
Fundamentals of Java Programming

Mitsunori Ogihara

Fundamentals of Java Programming

Mitsunori Ogihara
Department of Computer Science
University of Miami
Miami, FL, USA

ISBN 978-3-319-89490-4 ISBN 978-3-319-89491-1 (eBook)
<https://doi.org/10.1007/978-3-319-89491-1>

Library of Congress Control Number: 2018940440

© Springer Nature Switzerland AG 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To my family

Preface

This book was born from the desire of having an introductory Java programming textbook whose contents can be covered in one semester. The book was written with two types of audience in mind: those who intend to major in computer science and those who want to get a glimpse of computer programming. The book does not cover graphical user interfaces or the materials that are taught in a data structure course. The book very quickly surveys the Java Collection Framework and the generics in the penultimate chapter. The book also covers the concepts of online and recursive algorithms in the last chapter. The instructors who choose to use this textbook are free to skip these chapters if there is no sufficient time. Except for the code examples that receive parameters from the command line, the code examples can be compiled and run in a command-line environment as well as in IDEs. To execute those code examples in an IDE, the user must follow the step of provide args before execution. The code examples appearing in the book have very few comments, since the actions of the code are explained in the prose. The code examples with extensive comments are available for the publisher. There are PDF lecture slides accompanying the book. They are prepared using the Beamer environment of $\text{\LaTeX}$. The source codes of the lecture slides may be available through the publisher.

Acknowledgements I would like to take this opportunity to thank those who helped me in writing this book: Paul Drougas (Springer), Victor Milenkovic (University of Miami), Ted Pawlicki (University of Rochester), Ken Regan (University at Buffalo), Geoff Sutcliffe (University of Miami), and Osamu Watanabe (Tokyo Institute of Technology). Huge thanks go to my daughter Ellen, who painstakingly read through the draft and provided constructive criticisms.

Miami, FL, USA

Mitsunori Ogihara

Contents

Part I Programming Basics

1	“Hello, World!”	3
1.1	The Programming Environment for Java	3
1.1.1	The Java Virtual Machine (JVM)	3
1.1.2	Changing Folders in a Command Line Interface	4
1.1.3	Source Codes, Bytecodes, and Compilation	6
1.2	The First Program, “Hello, World!”	7
1.2.1	Methods and Their Declarations	9
1.2.2	<code>System.out.println</code> and <code>System.out.print</code>	10
1.2.3	Spacing in the Source Code	11
1.2.4	Commenting	11
1.2.5	Errors	13
1.3	Using Multiple Statements	15
1.3.1	<code>System.out.println</code> and <code>System.out.print</code> (Reprise)	15
1.3.2	Printing Multiple-Line Texts on the Screen	16
1.3.3	Escaping Characters	18
1.3.4	Printing Shapes	20
2	Using Data for Computation	29
2.1	Data	29
2.1.1	Data and Their Taxonomy	29
2.1.2	Number Literals	30
2.1.3	Variable Declarations	32
2.1.4	Assigning Values to Variables	33
2.2	The Primitive Data Types	35
2.3	Using Variables for Computation	37
2.3.1	Quarterbacks Program (Reprise)	37
2.3.2	Number Arithmetics	39
2.3.3	Computing the Body-Mass Index	46
2.3.4	Sum of Integers from 1 to 100 à la Gauss	48
2.3.5	Simplified Expressions	51
2.4	An Introduction to the String Data Type	56
2.4.1	The String Data Type	56
2.4.2	String Concatenation	59

3	Reading Keyboard Input	69
3.1	Class Scanner	69
3.1.1	Importing Source Codes	69
3.1.2	The Instantiation of a Scanner Object	70
3.2	Reading Data with a Scanner Object	71
3.3	Reading Input from the Keyboard	77
4	Decomposing Code into Components	87
4.1	Procedural Decomposition	87
4.1.1	Printing Rectangles	87
4.1.2	Printing Quadrangles	92
4.1.3	“Old MacDonald Had a Farm”	95
4.1.4	A General Strategy for Procedural Decomposition	101
4.2	Using Multiple Program Files	104
5	Passing Values to and from Methods	109
5.1	Passing Values to Methods	109
5.1.1	Methods That Work with Parameters	109
5.1.2	Method Overloading	115
5.2	Receiving a Value from a Method	119
5.3	Class Math	124
5.3.1	Mathematical Functions in Java	124
5.3.2	Mortgage Calculation	131
6	Conditions and Their Use for Controlling the Flow of Programs	143
6.1	Condition and Its Evaluation	143
6.2	The If Statements	149
6.2.1	If	149
6.2.2	Else	156
6.2.3	If-Else Inside If/Else	158
6.2.4	Truncation of Conditional Evaluations	162
 Part II Loops		
7	For-Loops	173
7.1	Using For-Loops for Repetition	173
7.2	Iteration	180
7.2.1	Simple Iteration	180
7.2.2	Iteration with an Auxiliary Variable	186
7.3	Double For-Loops	187
7.4	Computing the Maximum and Minimum in a Series of Numbers	193
7.5	A Betting Game	195
7.5.1	For-Loops with Skipped Execution	195
7.5.2	The Statements <code>continue</code> and <code>break</code>	199
7.6	Computing the Fibonacci Sequence	201
8	Formatted Printing Using <code>printf</code>	211
8.1	General Rules for <code>printf</code>	211
8.2	Formatted Printing of String Data	212
8.3	Formatted Printing of Integers	214

8.4	Formatted Printing of Floating Point Numbers	216
8.5	Printing the Fibonacci Sequence (Reprise).....	218
9	Classes <code>String</code> and <code>StringBuilder</code>	225
9.1	Methods for Obtaining Information from <code>String</code> Data	225
9.2	Methods for Comparing <code>String</code> Data with Another.....	228
9.2.1	The Equality Test and the Comparison in Dictionary Order	228
9.2.2	The Prefix and Suffix Tests	230
9.3	Methods for Searching for a Pattern in a <code>String</code> Data	231
9.4	Methods for Creating New <code>String</code> Data from Another	234
9.4.1	<code>String.format</code>	237
9.5	Class <code>StringBuilder</code>	237
10	The Switch Statements	245
10.1	The Syntax of Switch Statements	245
10.2	Using a <code>char</code> Data in a Switch-Statement.....	252
10.3	Using a <code>String</code> Data in a Switch Statement	257
11	While-Loops and Do-While Loops	263
11.1	Using While-Loops	263
11.1.1	The Syntax of While-Loops	263
11.1.2	Summing Input Numbers Until the Total Reaches a Goal	265
11.1.3	Integer Overflow	266
11.1.4	Vending Machines	268
11.1.5	The Collatz Conjecture	270
11.1.6	Converting Decimal Numbers to Binary Numbers	273
11.1.7	Infinite Loops and Their Termination.....	276
11.2	Using Do-While Loops.....	276
11.2.1	The Syntax of Do-While Loops	276
11.2.2	“Waiting for Godot”	277
11.2.3	Converting Decimal Numbers to Binary Numbers (Reprise)	278
11.3	CTRL-D	279
11.4	Approximating the Square Root of a Real Number	283

Part III Arrays and Objects

12	Arrays	295
12.1	Arrays.....	295
12.1.1	The Structure of an Array	295
12.1.2	Computing the Number of Occurrences.....	296
12.1.3	<code>ArrayIndexOutOfBoundsException</code>	303
12.2	Relative Indexing	305
12.2.1	The Concept of Relative Indexing	305
12.2.2	Calculating the BMI for a Range of Weight Values	306
12.2.3	Counting the Occurrences of Characters	307
12.3	Arrays of <code>boolean</code> Data	310
12.4	Using Multiple Arrays	314
12.5	<code>String</code> Methods That Return an Array	317

13 The Class Arrays and Resizing Arrays	325
13.1 The Class Arrays	325
13.2 Reordering Elements in an Array	329
13.2.1 Reversing the Order of Elements	332
13.2.2 Cyclic Shifting	332
13.3 Modifications of an Array That Require Resizing	336
13.3.1 Insertion and Deletion	336
13.3.2 Adjoining Two Arrays	341
13.4 args	342
13.5 Searching for an Element in an Array	343
13.5.1 Sequential Search	343
13.5.2 Binary Search	344
13.6 Arrays with Capacity and Size	345
14 Multidimensional Arrays	357
14.1 Rectangular Arrays	357
14.1.1 Defining Multi-Dimensional Arrays	357
14.1.2 Summing the Elements in Subsequences	358
14.2 Jagged Arrays	361
15 Class File	367
15.1 Class File	367
15.1.1 The File Path and the Instantiation of a File Object	367
15.1.2 File Methods	368
15.1.3 Exception Handling	370
15.1.4 File Listing	376
15.2 Using Scanner Objects to Read from Files	378
15.3 Using PrintStream to Write to Files	384
16 Designing Object Classes	391
16.1 Packaging a Group of Data as an Object	391
16.1.1 The Position of a Game Piece	391
16.1.2 Private Instance Variables and the toString Method	395
16.1.3 Using Constants in an Object Class	398
16.1.4 Information Hiding	402
16.2 An Object Class Representing a Bank Account	408
16.3 Array with Capacity and Size (Reprise)	415
Part IV Advanced Concepts	
17 Interfaces, Inheritance, and Polymorphism	427
17.1 Interface	427
17.1.1 The Structure of an Interface	427
17.1.2 A Simple Pizza Application	427
17.2 Subclasses and Superclasses	435
17.2.1 Extending Existing Classes and Interfaces	435
17.2.2 Writing Extensions	437
17.3 Polymorphism	445

17.4	Boxed Data Types	446
17.5	Interface Comparable	447
18	Generic Class Parameters and the Java Collection Framework	457
18.1	ArrayList	457
18.1.1	Maintaining a Collection of Merchandise Items	457
18.1.2	The Class for Merchandise Item	457
18.1.3	The Comparator Class	458
18.1.4	The Collection of Merchandise Items That Uses ArrayList	459
18.1.5	The Main Class	463
18.2	The Dynamic Maintenance of the Largest K Values	471
18.3	The Java Collection Framework	473
18.3.1	The Framework	473
18.3.2	Some Classes from the Framework	475
18.3.3	A Demonstration	477
19	Online and Recursive Algorithms	485
19.1	Online Algorithms	485
19.1.1	The Definition of Online Algorithms	485
19.1.2	Computing Recurrences	486
19.1.3	Computing the Factorial Function	489
19.2	Recursive Algorithms	492
19.2.1	Computing the Factorial Function Recursively	492
19.2.2	The Greatest Common Divisor of Two Integers	497
19.2.3	The Tower of Hanoi	500
Index		507

List of Figures

Fig. 1.1	The program layers, JVM, JRE, and JDK	4
Fig. 1.2	A screen short of Terminal on a Mac OS X machine	4
Fig. 1.3	An IDE screen of Eclipse	5
Fig. 1.4	The compilation and execution of HelloWorld.java	6
Fig. 3.1	The results of five consecutive calls of next	73
Fig. 3.2	The results of executing a program that uses next and nextLine	77
Fig. 4.1	The method calls in Rectangle01	91
Fig. 4.2	The method calls in Rectangle02	93
Fig. 4.3	The decomposition of actions in the generation of the quadrant	94
Fig. 4.4	The dependency among methods in OldMacDonaldDecomposed.java ..	101
Fig. 4.5	The dependency among methods in the two source code	104
Fig. 5.1	The call-by-reference concept	116
Fig. 6.1	The execution diagram of an if-statement	150
Fig. 6.2	The execution diagram of Temperature01.java	152
Fig. 6.3	The combinations of temperature and humidity considered in Temperature03	154
Fig. 6.4	The execution diagram of an if-else statement	156
Fig. 6.5	A hypothetical situation with interwoven conditions	160
Fig. 7.1	A generic flow chart of for-loops	176
Fig. 7.2	The code execution diagram of ForExample	177
Fig. 11.1	A diagram that represents the while-loop	264
Fig. 12.1	A view of an array	296
Fig. 13.1	Swapping values between two array elements	331
Fig. 13.2	Reversing the order of appearance of elements in an array	332
Fig. 13.3	The results obtained by executing cyclic shifts	333
Fig. 13.4	An algorithm for left cyclic shift	334
Fig. 13.7	An array with capacity and size	346

Fig. 13.8	The concept of a array with capacity and size	346
Fig. 14.1	The structure of a multi-dimensional array	358
Fig. 15.1	The mechanism for handling run-time errors	371
Fig. 16.1	An 8×8 game board	392
Fig. 16.3	A black box	402
Fig. 17.1	Two interfaces and their implementations	436
Fig. 17.2	A <code>PizzaComplex</code> object	438
Fig. 18.1	The Java Collection Framework	474
Fig. 18.2	A <code>LinkedList</code>	476
Fig. 18.3	A hash table	477
Fig. 19.1	The value passing that occurs during the computation of the factorial (part 1) ..	495
Fig. 19.2	The value passing that occurs during the computation of the factorial (part 2) ..	495
Fig. 19.3	The value passing that occurs during the computation of the factorial (part 3) ..	495
Fig. 19.4	The value passing that occurs during the computation of the factorial (part 4) ..	495
Fig. 19.5	The value passing that occurs during the computation of the factorial (part 5) ..	495
Fig. 19.6	The value passing that occurs during the computation of the factorial (part 6) ..	496
Fig. 19.7	The value passing that occurs during the computation of the factorial (part 7) ..	496
Fig. 19.8	The value passing that occurs during the computation of the factorial (part 8) ..	496
Fig. 19.9	The value passing that occurs during the computation of the factorial (part 9) ..	496
Fig. 19.10	The value passing that occurs during the computation of the factorial (part 10) .	496
Fig. 19.11	The value passing that occurs during the computation of the factorial (part 11) .	497
Fig. 19.12	An example of the tower of Hanoi	500
Fig. 19.13	The solution to a small Tower of Hanoi problem	502

List of Tables

Table 1.1	A short list of commands	5
Table 1.2	The list of meaningful symbols in Java	21
Table 2.1	The primitive data type	35
Table 2.2	The list of reserved words in Java	39
Table 2.3	The selection of number types	45
Table 3.1	Selected methods of <code>Scanner</code>	75
Table 5.1	One-parameter functions in <code>Math</code>	125
Table 5.2	Two-parameter functions in <code>Math</code>	125
Table 9.1	A list of <code>String</code> methods	226
Table 10.1	The output generated by the three examples of <code>switch</code>	246
Table 15.1	A list of <code>File</code> methods	369