26.3 Linear Search (aka Sequential Search)	588
26.4 Logarithmic Search (aka Binary Search)	589
26.5 Binary Tree Search	593
26.6 Searching using Hash Tables	594
26.7 Summary	595
<b>27 Sorting — Comparative Methods</b>	<b>597</b>
27.1 Introduction	597
27.2 Comparison Sorts	598
27.3 Selection Sort	599
27.4 Bubble Sort	601
27.5 Insertion Sort (aka Sifting)	602
27.6 Quicksort	605
27.7 Binary Tree Sort	609
27.8 Heap Sort	610
27.9 Merge Sort	612
27.10 Summary	615
<b>28 Sorting — Distributive Methods</b>	<b>617</b>
28.1 Introduction	617
28.2 Distributive Sorts	617
28.3 Count Sort (aka Pigeonhole Sort or Bucket Sort)	618
28.4 Radix Sort (aka Digit Sort)	622
28.5 Summary	627
<b>29 Sorting — External Methods</b>	<b>629</b>
29.1 Introduction	629
29.2 Architectural Issues	631
29.3 Balanced Multi-way Merge Sort	635
29.4 Polyphase Merge Sort	644
29.5 Summary	653
<b>Part 4 Case Studies in Developing Programs</b>	
<b>30 Introducing the Case Studies</b>	<b>657</b>
30.1 Introduction	657
30.2 The Case Studies	658
30.3 The Presentations of the Case Studies	659

<b>31 A Mail Merge System</b>	<b>661</b>
31.1 Introduction	661
31.2 Understanding the Problem	663
31.2.1 An Initial Analysis	663
31.2.2 An Initial Design	664
31.2.3 Objects?	667
31.2.4 A More Detailed Design	667
31.2.5 Streams and Files	669
31.3 A Command Line Implementation	670
31.4 A Version with a Graphical User Interface	679
31.4.1 Introduction	679
31.4.2 The User Interface	680
31.4.3 The Source Code	686
31.5 Summary	703
<b>32 Pedestrian Crossing Simulation</b>	<b>705</b>
32.1 Introduction	705
32.2 The Initial Problem Specification	705
32.3 The Initial Thinking	706
32.3.1 Lights	706
32.3.2 Traffic Lights	707
32.3.3 Pedestrian Lights	708
32.4 A First Pass	709
32.5 A Second Round	721
32.6 A Third Round	734
32.7 Oh Dear... Round Four is Round Three Re-visited	735
32.8 A Fifth Round	747
32.9 Summary	748
<b>33 Simulating Ants</b>	<b>749</b>
33.1 Introduction	749
33.2 The Simulation Model	750
33.3 The Program Specification	752
33.4 Initial Analysis and Design	753
33.5 A Second Pass	754
33.6 The Abstract Class Framework	755
33.6.1 Class Grid	756
33.6.2 Interface Factory	760
33.6.3 Class Patch	760
33.6.4 Class Turtle	764
33.6.5 Class Display	765
33.6.6 Class MainLoop	766
33.6.7 Class RandomGen	767
33.6.8 The Complete Framework	768

33.7 The Concrete Classes	768
33.7.1 Class AntPatch	769
33.7.2 Class Ant	770
33.7.3 Class AntWorld	772
33.7.4 Class AntFactory	773
33.7.5 Class AntApp — The Application Class	774
33.7.6 Class AntDisplay	776
33.7.7 Class GraphicDisplayCanvas	778
33.8 The Complete Program	779
33.9 Summary	780

## **Part 5 Endmatter**

<b>Appendix A Appendices</b>	<b>783</b>
<b>Appendix B The CRC Method</b>	<b>793</b>
<b>Appendix C Bibliography</b>	<b>799</b>
<b>Index</b>	<b>803</b>