Building Kotlin Applications

A comprehensive guide for Android, Web, and Server-Side Development

Mounir Boussetta

www.bpbonline.com

First published: 2024

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

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-5551-633-6

www.bpbonline.com

Dedicated to

My daughter:

Inas

&

My son ***Saad***

About the Author

Mounir Boussetta a highly accomplished and dedicated software engineer with a passion for the ever-evolving world of technology. As a co-founder and a project manager at FIRETHUNDER, a leading company in medical information systems, Mounir brings an exceptional level of expertise and leadership to his role.

Mounir's journey in the IT industry began after he graduated with a degree in electronics and telecoms engineering in 2013. Driven by his love for programming, he transitioned into the IT domain, where he has thrived ever since. With over 13 years of experience in Java development, starting from version 1.6, and 6 years of experience with Kotlin, Mounir has an extensive understanding of the language and its applications.

Prior to his role at FIRETHUNDER, Mounir worked as a software developer at Vivace Consulting SARL, where he honed his skills in AngularJS/Angular, Java, Spring Boot, Spring Data, Spring Security, Hibernate, and REST APIs. His proficiency in IHE Framework standards such as DICOM 3.0 and HL7, which he gained through five years of experience, has been instrumental in his success in the field.

About the Reviewer

Devnath Jha is a seasoned Android developer with nearly a decade of experience in both Java and Kotlin.

In addition to his extensive work in the Android realm, Devnath has embarked on an exciting journey in the world of Java backend development with Spring Boot and MongoDB.

He is currently contributing his expertise to a forward-thinking IT startup company based in Bangalore.

His passion for technology extends beyond the confines of his workspace.

When he isn't immersed in code, he dedicates his time to staying updated with the latest advancements in both Android and backend technologies.

Devnath Jha's unique blend of Android expertise, coupled with his burgeoning backend development skills, positions him as a versatile and invaluable asset in the realm of software engineering.

Acknowledgement

I wish to extend my profound appreciation to my family and friends for their consistent support and motivation throughout the process of writing this book.

Additionally, I'd like to express my thanks to BPB Publications for their invaluable guidance and expertise in bringing this book to fruition. The journey of revising this book was a lengthy one, and I am immensely thankful for the valuable contributions and collaboration of reviewers, technical experts, and editors.

I would also like to acknowledge the significant contributions of my colleagues and coworkers from my years of working in the tech industry. They have not only taught me a great deal but have also provided valuable feedback on my work.

Lastly, I want to convey my gratitude to all the readers who have shown interest in my book and supported its realization. Your encouragement has been truly priceless.

Preface

Building applications is a multifaceted endeavor that demands a comprehensive grasp of cutting-edge technologies and programming languages. Kotlin has emerged as a powerful and increasingly popular tool in the realm of software development.

This book is meticulously crafted to serve as your all-encompassing guide to constructing enterprise applications with Kotlin. It delves into a vast array of topics, beginning with the fundamentals of Kotlin programming, progressing to advanced concepts like object-oriented programming, and culminating in the utilization of the Kotlin ecosystem to forge robust and scalable applications.

As you traverse the pages of this book, you will immerse yourself in the essential attributes of Kotlin and its ecosystem, equipping you to fashion enterprise applications that are not only efficient and reliable but also easily maintainable. You will glean insights into best practices and design patterns tailored to the unique demands of enterprise application development, and you will encounter a plethora of real-world examples that will solidify your comprehension of these concepts.

Whether you are a newcomer to Kotlin and the world of enterprise application development or an experienced developer seeking to fortify your command of these technologies and elevate your proficiency in crafting resilient and dependable applications, this book is your compass on this journey.

With this book as your companion, you will acquire the knowledge and expertise needed to flourish as an adept developer in the sphere of multiplatform applications development using Kotlin. I sincerely hope that you discover this book to be enlightening and invaluable on your path to mastery.

Chapter 1: Java and Kotlin - In this first chapter, we will put you in the context of the language, and you will be introduced to the history of Kotlin, why this new programming language has appeared, who created Kotlin, and what was their intention behind Kotlin. We will also learn about Kotlin in comparison to Java; we will learn about the advantages and disadvantages of each and what makes the Kotlin language the right choice for your future development and maybe your career.

Chapter 2: Kotlin Basics - The readers will get an understanding of the basics of the Kotlin programming language. They will be able to deal with variables, data types, the creation of

functions, the basic control structures, and also how to work with some Kotlin features like smart casting, class extensions, and operator overloading.

Chapter 3: OOP with Kotlin – This presents how to create some basic real-world applications in Kotlin by following an object-oriented way, and this goal will be possible by following this chapter's sections on object-oriented programming. It examines and discovers what makes a language object-oriented by learning about the pillars of this concept.

Chapter 4: Generics – The learners will be able to use generics in your programs to be more concise in writing code by getting most of the advantages they offer. They will explore some new concepts that are introduced in Kotlin, such as reified-type parameters and declaration-site variance. They will be taught some issues with generics in Java that are resolved in Kotlin in a very concise way.

Chapter 5: Annotations and Reflection – The readers will learn about annotations and reflection and how to be able to use and define their custom annotations, and will be able to use reflection to introspect classes at runtime as well.

Chapter 6: Functional Programming with Kotlin and RxKotlin - This will give learners some understandings about the functional programming paradigm, reactive programming, functional data structures, higher order functions, how error handling is done, and many other concepts.

Chapter 7: Observables, Observers, and Subjects – It will make readers understand the concepts of observables, observers, and subjects in the context of reactive programming. Learn how to create observables and observers in Kotlin using the RxKotlin library. Explore the various factory methods and operators the RxKotlin library provides for working with observables. Understand the different types of Subjects available in the RxKotlin library and how they can be used. Learn how to use observables, observers, and subjects to create asynchronous and event-based programs in Kotlin.

Chapter 8: Flowables and Backpressure - Flowables and observables are types of data streams that are used in reactive programming. They allow developers to represent asynchronous data streams and react to changes in the data over time. Flowables are a type of data stream designed to handle large volumes of data, known as backpressure. They allow developers to control the rate at which data is emitted and processed, ensuring that it can be processed efficiently and without overloading the system. On the contrary, observations are a simpler type of data stream that does not have built-in support for backpressure. They are often used for smaller data volumes or cases where backpressure is unnecessary.

Chapter 9: Data Transformers and Async Operators - This chapter aims to provide a comprehensive understanding of the various Rx operators and how they can be used to work with streams of data in a reactive programming paradigm. Understanding these operators and their use will help developers create powerful, efficient, and responsive data flows that can handle real-world scenarios.

Chapter 10: Concurrency and Parallel Processing - This chapter covers the concepts of concurrency, schedulers, and their usage in RxKotlin. We will start by understanding the basics of concurrency and how it can be applied in our systems. Then, we will delve into the world of schedulers, which are responsible for scheduling the execution of our tasks. We will learn how to use schedulers with the subscribeOn and observeOn operators in RxKotlin to achieve concurrency and parallel processing.

Chapter 11: Testing Reactive Applications - Testing reactive applications in Kotlin requires a different approach than testing traditional applications. Reactive applications are asynchronous and often handle concurrency, which can make them more difficult to test. However, there are several testing tools and techniques that can be used to make testing reactive applications easier.

Chapter 12: Spring Reactive for Kotlin - Spring Boot is a popular framework for building JVM-based applications, and Kotlin, as we already know, is a modern programming language that has gained popularity in recent years due to its concise syntax and powerful features. Together, they provide a powerful combination for building efficient and scalable applications.

Chapter 13: Asynchronous Programming and Coroutines - It will explore the concepts of multithreading, callbacks, coroutines, jobs, and UI threads. It discuss why multithreading is important in modern software development, including the benefits of improved performance, scalability, and responsiveness. And then dive into the topic of handling work completion using callbacks, which are a powerful mechanism for signaling the completion of asynchronous tasks.

Chapter 14: Suspending Functions and Async/Await- In the real world, suspending functions can be used to write code that performs network requests, database queries, and other asynchronous operations. For example, you could use suspending functions to write code that fetches data from a web service, updates a database, or plays a sound file.

Chapter 15: Contexts and Dispatchers - This chapter focuses on the management and control flow aspects of Kotlin coroutines. It will explore how coroutines are scheduled and executed, the different types of dispatchers available, and how exceptions propagate within coroutines. Additionally, It will delve into the critical topic of coroutine cancellation, understanding the mechanisms behind it and exploring techniques to effectively manage and handle cancellations.

Chapter 16: Coroutines Channels - The concept of channels is an important part of Kotlin coroutines. Channels can be used to implement communication patterns between coroutines, allowing data to be passed between them in a safe and efficient way. This chapter will explore the various features of channels, including generators, pipelines, and broadcast channels.

Chapter 17: Coroutine Flows – The chapter delve into the powerful concept of Coroutine Flows in Kotlin. Asynchronous programming often involves handling data streams that require efficient processing and management. Traditional approaches to stream processing may have limitations when it comes to handling backpressure, composing operators, and maintaining concurrency. Coroutine Flows offers a comprehensive solution to these challenges by providing a declarative and efficient way to work with data streams.

Chapter 18: Multiplatform and Kotlin - This final chapter of the book, will delve into the realm of multiplatform development with Kotlin. Multiplatform development has emerged as a powerful approach to building applications that can run on multiple platforms, enabling code sharing and reducing development effort, It will explore the principles and benefits of multiplatform development and guide readers through the process of setting up development environment for multiplatform projects.

Code Bundle and Coloured Images

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

https://rebrand.ly/wq5ve66

The code bundle for the book is also hosted on GitHub at **https://github.com/bpbpublications/Building-Kotlin-Applications**. 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
Java and Kotlin

Introduction

You probably heard about Kotlin, the new programming language that competes with Java and the other JVM-based languages, and you might also heard that Kotlin is now used by the biggest IT companies such as Google, Amazon, Netflix, Uber, Trello, Pinterest, Foursquare, and others.

You need to know that most of these companies had been using Java for a long time until the appearance of Kotlin, then they had chosen to migrate to this one because of the power it gives them and the interoperability with the existing Java code, which makes the switching to the new programming language gradually easy and not starting the process from scratch.

Taking Google as an example, in 2017, they announced that Kotlin will be the main programming language for Android applications over Java. Coursera, Uber, Duolingo, and other popular Android applications are now using Kotlin for their development.

Kotlin, as a multi-platform programming language, is used almost everywhere, from mobile development and Web applications to scripting, machine learning, and data science, thanks to the concise, expressive style and full interoperability with Java.

This will give you some confidence to go ahead with your choice to this journey of learning Kotlin language over Java and other JVM-based languages for your future IT career.

Structure

In this chapter, we will cover the following topics:

- History of Kotlin
- Kotlin versus Java
- Kotlin
  - Advantages
  - Disadvantages
- Java
  - Advantages
  - Disadvantages
- Next in Kotlin

Objectives

After reading this chapter, you will know the reasons behind creating Kotlin programming language, the differences between Kotlin and Java, and the advantages of using Kotlin over Java as a main programming language for building applications.

Overview

In this first chapter, we will put you in the context of the language, and you will be introduced to the history of Kotlin, why this new programming language has appeared, who created Kotlin, and what was their intention behind Kotlin. We will also learn about Kotlin in comparison to Java; we will learn about the advantages and disadvantages of each and what makes the Kotlin language the right choice for your future development and maybe your career.

History of Kotlin

First of all, we need to get to know a little bit of the history of any language we intend to learn; by that, we will get some understanding of the philosophy behind that language. At this very first stage, we will try to answer the following questions:

- Who is behind Kotlin, and when did it appear?
- Why another JVM-based language?
- What is Kotlin?
- The name Kotlin?

Who is behind Kotlin, and when did it appear?

JetBrains is a Czech software development company that makes tools for software developers and project managers. Initially called IntelliJ Software, it was founded in 2000 in Prague by three Russian software developers: *Sergey Dmitriev*, *Valentin Kipyatkov*, and *Eugene Belyaev*. The company's first product was IntelliJ Renamer, a tool for code refactoring in Java.

***Figure 1.1**: JetBrains logo from 2000 to 2016*
(source: https:// wikipedia.com)

The beams that stem from the Black Box in our logos and icons represent our boundless energy, innovation, and our drive to develop. Combined with the Black Box, it is the perfect visual representation of what JetBrains is and a unique visual element.

***Figure 1.2**: Current JetBrains logo from 2016*
(source: https://www.jetbrains.com/company/brand/)

The company offers many **integrated development environments** (**IDE**) for the programming languages Java, Groovy, Kotlin, Ruby, Python, PHP, C, Objective-C, C++, C#, Go, JavaScript, and the domain-specific language SQL.

InfoWorld magazine awarded the firm Technology of the Year Award in 2011 and 2015.

JetBrains announced Project Kotlin in 2011, a new JVM-based language that had been under development for a year. The first officially stable version of Kotlin was released on February 15, 2016. JetBrains has committed to long-term backward compatibility, starting with this version.

At Google I/O 2017, Google announced first-class support for Kotlin on Android.

Kotlin 1.2 was released on November 28, 2017. The sharing code between JVM and JavaScript platforms feature was newly added to this release (as of version 1.4 multiplatform programming is an alpha feature upgraded from *experimental*). A full-stack demo has been made with the new Kotlin/JS Gradle Plugin.

Kotlin 1.3 was released on October 29, 2018, bringing coroutines for asynchronous programming.

On May 7, 2019, Google announced that the Kotlin programming language is now its preferred language for Android app developers. Kotlin 1.4 was released in August 2020, with for example, some slight changes to the support for Apple's platforms, that is, to the Objective-C/Swift interop.

Kotlin 1.5 was released in May 2021. Kotlin 1.6 was released in November 2021.

At the time of writing this book, the latest version of Kotlin is 1.6.10.

Why Kotlin?

JetBrains development lead *Dmitry Jemerov* said that most existing languages lack the features they are looking for, with the exception of Scala, but Scala has one deficiency, which is the slow compilation time.

One of the targets of a new JVM-based language is to compile as quickly as Java and have the feature set of Scala and the other languages benefits. The other reason behind the investment in a new JVM-based programming language, as explained by Dmitry Jemerov in the JetBrains Blog, is their own productivity; he said that they need to be more productive by switching to a more expressive language; the other reason is to drive the IntelliJ IDEA (Integrated development environment for JVM-based languages) sales.

What is Kotlin?

Kotlin is an open-source, general-purpose, and statically typed programming language that runs on the Java Virtual Machine and also compiles to JavaScript and Native code, and it is concise, safe, and interoperable with other languages.

According to the Kotlin development lead Andrey Breslav, Kotlin is designed to be an industrial-strength object-oriented language and a better language than Java, but keeping the interoperability with Java code, which will give the companies interested in the language to gradually migrate their project from Java to Kotlin.

***Figure 1.3**: Current Kotlin logo*

Kotlin is also influenced by other languages like Python, Groovy, Scala, C#, Gosu, Eiffel, and JavaScript, which make the Kotlin language gain the benefits of these languages and implement them in one language that will be more expressive and easier to work with.

The name Kotlin

The name behind Kotlin comes from Kotlin Island, where most employees of JetBrains; it is a Russian island located near the head of the Gulf of Finland, 32 kilometers (20 mi) west of *Saint Petersburg* in the Baltic Sea (*Figure 1.4*), *Andrey Breslav* (Kotlin development lead) mentioned that the team decided to name it after an island, just like Java was named after the Indonesian island of Java (though the programming language Java was perhaps named after the coffee rather than the island).

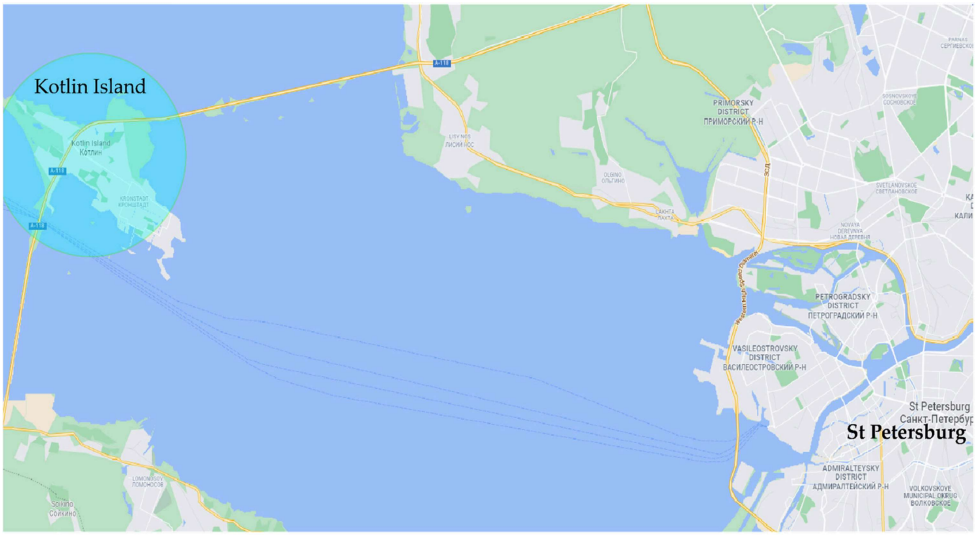

***Figure 1.4**: Kotlin Island on Google Maps*

Kotlin versus Java

You may be wondering: It is okay! You said that Kotlin is better than Java and that most IT companies migrated from Java to Kotlin, but what are the differences between the two languages that make one better than the other if they both work on top of the **Java Virtual Machine** (**JVM**) and what Kotlin has that Java does not or vice versa.

***Figure 1.5**: Java logo*