# 100+ Solutions in Java

**2nd Edition**

*Everything you need to know to develop Java applications*

**Dhruti Shah**

www.bpbonline.com

Copyright © 2024 BPB Online

*All rights reserved.* No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor BPB Online or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

BPB Online has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, BPB Online cannot guarantee the accuracy of this information.

First published: 2021

Second published: 2024

Published by BPB Online
WeWork
119 Marylebone Road
London NW1 5PU

**UK | UAE | INDIA | SINGAPORE**

ISBN 978-93-5551-571-1

www.bpbonline.com

Kup książkę

## Dedicated to

*My beloved Parents:*

***Late Dr. Jyotikant S. Shah***
***Mrs. Ushakiran J. Shah***
*&*
*My little* ***Swara, Aaria*** *&* ***Madhav***

Kup książkę

# About the Author

**Dhruti Shah** is a multi-skilled, tech-savvy person with over 15 years of experience as a software trainer, technical writer, and manager in the IT education industry. She has been working extensively with Java technology for the last 10 years. She is also a Microsoft Certified Training Specialist who has trained over 2000 candidates worldwide on more than 10 technologies.

She has been appreciated as a model representative for India for flawlessly managing two prestigious international projects to set up and upgrade the Centre of Excellence in Information Technology in Panama and Costa Rica, Central America (a collaboration project of the Indian government with the governments of Panama and Costa Rica).

Kup książkę

# Acknowledgement

Writing a book on a technology that I have been obsessed with since I first learned about it, years ago, was a gratifying experience. However, it could never have been possible without the support of all my family members and friends. They always believed in me, encouraged me to follow my dreams, and gave me the confidence to face my struggles.

I would like to thank my mother, Mrs. Ushakiran Shah, who is my friend, philosopher, and guide. Her belief in me and my aspirations has given me the courage to face challenges head on and emerge a victor. Further, I would like to thank all my readers whose valuable feedback has helped me while writing this revised edition of my book.

Finally, I would like to extend my gratitude to the entire team of BPB publications for their support and appreciation of my work.

Kup książkę

# Preface

Java is a programming language that has been around for decades and has proved its potential ever since as a versatile programming language. Over the years, Java has advanced tremendously and has become one of the preferred programming languages for the development of applications ranging from standalone to Web applications as well as mobile apps. It is no longer a simple client-side language but is more dynamic and supports development for application servers, embedded devices, and much more.

This book is a result of several years of application development in Java and the experience gained by using different features introduced with each new version of Java. The goal of this book is to give a beginner enough knowledge to start application development in Java. Java is an evolving technology, and this book attempts to introduce the readers to basic features of Java and the new features of Java 8 to Java 19. It aims to gradually introduce the reader to a new programming approach called modular programming. The book presumes that the reader has a basic idea about programming and aspires to begin development using Java. It has been written after extensive research and provides ample examples and demonstrations to help you take the first step to learn new technology. This book will prove to be a great reference for beginners as well as professionals to begin development in Java. Over the 14 chapters of this book, you will learn the following:

**Chapter 1: Introduction to Java-** introduces the concept of object-oriented programming and explains how Java has evolved as object-oriented programming (OOP) language. It explains the versions and features of Java and steps to create an application by using JDK 8 and JDK 10.

**Chapter 2: Java Programming Constructs -** discusses the different programming constructs of Java language such as comments, variables, datatypes, and operators. It then shows you the use of decision-making constructs, looping constructs, and branch statements.

**Chapter 3: Java Application Components -** introduces Java classes, objects, variables, methods, access specifiers, and constructors. It explains the implementation of polymorphism, creation of packages and use of keywords such as static, final, and this keyword.

**Chapter 4: Java Reference Types -** discusses different types of Arrays and String class. It then shows you the use of StringBuilder and StringTokenizer classes, command-line arguments, and wrapper classes.

Kup książkę

**Chapter 5: Subclasses and Interfaces -** discusses the concept of inheritance in Java in-depth and different ways of implementing inheritance by using abstract classes, nested classes, and so on. It then explains the interfaces and lambda expressions.

**Chapter 6: Exceptions and Regular Expressions -** describes exception handling with built-in exception classes and custom exceptions. It further introduces the important classes of java.lang and java.util.regex packages.

**Chapter 7: Collections and Stream API -** introduces the more advanced features such as Collections framework with the different utility classes and interfaces of the java.util package.

**Chapter 8: Generics and Time API -** describes the use of generics in Java to create generic classes, methods, and collections. It also introduces the Time API that provides better support for date and time.

**Chapter 9: File Manipulation in Java -** describes different types of streams of java.io package for file management. Further, it introduces the different classes of the java.util.zip and java.nio package.

**Chapter 10: Threads and JDBC -** explains the creation of thread and multithreading to improve the performance of applications. It then describes how to connect to databases by using JDBC API.

**Chapter 11: Design Patterns and Internationalization -** explains the use of design patterns as solutions to common problems encountered during software development. It also describes the internationalization and localization of an application.

**Chapter 12: More about JDK 8, 9 and 10 -** describes some prominent new features of Java 8 to 10, such as Java Platform Module System (JPMS), JShell, JLink tool, Local Variable Type Inference, and so on.

**Chapter 13: Java 11 (LTS) and New Updates -** introduces the new feature of Java 11 (LTS), such as string and file methods, HTTPClient, Garbage Collectors, etc. It also describes the different JDK providers and the new features launched in Java 12 to 16.

**Chapter 14: Java 17 (LTS) and New Updates -** explains Java 17 and the new features added to the JDK, such as pattern matching for switch. It also describes the new features and enhancements in Java versions 18 and 19.

Kup książkę

# Code Bundle and Coloured Images

Please follow the link to download the
***Code Bundle*** and the ***Coloured Images*** of the book:

**https://rebrand.ly/e3qa9xg**

The code bundle for the book is also hosted on GitHub at **https://github.com/bpbpublications/100Plus-Solutions-in-Java-2nd-Edition**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

# Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

**errata@bpbonline.com**

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.bpbonline.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at :

**business@bpbonline.com** for more details.

At **www.bpbonline.com**, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

Kup książkę

## Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

## If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

## Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

## Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

**https://discord.bpbonline.com**

Kup książkę

# Table of Contents

**1. Introduction to Java ..... 1**
Introduction ..... 1
Structure ..... 1
Objectives ..... 2
1.1 Introduction to object-oriented programming ..... 2
1.2 Java programming language ..... 4
*1.2.1 Features and advantages of Java ..... 4*
1.3 Java platform and editions ..... 5
1.4 Java SE platform components ..... 7
1.5 Java SE version history ..... 8
1.6 Features of Java SE 9 and Java SE 10 ..... 11
1.7 Download and install JDK 8 and JDK 10 ..... 15
1.8 Setting up the path for JDK ..... 19
1.9 ava application development ..... 20
1.10 Structure of a Java program ..... 21
1.11 Java code compilation and execution ..... 23
1.12 Creating and executing a Java program using JDK 8 ..... 24
1.13 Creating and executing a Java program using JDK 10 ..... 27
1.14 Creating and executing a Java 8 project in NetBeans ..... 30
1.15 Creating and executing a Java 10 project in NetBeans ..... 34
Conclusion ..... 44
Multiple choice questions ..... 44
*Answers ..... 45*
Assignment ..... 45

**2. Java Programming Constructs ..... 47**
Introduction ..... 47
Structure ..... 47
Objectives ..... 48
2.1 Java comments ..... 48
2.2 Variables in Java ..... 49

Kup książkę

2.3 Data types in Java ... 52
2.4 Type casting ... 56
2.5 Literals and escape sequences ... 56
2.6 Constants and enumerations ... 58
2.7 Operators in Java ... 59
2.8 Operator precedence and associativity ... 67
2.9 Formatting the output ... 68
2.10 Scanner class for input ... 70
2.11 Decision-making constructs ... 71
2.12 Looping constructs ... 77
2.13 Jump statements ... 84
Multiple choice questions ... 87
*Answers* ... *89*
Assignment ... 89

**3. Java Application Components ... 91**
Introduction ... 91
Structure ... 91
Objectives ... 92
3.1 Java class and objects ... 92
3.2 Access specifiers ... 93
3.3 Instance variables and methods ... 94
3.4 Constructor ... 97
3.5 Initializer block ... 98
3.6 Pass-by-value and pass-by-reference ... 99
*3.6.1 Passing arguments by value* ... *99*
*3.6.2 Passing arguments by reference* ... *100*
3.7 Variable argument method ... 101
3.8 Method overloading ... 102
3.9 Constructor overloading ... 104
3.10 'this' keyword ... 106
3.11 'final' keyword ... 107
3.12 'static' keyword ... 108
3.13 Packages ... 110
Conclusion ... 112

Kup książkę

Multiple choice questions....113
*Answers*....*114*
Assignment....114

**4. Java Reference Types....117**
Introduction....117
Structure....117
Objectives....118
4.1 Java arrays....118
*4.1.1 Single-dimensional array*....*118*
*4.1.2 Multi-dimensional array*....*120*
*4.1.3 Processing arrays with loops*....*123*
4.2 String class in Java....125
4.3 String arrays....128
4.4 StringBuilder class in Java....130
4.5 StringTokenizer class in Java....131
4.6 Passing arguments to main() method....133
4.7 Java wrapper classes....135
Conclusion....137
Multiple choice questions....137
*Answers*....*138*
Assignment....139

**5. Subclasses and Interfaces....141**
Introduction....141
Structure....141
Objectives....141
5.1 Inheritance in Java....142
*5.1.1 Creating subclass*....*143*
*5.1.2 Method overriding*....*145*
*5.1.3 Static versus dynamic binding*....*147*
5.2 Abstract methods and classes....149
5.3 Nested classes....151
5.4 Interfaces....157
*5.4.1 Implementing multiple interfaces*....*160*

Kup książkę

*5.4.2 Default, static, and private methods of interfaces* ..... *162*
*5.4.3 Issues with default methods* ..... *166*
5.5 Java functional interfaces ..... 167
5.6 Lambda expressions ..... 170
Conclusion ..... 175
Multiple choice questions ..... 175
*Answers* ..... *176*
Assignment ..... 176

**6. Exceptions and Regular Expressions ..... 179**
Introduction ..... 179
Structure ..... 179
Objectives ..... 180
6.1 Exceptions in Java ..... 180
*6.1.1 Types of exceptions* ..... *181*
6.2 Exception class in Java ..... 183
6.3 Exception handling in Java ..... 184
*6.3.1 'throw' and 'throws' keywords* ..... *187*
*6.3.2 Single catch block for multiple exceptions* ..... *189*
*6.3.3 Best practices for handling exceptions* ..... *190*
6.4 Using the try-with-resources statement ..... 190
6.5 Custom exceptions ..... 193
6.6 Wrapper exceptions ..... 194
6.7 Assertions in Java ..... 196
6.8 Classes of the java.lang package ..... 201
*6.8.1 Object class* ..... *202*
*6.8.2 Class class* ..... *204*
*6.8.3 Math class* ..... *205*
*6.8.4 ThreadGroup class* ..... *206*
*6.8.5 System class* ..... *207*
*6.8.6 Runtime class* ..... *208*
6.9 Regular expressions ..... 209
*6.9.1 Pattern class* ..... *209*
*6.9.2 Matcher class* ..... *211*

Kup książkę

*6.9.3 PatternSyntaxException* ..... *214*
6.10 Character classes ..... 214
Conclusion ..... 215
Multiple choice questions ..... 215
*Answers* ..... *217*
Assignment ..... 217

**7. Collections and Stream API ..... 219**
Introduction ..... 219
Structure ..... 219
Objectives ..... 220
7.1 Collections framework ..... 220
7.2 Iterable interface ..... 221
7.3 Collection interface ..... 221
*7.3.1 Traversing collections* ..... *223*
*7.3.2 Collection factory methods* ..... *223*
7.4 List interface ..... 224
*7.4.1 ArrayList class* ..... *225*
*7.4.2 Vector class* ..... *227*
*7.4.3 Stack class* ..... *228*
*7.4.4 LinkedList class* ..... *230*
7.5 Queue interface ..... 232
*7.5.1 Deque interface* ..... *233*
*7.5.2 ArrayDeque class* ..... *233*
*7.5.3 PriorityQueue class* ..... *235*
7.6 Set interface ..... 237
*7.6.1 HashSet class* ..... *239*
*7.6.2 LinkedHashSet class* ..... *239*
*7.6.3 SortedSet interface* ..... *241*
*7.6.4 NavigableSet interface* ..... *241*
*7.6.5 TreeSet class* ..... *242*
*7.6.6 ConcurrentSkipListSet* ..... *243*
7.7 Map interface ..... 245
*7.7.1 Hashtable class* ..... *247*

Kup książkę

*7.7.2 HashMap class* ..... *248*
*7.7.3 LinkedHashMap class* ..... *250*
*7.7.4 TreeMap class* ..... *252*
*7.7.5 ConcurrentSkipListMap* ..... *253*
7.8 Arrays class ..... 254
7.9 Sorting collections ..... 257
7.10 Java Stream API ..... 261
*7.10.1 Improvements of Stream API in Java 9* ..... *265*
Conclusion ..... 267
Multiple choice questions ..... 267
*Answers* ..... *268*
Assignment ..... 269

**8. Generics and Time API ..... 273**
Introduction ..... 273
Structure ..... 273
Objectives ..... 274
8.1 Generics ..... 274
8.2 Generic classes and methods ..... 275
8.3 Type inference ..... 278
8.4 Using generic constructors with generic and non-generic classes ..... 278
8.5 Using generics with collections ..... 279
8.6 Using wildcards with generics ..... 280
8.7 Using generics with exceptions ..... 283
8.8 Implementing generics with inheritance ..... 284
8.9 Type erasure ..... 285
8.10 Time API ..... 286
*8.10.1 ZonedDateTime API* ..... *287*
*8.10.2 ChronoUnit Enum* ..... *288*
*8.10.3 Period and Duration classes* ..... *289*
*8.10.4 TemporalAdjusters class* ..... *291*
*8.10.5 Backward compatibility* ..... *292*
Conclusion ..... 293
Multiple choice questions ..... 293

Kup książkę

*Answers* ..... 294
Assignment ..... 295

**9. File Manipulation in Java ..... 297**
Introduction ..... 297
Structure ..... 297
Objectives ..... 298
9.1 Files and streams ..... 298
*9.1.1 File class* ..... 299
*9.1.2 FileDescriptor class* ..... 301
9.2 DataInput and DataOutput interfaces ..... 302
9.3 FilenameFilter interface ..... 303
9.4 Byte streams ..... 304
*9.4.1 InputStream class hierarchy* ..... 304
*9.4.1.1 FileInputStream class* ..... 305
*9.4.1.2 ByteArrayInputStream class* ..... 306
*9.4.1.3 FilterInputStream class* ..... 307
*9.4.1.4 BufferedInputStream* ..... 307
*9.4.1.5 ObjectInputStream class* ..... 309
*9.4.2 OutputStream class hierarchy* ..... 310
*9.4.2.1 FileOutputStream* ..... 311
*9.4.2.2 ByteArrayOutputStream class* ..... 312
*9.4.2.3 FilterOutputStream class* ..... 313
*9.4.2.4 BufferedOutputStream class* ..... 315
*9.4.2.5 ObjectOutputStream class* ..... 316
9.5 Serialization ..... 318
9.6 Character streams ..... 318
*9.6.1 Reader class hierarchy* ..... 318
*9.6.1.1 CharArrayReader class* ..... 319
*9.6.1.2 FileReader class* ..... 321
*9.6.2 Writer class hierarchy* ..... 322
*9.6.2.1 PrintWriter class* ..... 323
*9.6.2.2 CharArrayWriter class* ..... 325
9.7 Console class ..... 326

Kup książkę

9.8 java.util.zip package ..... 328
*9.8.1 Deflater class* ..... *328*
*9.8.2 Inflater class* ..... *329*
*9.8.3 DeflaterInputStream class* ..... *331*
*9.8.4 DeflaterOutputStream class* ..... *333*
*9.8.5 InflaterInputStream class* ..... *334*
*9.8.6 InflaterOutputStream class* ..... *336*
9.9 java.nio package ..... 338
*9.9.1 Path interface* ..... *339*
*9.9.2 Files class* ..... *341*
*9.9.2.1 List the contents of a directory* ..... *341*
*9.9.2.2 Create directories and files* ..... *342*
*9.9.2.3 Check the existence of a file or directory* ..... *342*
*9.9.2.4 Read and write operation on files* ..... *343*
*9.9.2.5 Copy a file or directory* ..... *345*
*9.9.2.6 Move a file or directory* ..... *347*
*9.9.2.7 Delete a file or directory* ..... *348*
*9.9.2.8 Randomly access a file* ..... *349*
*9.9.3 FileSystem class* ..... *351*
*9.9.4 Watch service* ..... *351*
*9.9.5 PathMatcher interface* ..... *352*
Conclusion ..... 356
Multiple choice questions ..... 356
*Answers* ..... *357*
Assignment ..... 357

**10. Threads and JDBC ..... 359**
Introduction ..... 359
Structure ..... 359
Objectives ..... 360
10.1 Threads ..... 360
*10.1.1 Java thread states and lifecycle* ..... *360*
10.2 Thread class ..... 362
*10.2.1 Methods of the Thread class* ..... *364*

Kup książkę

*10.2.2 Thread priority*....365
*10.2.3 Methods for thread priority*....366
10.3 Runnable interface....368
10.4 Daemon threads....369
10.5 Multithreading....371
*10.5.1 isAlive() method*....373
*10.5.2 join() method*....374
10.6 Thread synchronization....375
*10.6.1 Synchronized block*....375
*10.6.2 Synchronized method*....377
*10.6.3 Wait-notify mechanism*....379
*10.6.4 Deadlock*....382
10.7 Concurrent collection APIs....386
10.8 Atomic variables....389
*10.8.1 java.util.concurrent.atomic package*....390
10.9java.util.concurrent.locks package....393
10.10 Executors and thread pools....394
*10.10.1 Thread pools*....395
*10.10.2 ForkJoinPool*....398
10.11 Java Database Connectivity (JDBC)....402
*10.11.1 JDBC API classes and interfaces*....404
*10.11.2 Connecting to a database from a Java program*....405
*10.11.3 Parameterized queries*....408
*10.11.4 Manage SQL exceptions*....409
*10.11.5 Connect to a database with Type 4 driver*....410
*10.11.6 Execute queries with Statement object*....410
*10.11.7 Execute queries with PreparedStatement object*....413
10.12 DatabaseMetaData....415
10.13 ResultSetMetaData....417
10.14 Execute stored procedure with CallableStatement object....418
*10.14.1 CallableStatement interface*....420
10.15 Scrollable ResultSet....424
*10.15.1 Insert, update, delete operations on a ResultSet object*....426
10.16 Batch updates....430

Kup książkę

10.17 Transactions ..... 435
10.18 RowSet interface ..... 438
*10.18.1 Types of RowSets* ..... *439*
*10.18.2 Connected RowSets* ..... *440*
*10.18.3 Disconnected RowSets* ..... *444*
Conclusion ..... 447
Multiple choice questions ..... 447
*Answers* ..... *448*
Assignment ..... 449

**11. Design Patterns and Internationalization ..... 451**
Introduction ..... 451
Structure ..... 451
Objectives ..... 452
11.1 Design patterns and polymorphism ..... 452
*11.1.1 instanceOf operator* ..... *454*
11.2 Design patterns ..... 456
*11.2.1 Creational patterns* ..... *457*
*11.2.1.1 Singleton pattern* ..... *457*
*11.2.1.2 Factory pattern* ..... *458*
11.3 Structural patterns ..... 460
*11.3.1 Data Access Object (DAO) pattern* ..... *460*
11.4 Behavioral patterns ..... 464
*11.4.1 Observer pattern* ..... *464*
11.5 Other design concepts ..... 468
*11.5.1 Delegation* ..... *469*
*11.5.2 Composition and aggregation* ..... *470*
11.6 Internationalization and localization ..... 471
*11.6.1 ISO Codes* ..... *472*
*11.6.2 Unicode* ..... *472*
11.7 Implementing internationalization and localization ..... 472
11.8 Internationalization elements ..... 476
*11.8.1 Formatting number* ..... *476*
*11.8.2 Formatting percentage* ..... *478*
*11.8.3 Formatting currency* ..... *479*

Kup książkę

*11.8.4 Formatting date* ..... *481*
*11.8.5 Formatting messages* ..... *482*
Conclusion ..... 485
Multiple choice questions ..... 485
*Answers* ..... *486*
Assignment ..... 486

**12. More about JDK 8, 9, and 10 ..... 489**
Introduction ..... 489
Structure ..... 489
Objectives ..... 489
12.1 Features of Java 8 ..... 490
12.2 Features of Java 9 ..... 495
12.3 Features of Java 10 ..... 511
Conclusion ..... 517
Multiple choice questions ..... 517
*Answers* ..... *518*
Assignment ..... 518

**13. Java 11 (LTS) and New Updates ..... 521**
Introduction ..... 521
Structure ..... 521
Objectives ..... 521
13.1 Introduction to Java 11 (LTS) ..... 522
*13.1.1 Oracle versus OpenJDK* ..... *523*
13.2 New features of Java 11 (LTS) ..... 524
*13.2.1 New String class methods* ..... *524*
*13.2.2 New Files class methods* ..... *526*
*13.2.3 Collection to an array* ..... *526*
*13.2.4 The static 'not' method* ..... *527*
*13.2.5 Local-variable type inference for lambda parameters* ..... *529*
*13.2.6 HttpClient* ..... *530*
*13.2.7 Run source files* ..... *531*
*13.2.8 No-Op garbage collector* ..... *531*
*13.2.9 Flight Recorder* ..... *531*

Kup książkę

*13.2.10 Removed and deprecated modules* ..... *532*
13.3 Features of Java versions 12 to 16 ..... 532
*13.3.1 Java 12* ..... *532*
*13.3.2 Java 13* ..... *533*
*13.3.3 Java 14* ..... *534*
*13.3.4 Java 15* ..... *536*
*13.3.5 Java 16* ..... *537*
Conclusion ..... 538
Multiple choice questions ..... 539
*Answers* ..... *540*
Assignment ..... 540

**14. Java 17 (LTS) and New Updates ..... 541**
Introduction ..... 541
Structure ..... 541
Objectives ..... 541
14.1 Introduction to Java 17 (LTS) ..... 542
*14.1.1 Pattern matching for switch (Preview)* ..... *542*
*14.1.2 Sealed classes (Finalized)* ..... *545*
*14.1.3 Foreign function and Memory API - Incubator* ..... *545*
*14.1.4 Security manager deprecated* ..... *546*
14.2 Features of Java 18 and Java 19 ..... 546
*14.2.1 Features of Java 18* ..... *546*
*14.2.2 Features of Java 19* ..... *546*
Conclusion ..... 546
Multiple choice questions ..... 546
*Answers* ..... *547*
Assignment ..... 547

**Index ..... 549-560**

Kup książkę

Kup książkę

# CHAPTER 1
# Introduction to Java

## Introduction

This chapter introduces the concept of object-oriented programming and explains how Java has evolved as an **Object-Oriented Programming (OOP**) language. You will learn about the various versions and features of Java and the steps to install a **Java Development Kit (JDK**). You will also learn to create an application by using JDK 8 and JDK 10.

## Structure

Following are the topics that are covered in this chapter:

- Introduction to object-oriented programming
- Java programming language
- Java platform and editions
- Java SE platform components
- Java SE version history
- Features of Java SE 9 and Java SE 10
- Download and install JDK 8 and JDK 10
- Setting up the path for JDK
- Java application development

Kup książkę

- Structure of a Java program
- Java code compilation and execution
- Create and execute a Java program using JDK 8
- Create and execute a Java program using JDK 10
- Creating and executing a Java 8 project in NetBeans
- Creating and executing a Java 10 project in NetBeans

# Objectives

In this chapter, you will learn the concept of object-oriented programming. You will also learn to download and install JDK 8 and JDK 10. Finally, you will understand the structure of a Java program and learn to develop a Java project in NetBeans. You will learn to download and install JDK 8 and JDK 10

# 1.1 Introduction to object-oriented programming

With the advancement in technology and the increasing complexity of software, a requirement for new and flexible modes of programming was observed. A need to make reliable software and reduce the overall development and maintenance cost, and deliver completed software on decided timelines, resulted in the development of the object-oriented programming model.

The primary focus of object-oriented programming is on objects. Any real-world entity that has certain characteristics and behavior that can be used to describe it is considered as an object. There are several objects that have certain common characteristics. These can be grouped into categories or classes. Thereby, every object of a class will be considered as an instance of that class. Programmatically, a class is a structure that contains the data (characteristics) and methods (behavior) to work on that data.

For example, a class **`Vehicle`** can have characteristics such as color, type, and behavior such as start, stop, accelerate, and so on. The following image shows a **Unified Modeling Language** (**UML**) class diagram representation of the Vehicle class:

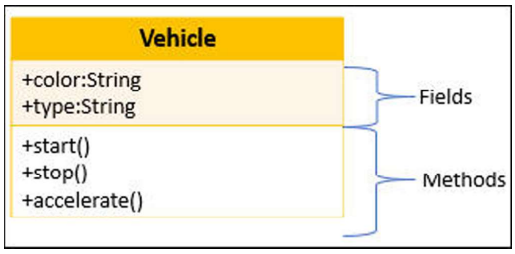


***Figure 1.1:*** *Class diagram*

Kup książkę

Here, the **Fields** represent the characteristics, and **Methods** represent the behavior of the object. The Vehicle class can then have instances of type Vehicle, such as bike, bicycle, car, and so on. This is explained in detail in the following figure:

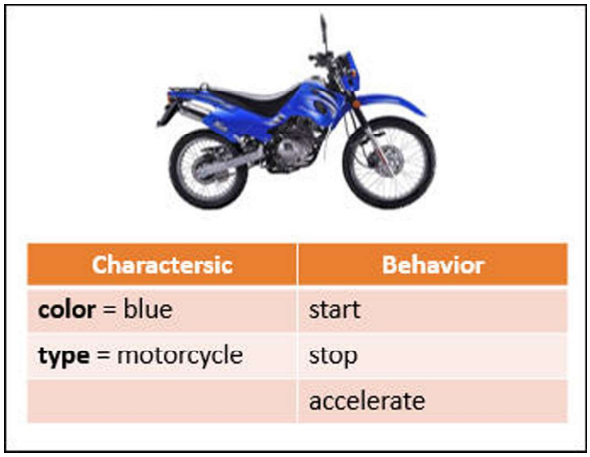


***Figure 1.2:*** *Object of Vehicle class*

Here, the object bike has the characteristics **color=blue** and **type=motorcycle** with the behavior including start, stop, and accelerate. Similarly, there can be other instances of class Vehicle such as car, bicycle, and so on. with the same or different values for characteristics and similar behavior.

An object-oriented programming language is based on the following principles:

- **Encapsulation**: To encapsulate means to enclose. Hence, encapsulation allows enclosing the data members and methods into a closed structure called a class. Encapsulation ensures data security through data hiding so that a user cannot access the members of a class directly.
- **Abstraction**: Abstraction is a programming concept in which the non-essential details of an entity are hidden from user view. For example, in the case of a washing machine, the user only presses the button on a digital panel to set up the process and start the machine. However, the internal functioning of the washing machine is not known to the user. This means that the non-essential aspect of how the washing machine washes the clothes is abstracted from the user. Similarly, abstraction can also be implemented in code to hide the unnecessary details from the user.
- **Inheritance**: To inherit means to acquire some feature or asset from an ancestor. For example, a child acquires certain aspects of physical appearance and certain behavior of his/her biological parents. In programming also, inheritance plays a similar role. It allows us to combine the common characteristics and behavior of objects into a parent class also called a superclass. This class can then be inherited by other classes that allow a developer to extend and reuse the feature of existing classes. The new/inherited classes are called child classes, derived classes, or subclasses.

Kup książkę

- **Polymorphism**: Polymorph is a combination of words *poly* which means many and *morph* which means forms. This polymorph is an object that can have multiple forms/ behavior. For example, a chameleon can change its color as per the environment to protect itself from predators. In programming, polymorphism is the ability of an object to behave in different ways based on requirements. Polymorphism can be implemented in several ways in programming based on the programming language used.

# 1.2 Java programming language

Java is a popular object-oriented, platform-independent programming language. It allows the development of a variety of applications that can run on different hardware and operating systems. Java also provides a runtime environment for executing Java applications on different devices.

Java was originally developed in 1991 by *James Gosling* and a team of engineers at *Sun Microsystems* which was later acquired by *Oracle Corporation*. It was initially designed for consumer devices such as washing machines, television, and so on. For such devices, it was necessary to have a language that was small, efficient, fast, and platform-independent. Languages such as C and C++ were not preferred due to the compiler's dependence on specific CPUs and also high development time and cost. Thus, Java was developed as a portable and platform-independent language that could execute codes on any platform. Initially, it was named *Oak* but later renamed to Java.

Even though Java was developed to cater to the programming needs for smaller devices, it was found to be able to address larger problems including Web and mobile applications. It gained instant popularity and was adopted all over the world for the development of applications ranging from embedded, desktop, Web, and mobile applications. Java can be used to create applications for small to large businesses and even for supercomputers.

## 1.2.1 Features and advantages of Java

Following are some features and advantages of Java programming language:

- **Simple and robust:** Java syntax is derived from its predecessor programming languages like C, C++. This makes it easy for developers to learn Java quickly. Further, the complexity of pointers, operator overloading, multiple inheritances and other such features has been removed in Java. Instead, it has been made more robust through efficient memory management and exception handling features.
- **Object-oriented:** Java is based on the object-oriented programming paradigm. Thereby, it is well suited for the development of real-world applications.
- **Platform independent**: Java provides a solution to a major problem faced by earlier languages, that is, code portability. During compilation, it converts the source code into an intermediate, architecture-neutral format called bytecode. This bytecode can be executed on any platform which has a **Java Virtual Machine** (**JVM**) installed. Further, even the language specifications, such as the size of primitive data types and operators,

Kup książkę

have been defined to be independent of the hardware. This ensures that the code will function properly in case of a change in the operating system, processor, or system resources.

- **Secure:** Security is an important issue in Java applications since they are designed for multiple and distributed platforms. Java provides security checks at different levels during application development. The JVM is designed to apply its security features during code execution to ensure that the code is well-formed and written as per Java standards.
- **Multithreaded**: Java supports the development of multithreaded applications to perform multiple tasks concurrently. In a multithreaded application, a single program can have multiple threads performing tasks independently and concurrently. Java allows creating thread pools that can be used to obtain threads when required.
- **Distributed and dynamic**: Java supports distributed programming to deploy and access resources across the network. It provides several **Application Programming Interfaces** (**APIs**) to handle remote transmission and requests over a network. Java also allows dynamic execution of classes by storing them in a separate location and loading the necessary classes dynamically at runtime.
- **Modular**: The concept of modularity has been introduced since Java 9. It was supposed to be incorporated in Java 7 and Java 8 but was not accomplished. Until Java 1.8, the packages were bundled into JAR files that were the final executable for a Java application. But, with Java 9, a new construct called *Module* has been introduced. A module is similar to the JAR file but unlike the JAR file, it also contains configuration information in the form of a `module-info.java` file. This allows a module to be more powerful and flexible as compared to a JAR file since all dependencies are specified in the `module-info.java` file. While using a JAR, the entire JAR is loaded during application execution, but with the module, only those modules that are part of the dependency list will be loaded. This allows applications to remain light weight as well as execute faster.

# 1.3 Java platform and editions

The Java platform is a development and execution environment for Java applications which is a mixture of software and hardware components. Following image shows the components of a Java platform:

Kup książkę

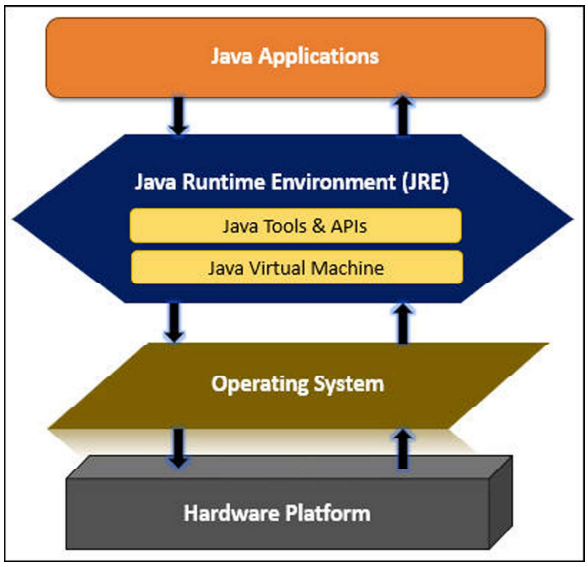


*Figure 1.3: Java platform*

The above figure depicts the components of a Java platform. It is formed of the **Java Runtime Environment** (**JRE**) which contains the Java library (Java API) and **Java Virtual Machine** (**JVM**). In languages such as C and C++, the compiled code is platform dependent as it is in the form of executable binary code. However, the Java compiler converts the code into an intermediate bytecode which is an optimized set of instructions that can be executed on any machine that has the appropriate JVM. Thus, JVM provides platform independence to Java code. Each operating system such as Windows, Linux, Mac, and so on. has its JVM. Thus, Java code follows the principle of write-once use many.

The Java API/library is a collection of ready-to-use components such as calluses and interfaces, that can be used to create applications. For example, the Swing library is used to create a **User Interface** (**UI**) of a Java application. Java is released under several editions to meet the requirements of a specific type of device and application. Following is a brief description of the different Java Editions:

- **Java Standard Edition (Java SE)**: Java SE is the base for creating applications that are console or network-based applications, mainly for desktop computers. It contains the core APIs, including the basic types and classes for higher-level programming such as networking, security, **Graphical User Interface** (**GUI**), database manipulation, and parsing of XML data. It also provides the virtual machine, development and deployment tools, and other toolkits and libraries for advanced programming.
- **Java Enterprise Edition (Java EE)**: The Java EE platform is an extension of the Java SE platform. It provides the tools and APIs for the creation and deployment of large-scale, distributed, scalable, multi-tier, reliable, and secure enterprise applications with complex business logic.

Kup książkę