

A **Chapman & Hall** Book



CRC Press
Taylor & Francis Group



JAVA Programming Exercises

Volume Two:
Java Standard Library



Christian Ullenboom

Java Programming Exercises

Take the next step in raising your coding skills and dive into the intricacies of Java Standard Libraries. You will continue to raise your coding skills, and test your Java knowledge on tricky programming tasks, with the help of the pirate *Captain CiaoCiao*. This is the second of two volumes which provide you with everything you need to excel in your Java journey, including tricks that you should know in detail as a professional, as well as intensive training for clean code and thoughtful design that carries even complex software.

Features:

- 149 tasks with commented solutions on different levels
- For all paradigms: object-oriented, imperative, and functional
- Clean code, reading foreign code, and object-oriented modeling

With numerous best practices and extensively commented solutions to the tasks, these books provide the perfect workout for professional software development with Java.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Java Programming Exercises

Volume Two: Java Standard Library

Christian Ullenboom



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

A CHAPMAN & HALL BOOK

Designed cover image: Mai Loan Nguyen Duy, Rheinwerk Verlag GmbH

First edition published 2025
by CRC Press
2385 NW Executive Center Drive, Suite 320, Boca Raton FL 33431

and by CRC Press
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

©2023 Christian Ullenboom. First published in the German language under the title “Captain CiaoCiao erobert Java” (ISBN 978-3-8362-8427-1) by Rheinwerk Verlag GmbH, Bonn, Germany.

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, access www.copyright.com or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. For works that are not available on CCC please contact mpkbookspermissions@tandf.co.uk

Trademark notice: Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

ISBN: 978-1-032-80117-9 (hbk)
ISBN: 978-1-032-79801-1 (pbk)
ISBN: 978-1-003-49555-0 (ebk)

DOI: 10.1201/9781003495550

Typeset in Times
by codeMantra

Access the Support Material: <https://routledge.com/9781032798011>

Contents

<i>About the Author</i>	xiii
Introduction	1
Previous Knowledge and Target Audience	1
Working with the Book	2
The Suggested Solutions	2
Use of the Book	3
Required Software	3
Used Java Version in the Book	4
JVM	4
Development Environment	4
Conventions	4
Helping Captain CiaoCiao and Bonny Brain	5
1 Advanced String Processing	6
Format Strings	6
Build ASCII Table ★	6
Aligned Outputs ★	8
Regular Expressions and Pattern Recognition	8
Quiz: Define Regex ★	8
Determine Popularity in Social Media ★	8
Detect Scanned Values ★	9
Quiet Please! Defuse Shouting Texts ★	10
Converting Time from AM/PM Format to 24-Hour Format ★★	11
Decompose Strings into Tokens	12
Split Address Lines with the StringTokenizer ★	12
Split Sentences into Words and Reverse Them ★	13
Check Relations between Numbers ★	13
Convert A1 Notation to Columns and Rows ★★	13
Parse Simple CSV Files with Coordinates ★	14
Compress Strings Lossless by Runlength Encoding ★★★	14
Character Encodings and Unicode Collation Algorithm	15
Quiz: Encoding for Unicode Characters ★	15
Quiz: Order of Strings with and without Collator ★	16
Suggested Solutions	16
Build ASCII Table	16
Aligned Outputs	17
Quiz: Define Regex	18
Determine Popularity in Social Media	19
Detect Scanned Values	19
Quiet Please! Defuse Shouting Texts	20
Converting Time from AM/PM Format to 24-Hour Format	20
Split Address Lines with the StringTokenizer	21

Split Sentences into Words and Reverse Them	22
Check Relations between Numbers	23
Convert A1 Notation to Columns and Rows	24
Parse Simple CSV Files with Coordinates	25
Compress Strings Lossless by Runlength Encoding	26
Quiz: Encoding for Unicode Characters	28
Quiz: Order of Strings with and without Collator	28
2 Mathematics	30
The Class Math	30
Quiz: Rule of Thumb ★	30
Check If Tin Tin Cheated on Rounding ★	31
Huge and Very Precise Numbers	32
Calculate Arithmetic Mean of a Large Integer ★	32
Number by Number over the Phone ★	33
Develop Class for Fractions and Truncate Fractions ★★	34
Suggested Solutions	35
Quiz: Rule of Thumb	35
Check If Tin Tin Cheated on Rounding	36
Calculate Arithmetic Mean of a Large Integer	37
Number by Number over the Phone	38
Develop Class for Fractions and Truncate Fractions	38
3 Locale, Date, and Time	42
Languages and Countries	42
Apply Country-/Language-Specific Formatting for Random Number ★	43
Date and Time Classes	44
Formatting Date Output in Different Languages ★	44
On Which Day Does Sir Francis Beaufort Celebrate His Birthday? ★	45
Find All Friday the 13th ★	45
Get Average Duration of Karaoke Nights ★	46
Parse Different Date Formats ★★★	46
Suggested Solutions	47
Apply Country-/Language-Specific Formatting for Random Number	47
Formatting Date Output in Different Languages	47
On Which Day Does Sir Francis Beaufort Celebrate His Birthday?	48
Find All Friday the 13th	50
Get Average Duration of Karaoke Nights	51
Parse Different Date Formats	52
4 Concurrent Programming with Threads	54
Create Threads	55
Create Threads for Waving and Flag Waving ★	56
No More Waving Flags: End Threads ★	56
Parameterize Runnable ★★	56
Execute and Idle	57
Delay Execution by Sleeping Threads ★★	57
Watch File Changes by Threads ★	58
Catch Exceptions ★	58
Thread Pools and Results	59
Using Thread Pools ★★	60
Get Last Modification of Web Pages ★★	60

Protect Critical Sections	61
Writing Memories into a Poetry Album ★	62
Thread Cooperation and Synchronization Helper	64
Attending the Banquet with the Captains—Semaphore ★★	64
Swearing and Insulting—Condition ★★	65
Take Pens Out of Paintbox—Condition ★★	65
Play Rock, Paper, Scissors—CyclicBarrier ★★★	66
Find the Fastest Runner—CountDownLatch ★★	67
Suggested Solutions	68
Create Threads for Waving and Flag Waving	68
No More Waving Flags: End Threads	69
Parameterize Runnable	70
Delay Execution by Sleeping Threads	70
Watch File Changes by Threads	72
Catch Exceptions	73
Using Thread Pools	74
Get Last Modification of Web Pages	75
Writing Memories into a Poetry Album	77
Attending the Banquet with the Captains—Semaphore	78
Swearing and Insulting—Condition	79
Take Pens Out of Paintbox—Condition	80
Play Rock, Paper, Scissors—CyclicBarrier	82
Find the Fastest Runner—CountDownLatch	83
5 Data Structures and Algorithms	85
The Types of the Collection API	86
Quiz: Search for StringBuilder ★	87
Lists	88
Singing and Cooking: Traverse Lists and Check Properties ★	88
Filter Comments from Lists ★	89
Shorten Lists Because There Is No Downturn ★	89
Eating with Friends: Compare Elements, Find Commonalities ★	90
Check Lists for the Same Order of Elements ★	90
And Now the Weather: Find Repeated Elements ★	91
Generate Receipt Output ★	92
Quiz: Arrays Decorated ★	92
Quiz: Searched and Not Found ★	92
Everything Tastes Better with Cheese: Insert Elements into Lists ★	93
Quiz: With Nothing but Trouble ★	93
Search Elements with the Iterator and Find Covid Cough ★★	94
Move Elements, Play Musical Chairs ★	95
Programming a Question Game with Planets ★★	95
Sets	97
Form Subsets, Find Common Elements ★	97
Quiz: Great Swords ★	98
Get All Words Contained in a Word ★★	99
Exclude Duplicate Elements with a Uniqueliterator ★★	99
Map Keys to Values	100
Convert Two-Dimensional Arrays to Map ★	100
Convert Text to Morse Code and Vice Versa ★	101
Remember Word Frequency with Associative Memory ★★	101
Read In and Read Out Colors ★★	102

Read in Names and Manage Lengths ★	103
Find Missing Characters ★★	103
Calculate Number of Paths to the Three-Headed Monkey ★★	103
Manage Holidays in a Sorted Associative Store ★	104
Quiz: Keys in a HashMap ★★	105
Determine Commonality: Party Set and Souvenir ★	106
Properties	106
Develop Convenient Properties Decorator ★★	106
Stack and Queues	108
Program RPN Pocket Calculator ★	108
BitSet	109
Forget No Ship ★	109
Find Duplicate Entries and Solve the Animal Chaos ★	110
Thread-Safe Data Structures	110
Loading Ship ★★	111
Handle Important Messages First ★★	112
If Used Up, Create a New One ★★★	113
Suggested Solutions	114
Quiz: Search for StringBuilder	114
Singing and Cooking: Traverse Lists and Check Properties	114
Filter Comments from Lists	116
Shorten Lists Because There Is No Downturn	117
Eating with Friends: Compare Elements, Find Commonalities	118
Check Lists for the Same Order of Elements	118
And Now the Weather: Find Repeated Elements	120
Generate Receipt Output	121
Quiz: Arrays Decorated	123
Quiz: Searched and Not Found	124
Everything Tastes Better with Cheese: Insert Elements into Lists	124
Quiz: With Nothing but Trouble	125
Search Elements with the Iterator and Find Covid Cough	126
Move Elements, Play Musical Chairs	128
Programming a Question Game with Planets	129
Form Subsets, Find Common Elements	130
Quiz: Great Swords	131
Get All Words Contained in a Word	132
Exclude Duplicate Elements with a Uniqueliterator	133
Convert Two-Dimensional Arrays to Map	134
Convert Text to Morse Code and Vice Versa	136
Remember Word Frequency with Associative Memory	138
Read In and Read Out Colors	140
Read in Names and Manage Lengths	141
Find Missing Characters	143
Calculate the Number of Paths to the Three-Headed Monkey	145
Manage Holidays in a Sorted Associative Store	146
Quiz: Keys in a HashMap	147
Determine Commonality: Party Set and Souvenir	148
Develop Convenient Properties Decorator	148
Program RPN Pocket Calculator	150
Forget No Ship	151
Find Duplicate Entries and Solve the Animal Chaos	152
Loading Ship	153

Handle Important Messages First	154
If Used Up, Create a New One	155
Notes	156
6 Java Stream-API	157
Regular Streams with Their Terminal and Intermediate Operations	157
Hero Epic: Meet Stream API ★	157
Quiz: Double Output ★	158
Get the Beloved Captain from a List ★	158
Frame Pictures ★	159
Look and Say ★★	160
Remove Duplicate Islands of Rare Earth Metals ★★★	161
Where Are the Sails? ★★	162
Buy the Most Popular Car ★★★	163
Primitive Streams	163
Detect NaN in an Array ★	163
Generate Decades ★	164
Generate Array with Constant Content via Stream ★	164
Draw Pyramids ★	165
Teddies Labeled with Letters ★	165
Get the Letter Frequency of a String ★	166
From 1 to 0, from 10 to 9 ★★	166
The Annual Octopus Contest ★★	166
Merge Three int Arrays ★	167
Determine Winning Combinations ★★	167
Statistics	168
The Fastest and Slowest Paddlers ★	168
Calculate Median ★★	169
Calculate Temperature Statistics and Draw Charts ★★★	169
Suggested Solutions	170
Hero Epic: Meet Stream API	170
Quiz: Double Output	176
Get the Beloved Captain from a List	176
Frame Pictures	177
Look and Say	178
Remove Duplicate Islands of Rare Earth Metals	179
Where Are the Sails?	181
Buy the Most Popular Car	182
Detect NaN in an Array	183
Generate Decades	183
Generate Array with Constant Content via Stream	184
Teddies Labeled with Letters	184
Draw Pyramids	185
Get the Letter Frequency of a String	185
From 1 to 0, from 10 to 9	186
The Annual Octopus Contest	187
Merge Three int Arrays	188
Determine Winning Combinations	189
The Fastest and Slowest Paddlers	189
Calculate Median	190
Calculate Temperature Statistics and Draw Charts	191
Note	195

7 Files, Directories, and File Access	196
Path and Files	196
Display Saying of the Day ★	197
Merge Hiding Places ★	197
Create Copies of a File ★★	197
Generate a Directory Listing ★	198
Search for Large GIF Files ★	198
Descend Directories Recursively and Find Empty Text Files ★	198
Develop Your Own Utility Library for File Filters ★★★	199
Random Access to File Contents	199
Output Last Line of a Text File ★★	200
Suggested Solutions	200
Display Saying of the Day	200
Merge Hiding Places	201
Create Copies of a File	201
Generate a Directory Listing	202
Search for Large GIF Files	203
Descend Directories Recursively and Find Empty Text Files	204
Develop Your Own Utility Library for File Filters	205
Output Last Line of a Text File	208
8 Input/Output Streams	210
Direct Data Streams	211
Get the Number of Different Places (Read Files) ★	211
Convert Python Program to Java (Write File) ★	211
Generate Target Code (Write File) ★	212
Convert File Contents to Lowercase (Read and Write File) ★	213
Convert PPM Graphics to ASCII Grayscale ★★★	213
Split Files (Read and Write Files) ★★	215
Nesting Streams	215
Quiz: DataInputStream and DataOutputStream ★	215
Compress Number Sequences with the GZIPOutputStream ★	216
Serialization	216
(De)serialize Data for Chat and Convert It to Text ★★	216
Quiz: Requirement for Serialization ★	217
Save Last Inputs ★★	217
Suggested Solutions	218
Get the Number of Different Places (Read Files)	218
Convert Python Program to Java (Write File)	219
Generate Target Code (Write File)	220
Convert File Contents to Lowercase (Read and Write File)	221
Convert PPM Graphics to ASCII Gray Scale	222
Split Files (Read and Write Files)	224
Quiz: DataInputStream and DataOutputStream	225
Compress Number Sequences with the GZIPOutputStream	228
(De)serialize Data for Chat and Convert Them to Text	229
Quiz: Requirement for Serialization	230
Save Last Inputs	231
Note	233

9 Network Programming	234
URL and URLConnection	234
Download Remote Images via URL ★	235
Read Remote Text File from URL ★	235
HTTP Client	236
Top News from Hacker News ★★	236
Socket and ServerSocket	237
Implement a Swear Server and a Client ★★	237
Implement a Port Scanner ★★	237
Suggested Solutions	238
Download Remote Images via URL	238
Read Remote Text File from URL	239
Top News from Hacker News	240
Implement a Swear Server and a Client	241
Implement a Port Scanner	243
10 Process XML, JSON, and Other Data Formats	246
XML Processing with Java	247
Write XML File with Recipe ★	247
Check If All Images Have an alt Attribute ★	248
Writing Java Objects with JAXB ★	248
Read in Jokes and Laugh Heartily ★★	249
JSON	251
Hacker News JSON Exploit ★	251
Read and Write Editor Configurations as JSON ★★	252
HTML	253
Load Wikipedia Images with jsoup ★★	253
Office Documents	253
Generate Word Files with Screenshots ★★	253
Archives	254
Play Insect Sounds from ZIP Archive ★★	254
Suggested Solutions	255
Write XML File with Recipe	255
Check If All Images Have an alt Attribute	258
Writing Java Objects with JAXB	259
Read in Jokes and Laugh Heartily	260
Hacker News JSON Exploit	261
Read and Write Editor Configurations as JSON	262
Load Wikipedia Images with jsoup	263
Generate Word Files with Screenshots	264
Play Insect Sounds from ZIP Archive	265
Note	266
11 Database Access with JDBC	267
Database Management Systems	267
Prepare H2 Database ★	268
Database Queries	268
Query All Registered JDBC Drivers ★	268
Build Database and Execute SQL Script ★	269
Insert Data into the Database ★	270
Insert Data into the Database in Batch Mode ★	271

Insert Data with Prepared Statements ★	271
Request Data ★	272
Interactive Scrolling through the ResultSet ★	272
Pirate Repository ★★	272
Query Column Metadata ★	273
Suggested Solutions	274
Query All Registered JDBC Drivers	274
Build Database and Execute SQL Script	274
Insert Data into the Database	275
Insert Data into the Database in Batch Mode	276
Insert Data with Prepared Statements	277
Request Data	277
Interactive Scrolling through the ResultSet	278
Pirate Repository	279
Query Column Metadata	282
12 Operating System Access	284
Console	284
Colored Console Outputs ★	285
Properties	286
Windows, Unix, or macOS? ★	286
Unify Command-Line Properties and Properties from Files ★	286
Execute External Processes	287
Read the Battery Status via Windows Management Instrumentation ★★	287
Suggested Solutions	288
Colored Console Outputs	288
Windows, Unix, or macOS?	288
Unify Command-Line Properties and Properties from Files	289
Read the Battery Status via Windows Management Instrumentation	290
13 Reflection, Annotations, and JavaBeans	294
Reflection API	294
Create UML Class Diagram with Inheritance Relationships ★	295
Create UML Class Diagram with Properties ★	296
Generate CSV Files from List Entries ★★	297
Annotations	297
Create CSV Documents from Annotated Instance Variables ★★	297
Suggested Solutions	298
Create UML Class Diagram with Inheritance Relationships	298
Create UML Class Diagram with Properties	299
Generate CSV Files from List Entries	301
Create CSV Documents from Annotated Instance Variables	301
<i>Epilogue</i>	307
Code Golf Stack Exchange	307
Project Euler	307
Daily Programmer	307
Rosetta Code	308

About the Author

Christian Ullenboom started his programming journey at the tender age of ten, typing his first lines of code into a C64. After mastering assembler programming and early BASIC extensions, he found his calling on the island of Java, following his studies in computer science and psychology. Despite indulging in Python, JavaScript, TypeScript, and Kotlin vacations, he remains a savant of all things Java.

For over 20 years, Ullenboom has been a passionate software architect, Java trainer (check out <http://www.tutego.com>), and IT specialist instructor. His expertise has resulted in a number of online video courses and reference books:

- *Java: The Comprehensive Guide* (ISBN-13: 978-1493222957)
- *Java ist auch eine Insel: Java programmieren lernen mit dem umfassenden Standardwerk für Java-Entwickler* (ISBN-13: 978-3836287456)
- *Java SE 9 Standard-Bibliothek: Das Handbuch für Java-Entwickler* (ISBN-13: 978-3836258746)
- *Captain CiaoCiao erobert Java: Das Trainingsbuch für besseres Java* (ISBN-13: 978-3836284271)
- *Spring Boot 3 und Spring Framework 6: Professionelle Enterprise-Anwendungen mit Java* (ISBN-13: 978-3836290494)

Christian Ullenboom has been spreading Java love through his books for years, earning him the coveted title of Java Champion from Sun (now Oracle) way back in 2005. Only a select few—about 300 worldwide—have achieved this status, making him a true Java superstar.

As an instructor, Ullenboom understands that learning by doing is the most effective way to master a skill. So, he has compiled a comprehensive catalog of exercises that accompany his training courses. This book features a selection of those exercises, complete with documented solutions.

His roots are in Sonsbeck, a small town in the Lower Rhine region of Germany.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Introduction

Many beginners in programming often ask themselves, “How can I strengthen my skills as a developer? How can I become a better programmer?” The answer is simple: study, attend webinars, learn, repeat, practice, and discuss your work with others. Many aspects of programming are similar to learning new skills. Just as a book can’t teach you how to play a musical instrument, watching the *Fast and the Furious* movie series won’t teach you how to drive. The brain develops patterns and structures through repeated practice. Learning a programming language and a natural language have many similarities. Consistent use of the language, and the desire and need to express and communicate in it (just as you need to do so to order a burger or a beer), leads to gradual improvement in skills.

Books and webinars on learning a programming language are available, but reading, learning, practicing, and repeating are just one aspect of becoming a successful software developer. To create effective software solutions, you need to creatively combine your knowledge, just as a musician regularly practices finger exercises and maintains their repertoire. The more effective your exercises are, the faster you will become a master. This book aims to help you progress and gain more hands-on experience.

Java 21 declares more than 2300 classes, about 1400 interfaces, around 140 enumerations, approximately 500 exceptions, and a few annotation types and records are added to this. However, in practical terms, only a small subset of these types proves to be relevant. This book selects the most important types and methods for tasks, making them motivating, and following Java conventions. Alternative solutions and approaches are also presented repeatedly. The goal is to make non-functional requirements clear because the quality of programs is not just about “doing what it should.” Issues such as correct indentation, following naming conventions, proper use of modifiers, best practices, and design patterns are essential. The proposed solutions aim to demonstrate these principles, with the keyword being *Clean Code*.

PREVIOUS KNOWLEDGE AND TARGET AUDIENCE

The book is aimed at Java developers who are either new to Java or are already advanced and wish to learn more about the Java SE standard libraries. The intended audience includes:

- Computer science students
- IT specialists
- Java programmers
- Software developers
- Job applicants

The book is centered around tasks and fully documented solutions, with detailed explanations of Java peculiarities, good object-oriented programming practices, best practices, and design patterns. The exercises are best solved with a textbook, as this exercise book is not a traditional textbook. A useful approach is to work through a topic with a preferred textbook before attempting the exercises that correspond to it.

The first set of tasks are designed for programming beginners who are new to Java. As you gain more experience with Java, the tasks become more challenging. Therefore, there are tasks for both beginners and advanced developers.

The Java Standard Edition is augmented by numerous frameworks and libraries. However, this exercise book does not cover specific libraries or Java Enterprise frameworks like Jakarta EE or Spring (Boot). There are separate exercise books available for these environments. Additionally, the book does not require the use of tools like profiling tools, as these are beyond the scope of the book.

WORKING WITH THE BOOK

The task book is organized into different sections. The first section covers the Java language, followed by selected areas of the Java standard library, such as data structures or file processing. Each area is accompanied by programming tasks and “quiz” questions that contain surprises. Each section starts with a small motivation and characterization of the topic, followed by the exercises. Additional tips and hints are provided for particularly challenging assignments, while other exercises offer optional extensions for further exploration.

The majority of exercises are independent of each other, making it easy for readers to dive in anywhere. However, in the chapter on imperative programming, some tasks build on each other to develop a larger program, and the same goes for the chapter on object-oriented programming. The problem definitions make this clear, and more complex programs help to provide context for understanding different language characteristics. Furthermore, a more complex program can motivate readers to continue.

The exercises are rated with one, two, or three stars to indicate their complexity, although this rating is subjective to the author.

- 1 star ★: Simple exercises, suitable for beginners. They should be easy to solve without much effort. Often only transfer of knowledge is required, for example, by writing down things that are in a textbook differently.
 - 2 stars ★★: The effort is higher here. Different techniques have to be combined. Greater creativity is required.
 - 3 stars ★★★: Assignments with three stars are more complex, require recourse to more prior knowledge, and sometimes require research. Frequently, the tasks can no longer be solved with a single method, but require multiple classes that must work together.
-

THE SUGGESTED SOLUTIONS

The task book provides at least one suggested solution for each problem. The term “sample solution” is not used to avoid implying that the given solution is the best one and that all other solutions are useless. Readers are encouraged to compare their solutions with the proposed solution and can be satisfied if their solution is more elegant. All proposed solutions are commented, making it possible to follow all steps well.

The suggested solutions are compiled at the end of each chapter to reduce the temptation to look into a solution directly after the task, which takes the fun out of solving the task. The suggested solutions can also be found on the website <https://github.com/ullenboom/captain-ciaociao>. Some solutions contain comments of the type `//tag::solution[]`, which marks the parts of the solutions printed in the book.

USE OF THE BOOK

To become a software developer, you must master the art of turning problems into code, and that's where practice and role models come in. While there are plenty of exercises available online, they're often disorganized, poorly documented, and outdated. That's where this book shines, by offering a systematic approach to tasks and well-thought-out solutions. Studying these solutions and reading code in general helps the brain develop patterns and solutions that can be applied to future coding challenges. It's like reading the Bible; you need to read to understand and learn. Surprisingly, many software developers write code without bothering to read others' code, which can lead to confusion and misunderstanding. Reading good code elevates our writing skills by building patterns and solutions that our brains unconsciously transfer to our own code. Our brains form neuronal structures independently based on templates, and the quality of the input we receive matters greatly. Therefore, we should only feed our brains with good code, as bad solutions make for bad models. The book covers important topics such as exception handling or error handling, discussing the correct input values, identifying erroneous states, and how to handle them. In software, things can and will go wrong, and we must be prepared to deal with the less-than-perfect world.

It's easy for developers to get stuck in their ways of writing code, which is why it's important to explore new approaches and "expand our vocabulary", so to speak. For Java developers, libraries are their vocabulary, but too many enterprise Java developers write massive, non-object-oriented code. The solution is to continuously improve object-oriented modeling, which is precisely what this book demonstrates. It introduces new methods, creates new data types, and minimizes complexity. Additionally, functional programming is becoming increasingly important in Java development, and all solutions in this book take advantage of modern language features.

While some solutions may appear overly complex, the tasks and proposed solutions in this book can help developers improve their ability to concentrate and follow through with steps. In practice, the ability to concentrate and quickly comprehend code is crucial for developers. Often, developers must join a new team and be able to understand and modify unfamiliar source code, and possibly fix bugs. Those who wish to expand upon existing open-source solutions can also benefit from honing their concentration skills through these exercises.

In addition to its emphasis on the Java programming language, syntax, libraries, and object orientation, this book provides numerous side notes on topics such as algorithms, the historical evolution of programming, comparisons to other programming languages, and data formats. These additional insights and perspectives offer readers a more well-rounded understanding of software development beyond just the technical aspects.

If you're looking for one more reason to add this book to your collection, it doubles as a fantastic sleep aid!

REQUIRED SOFTWARE

While solving a task with just a pen and paper is possible in theory, modern software development requires the proper use of tools. Knowing programming language syntax, object-oriented modeling, and libraries is just the tip of the iceberg. Understanding the JVM, using tools like Maven and Git for version management, and becoming proficient in an IDE are all crucial aspects of professional software development. Some developers can even perform magic in their IDE, generating code and fixing bugs automatically.

USED JAVA VERSION IN THE BOOK

While Java 8 remains prevalent in enterprise settings, it's crucial for learners to become acquainted with the latest language features. Accordingly, whenever feasible, the suggested solutions in this book leverage Java 17. Not only is this version equipped with Long-term Support (LTS), but runtime environment providers also offer extensive support, ensuring that the release retains its relevance for an extended period.

JVM

If we want to run Java programs, we need a JVM. In the early days, this was easy. The runtime environment first came from Sun Microsystems, later from Oracle, which took over Sun. Today, it is much more confusing. Although a runtime environment can still be obtained from Oracle, the licensing terms have changed, at least for Java 8 up to Java 16. Testing and development are possible with the Oracle JDK, but not in production. In this case, Oracle charges license fees. As a consequence, various institutions compile their own runtime environments from the OpenJDK, the original sources. The best known are *Eclipse Adoptium* (<https://adoptium.net/>), *Amazon Corretto* (<https://aws.amazon.com/de/corretto>), *Red Hat OpenJDK* (<https://developers.redhat.com/products/openjdk/overview>) and others such as those from *Azul Systems* or *Bellsoft*. There is no specific distribution that readers are required to follow.

Development Environment

Java source code is just plain text, so technically a simple text editor is all you need. However, relying solely on Notepad or vi for productivity is like trying to win a race on a tricycle. Modern integrated development environments support us with many tasks: color highlighting of keywords, automatic code completion, intelligent error correction, insertion of code blocks, visualization of states in the debugger, and much more. It is therefore advisable to use a full development environment. Four popular IDEs are: *IntelliJ*, *Eclipse*, *Visual Studio Code*, and (*Apache*) *NetBeans*. Just like with Java runtime environments, the choice of IDE is left to the reader. Eclipse, NetBeans, and Visual Studio Code are all free and open-source, while IntelliJ Community Edition is also free, but the more advanced IntelliJ Ultimate Edition will cost you some cash.

Halfway through the book, we delve into implementing project dependencies using Maven in a few places.

CONVENTIONS

Code is written in `fix width font`, filenames are *italicized*. To distinguish methods from attributes, methods always have a pair of parentheses, such as in “the variable `max` contains the maximum” or “it returns `max()` the maximum”. Since methods can be overloaded, either the parameter list is named, as in `equals(Object)`, or an ellipsis abbreviates it, such as in “various `println(...)` methods”. If a group of identifiers is addressed, `*` is written, like `print*(...)` prints something on the screen.

In the suggested solutions, there are usually only the relevant code snippets, so as not to blow up the book volume. The name of the file is mentioned in the listing caption, like this:

VanillaJava.java

```
class VanillaJava { }
```



Lituz.com

**To'liq qismini
Shu tugmani
bosish orqali
sotib oling!**