Code with Java 21

A practical approach for building robust and efficient applications

Aaron Ploetz

www.bpbonline.com

First published: 2024

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

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55519-993

www.bpbonline.com

Dedicated to

My children:
Khadiya**, **Avery**, **Emily**, and **Victoria
&
My godson:
Tim

The road to success
is paved with the footsteps of those who did not give up
and the regrets of those who did.

About the Author

Aaron Ploetz is a developer advocate at *DataStax*. He has been a professional software developer since 1997 and has a successful history of leading DBA and DevOps teams for both startups and Fortune 50 enterprises. He has been named an MVP for Apache Cassandra® three times, and has presented at multiple events; including the Fintech keynote at *Data Day Mexico City 2023*. Aaron frequently answers StackOverflow questions from other developers, and has previously authored books on distributed databases. He earned a B.S. in Management/Computer Systems from the *University of Wisconsin-Whitewater*, and an M.S. in Software Engineering (Database Technology emphasis) from *Regis University*. When not writing or coding, Aaron enjoys fishing, retro video gaming, and skijoring. Aaron and his wife, Coriene, live with their four children in the Twin Cities area.

About the Reviewer

Otavio Santana is a passionate architect and software engineer focused on cloud and Java technologies. He has deep expertise in polyglot persistence and high-performance applications in finance, social media, and e-commerce.

He has contributed to the Java and the open source ecosystem in several ways. Otavio has helped in the direction and objectives of the Java platform since Java 8 as a JCP executive member, besides being a committer and leader in several open-source products and specifications.

He is recognized for his open source contributions and has received many awards, including all JCP Awards categories and the Duke's Choice Award, to name a few. Otávio is also a distinguished Java Champions and Oracle ACE program member.

Acknowledgement

I would like to recognize my co-workers, Cédrick Lunven and Mary Grygleski, who were extremely helpful and central in influencing me to get back into Java. I would also like to recognize Otavio Santana, Sharat Chander, and countless others who made the Java community so welcoming and encouraging.

I would also like to thank Ursula Kellmann, who was taken from us too soon. She was an amazing mentor who taught me much about Java and was a brilliant example of what a true expert in their craft should aspire to be.

Finally, I would love to acknowledge my wife, Coriene, who continually inspires me to take on new challenges and to be a better version of myself.

Preface

Learning software development has always been a treacherous path. When I began my journey, books were the only medium available. When I started learning to program BASIC on my Tandy 1000, I was first introduced to Java during my final year at the *University of Wisconsin–Whitewater (1998)*. I loved the *code once*, *execute everywhere* message that Java trumpeted. In the 1990s, that aspect was a really big deal.

In the early 2010s, I wrote a lot of Java code while working on the *Mid-Tier Cassandra* team at *W.W. Grainger*. This was my first introduction to enterprise-level Java. The database behind our service layer (Apache Cassandra®) was also written in Java, so I was quite literally *thrown into the deep end* of troubleshooting the exceptions and nuances of Java 7.

After spending the better part of the next decade as a Cassandra DBA, I finally returned to (near) full-time Java coding in 2021. It was a breath of fresh air. I loved how much Java had evolved up through version 17. By that time, I had written two books about NoSQL databases. The more I worked with Java, the more I wanted to write a book about it.

There is so much Java code running in the world today. It runs across billions of devices and powers everything from video games to e-commerce websites that generate billions of dollars. There is no denying that knowing how to code in Java has become a valuable skill.

Today, one of the biggest obstacles for aspiring developers (aside from choosing a language) is getting a development environment installed and configured. By contrast, many of the computers that were common in the 1980s made it very easy to get started programming. Users were often only a few key presses away from getting into a software development environment.

Some early home computers (like the Tandy Color Computers and Apple II series) put the user at a BASIC programming prompt immediately after booting. These early machines were asking to be programmed by default! On the other hand, modern computers (especially phones and tablets) do not. Building a suitable programming environment on modern computers is often a struggle. This is why the first chapter of this book spends time on installing and configuring the Eclipse IDE.

However, making computers programmable again is more than just navigating technical obstacles. It is a culture shift. It is about looking at your device, and instead of asking, *What can it do for me*, it is asking, *What can I build with it?* It is this mindset that leads to a lifetime of learning.

Ultimately, my motivation for writing this book is to help flatten the learning curve for software developers. I believe that Java is the right language to do just that.

This book is structured to gradually introduce different aspects of writing code in Java, with each chapter building on lessons previously covered.

Chapter 1: Getting to Know Java – This chapter provides a simple introduction to Java. It also walks through configuring a development environment, including installing tools such as a dependency manager and an IDE. This chapter also discusses **object-oriented programming (OOP)** and introduces new features of Java 21.

Chapter 2: Fundamental Programming Structures – This chapter discusses some basic building blocks of Java's syntax. It starts with the obligatory hello world program and progresses to reading input, error handling, and controlling the flow of program logic. After covering the basics, it moves on to working with files and Java-specific ways of building methods and constructors.

Chapter 3: Strings, Characters, and Regular Expressions – This chapter covers different ways to work with and process text data. While starting with a simple example using ASCII character art, it progresses into showing how to use some of the more advanced methods included with the Java String class. It finishes by introducing regular expressions and showing how to use them in practical, real-world examples.

Chapter 4: Arrays, Collections, and Records – This chapter introduces the reader to different structures that can be used to store data in memory. While focusing on different concepts and use cases with arrays, lists, sets, and maps, this chapter also discusses records and introduces sequenced collections (new with Java 21).

Chapter 5: Arithmetic Operations – This chapter offers insight into how computers handle arithmetic at primitive levels, including the difference between integer and floating point operations. It also uses the deterministic nature of arithmetic to make a small introduction to unit testing.

Chapter 6: Common Data Structures – This chapter takes the reader through the process of using Java to build data structures like stacks, queues, and different kinds of linked lists. It also walks through building a binary tree and executing simple data searches.

Chapter 7: Working with Databases – This chapter shows how to build Java applications that store data in databases. It also introduces the PostgreSQL and Apache Cassandra® databases, showing how to build simple data models and execute common CQL and SQL commands.

Chapter 8: Web Applications – This chapter discusses building restful web services and full-stack web applications in Java. It also introduces the Spring and Vaadin frameworks, showing how to leverage them to build fully-featured web applications.

Chapter 9: Graphics in Java – This chapter uses an example-led approach to display graphics and animation. It also walks through building a classic arcade game in Java.

Chapter 10: Final Java Project – This chapter is a culmination of many topics covered in the previous chapters and shows how to use them to build a Movie data application. Vector search is introduced as a final new topic, giving the readers a simple way to build out a movie recommendation service.

Code Bundle and Coloured Images

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

https://rebrand.ly/nkskce0

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/Code-with-Java-21
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.

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

Table of Contents

Chapter 1
Getting to Know Java

Introduction

Welcome to **Code with Java 21**! Whether you are a new or experienced programmer, this book will help you to understand and effectively wield one of the most widely used programming languages in the world. In addition to covering the fundamental aspects of Java, we will also discuss the new features delivered in Java version 21 and show you how to use them effectively.

Java 21 is a **Long-Term Support** (**LTS**) release with support through 2031. This book is designed not only to help you learn, but also to be a reference for the long term. It contains code designed to guide you through each example and help you become a successful programmer.

Note: The terms programmer, software developer, and coder are often used interchangeably.

Structure

In this chapter, we will discuss the following topics:

- Advantages of building applications with Java.
- Examine the different components of a Java development environment

- Discuss common tools used to compliment the Java development process
- Introduce the principles of object-oriented programming
- Preview the new features in Java 21

Objectives

The goal of this book is to inspire you to build the next generation of Java applications. In this chapter, we will examine the Java language at a high level, aiming to provide enough detail to get started. By the end of the chapter, we will understand what makes Java different from other languages and how to leverage it to write powerful applications.

Why code with Java?

Java is everywhere; it is running on billions of devices around the world. It is also used by Fortune 500 enterprises to build services and applications that help them make billions of dollars each year. Needless to say, there is a high demand for Java developers, and it is likely to continue for a long time.

There are also several types of machines capable of running Java, including (but not limited to) the following:

- **Personal Computers** (**PCs**) for both home and business use
- Mobile devices
- Gaming consoles
- Embedded devices

Java's core properties of platform independence, versatility, and security have made it one of the most popular programming languages in the world. They also make it easy to get help with, as learning material for Java can be quickly found on *YouTube*, *LinkedIn*, and many other websites.

Whether you are interested in learning to program as a hobby or as a skill that can lead to a successful career, Java is a great skill to have.

Configuring your environment

Before we can begin writing Java programs, we need to ensure that our environment is properly built and configured. Here are the things we will need to be successful:

- A computer running Windows, Linux, or MacOS
- A **Java Runtime Environment** (**JRE**)
- An **Integrated Development Environment** (**IDE**)
- A Java dependency manager

- A source control platform

While this book makes some accommodations for developers new to Java, it is intended for those who have at least an intermediate level of overall programming experience. While an overview of configuring a development environment will be provided, exhaustive detail on every possible configuration is beyond the scope of this book. It is assumed that the reader will install and configure the necessary tools that are most familiar to them.

Operating System

One of the main advantages of Java is that it is easily portable. That means the same Java code can run on Windows, Linux, or MacOS without any changes. Likewise, it does not matter which **Operating System** (**OS**) platform the Java code is written on. As a programmer, it is important for you to know your OS well and to understand its nuances and differences from other OSs when appropriate.

For example, it is important to remember that Windows does not care about uppercase characters in filenames, while Linux and MacOS do. Windows also has different file line endings than Linux and MacOS. These things can pose challenges when building applications that work with files and other OS-level aspects.

Java Runtime Environment

Another part of the development environment that is required for Java is the Java Runtime Environment. This package provides all the available libraries required for your Java code to run. This book is written to focus on Java 21, which is the version of Java that should be installed to get the examples in this book to run properly.

Java Development Kit

It is also important to remember that downloads are available for both the JRE and the **Java Development Kit** (**JDK**). While the JRE provides a complete environment for Java programs to run, the JDK provides *both* a JRE and additional tools for developers to build and configure Java programs. As we will need the extra development tools, a JDK is required to follow the examples put forth in this book.

JDK vendors and editions

There are several software companies that build their JDKs, including *Microsoft*, *Oracle*, and *IBM*. Their builds of the JDK are usually intended for corporate use, and most require a paid license or contract to use.

Many vendors also produce different editions based on the intended uses and underlying infrastructures:

- **Micro Edition:** A smaller build of the JDK, intended for embedded systems and other devices with a smaller amount of compute resources.
- **Standard Edition:** A middle-tiered build focusing on developer machines and workstation-grade hardware.
- **Enterprise Edition:** A full-fledged build targeting enterprises and high-throughput systems.

For this book, we will use the OpenJDK, a free and open-source version of the Java Development Kit- Standard Edition. The latest versions of the OpenJDK (version 21) for various Operating Systems and architectures can be found at **https://jdk.java.net/21/.**

Installation

You can skip this step if your IDE installation comes with a JDK. Otherwise, OpenJDK downloads come as a compressed file; usually a tarball or a ZIP file. The location that the download needs to be uncompressed to differs by operating system. However, it needs to be put into a location *pathed-in* to the environment.

Note: You will likely need administrator or super user permissions to install a JDK.

You can run this command to verify your JDK installation or to see the version you have installed:

```
java -version
```

If there is already a JRE or JDK installed, you should see an output similar to this:

```
openjdk version "21-ea" 2023-09-19

OpenJDK Runtime Environment (build 21-ea+16-1326)

OpenJDK 64-Bit Server VM (build 21-ea+16-1326, mixed mode, sharing)
```

As the focus of this book is Java 21, the major version listed will need to be **21**.

Windows

The standard location for the JDK to reside is in the **Program Files** directory. Set the **`JAVA_HOME`** environment variable to that location. Additionally, you may need to add it to the **`PATH`** environment variable.

MacOS

The same approach can certainly be taken on a Mac. After uncompressing the tarball, add its location to the **`PATH`** environment variable (in the **`.bashrc`** file).

Additionally, there is a Homebrew formula available for the OpenJDK, which takes care of the install and environment variable config. It can be installed from the terminal as follows:

```
brew install openjdk@21
```

Linux

Likewise, the Linux tarball can be uncompressed and location-referenced via the **`PATH`** variable in the **`.bashrc`** file. Additionally, the delivered Linux package managers can also access the required OpenJDK repositories. The exact command used depends on the flavor of Linux.

If you are running on a Red Hat Linux derivative (for example, *Fedora, CentOS*), the OpenJDK can be installed with the **`yum`** package manager:

```
sudo yum install java-21-openjdk
```

Additionally, for those of you running on a Debian Linux derivative (for example, *Ubuntu, Cinnamon*), the OpenJDK can be installed with the **`apt`** package manager:

```
sudo apt install openjdk-21-jdk
```

It is important to note that if you have multiple JDKs/JREs installed, you may need to change your default version. This can be done by updating the system alternatives:

```
sudo update-alternatives --config java
```

Version management

Some developers may have multiple JREs/JDKs installed on their developer workstations. It is highly recommended that you use a Java environment manager. For example, MacOS and Linux users can install a tool like **jEnv** by heading to this website: **https://www.jenv.be/**.

There is also a jEnv for Windows, available in the following GitHub repository: **https://github.com/FelixSelter/JEnv-for-Windows**.

Integrated Development Environment

Before you can write code in any language (including Java), you will need a special tool. At the very minimum, a text editor like *Notepad*, *Sublime*, or *Vim* is required. However, most developers prefer using an Integrated Development Environment.

An IDE is more than just a code editor and gives the programmer access to tools designed to make writing code easier. Usually, this allows them to easily and quickly build and compile their code, interact with source control, select a different JDK, and set specific