
Android Application Development Using Kotlin Vs Other Languages

^[1]Sonal.J, ^[2] Srinidhi.R, ^[3] Shraddha.K.B, ^[4] Prof.Kanaiya V Kanzaria

^{[1][2][4]}School Of Computer Science and Engineering, ^[3] School of Computing and Information Technology, Reva University

Abstract

Over the past few years, mobile developers have had the opportunity to make use of new paradigms and tools for mobile application development. Android development has seen a spike in recent years. In 2017 Google announced Kotlin as an official Android application language and since then Android application developers have had the freedom to write Android applications using Kotlin, Java or any other languages such as C++, python, dart, etc...But to this day, Kotlin and Java remain the 2 most favourite languages for developers [24]. Kotlin is a programming language, which can be used together with Java as it combines object-oriented and functional features, Kotlin is interoperable with Java [25]. By the end of this paper, we will conclude which is the most suitable programming language for Android application development with the help of a comparative study.

Index Terms: Android, Kotlin, Java, JavaScript, Dart, Flutter.

1. INTRODUCTION

We have seen the rise in the use of mobile's day by day. Android is developed by Google and is an open-source mobile operating system. Kotlin is one of the most popular and famous programming languages used for Android Application development. Kotlin has proven to be a very effective language as it is more expressive and concise, i.e. you can express your ideas with ease and the number of lines of code will be less in Kotlin compared to other languages. Kotlin also helps users by eliminating common programming mistakes such as null pointer exception and it is interoperable with Java. Kotlin also has a structured concurrency. Android developers can easily code an android application with the help of Android Studio which helps in better functioning of our mobiles. Android studio is the IDE (Integrated Development Environment) which is available to build android applications for Google's Android Operating System and it is built on JetBrains' IntelliJ IDEA software. Kotlin was developed in 2010 and is open source. Android Developers received the option of building an android application in a new language in 2017 when google announced Kotlin as an official programming language for Android application development. Before the launch of Kotlin in 2017 developers used Java and C++ for some specific narratives. Java always remained the most popular programming language among developers but that changed after the launch of Kotlin in 2017. Though Java has been around for a long time and is being used by many developers it has minor setbacks when compared to Kotlin. Java was developed many years ago and hence has a lot of legacy baggage, as every new version of Java must be compatible with its previous version. But when compared, it's easier in Kotlin. Kotlin also has a few major advantages over Java which we will further look into in this paper. Google also announced that Android will be increasingly Kotlin-first. Kotlin-first means that all the new features and APIs will be provided in Kotlin first [1]. This led to further popularity of the language. Many types of research were conducted on how the developers were dealing with the adoption of this new language [2,3]. We will look into many more features, pros and cons of Kotlin in detail in this article.

2. LITERATURE REVIEW

In the comparative study: Prof. Matias Martinez and Bruno Góis Mateus et al.[2] have conducted a qualitative study on the reasons behind switching from Java code to Kotlin and have highlighted a few of the main reasons for the same, by gathering the relevant information from Android developers who switched from Java to Kotlin. A few of the reasons were: For accessing the features of the programming language (such as lambdas, extension functions, etc) which were not available in Java and to obtain safer code (That is to avoid null-pointer exceptions), to shorten the code length and make it easier, to lessen the redundancy, to make the syntax more clear and less Verbose. However, recently the developers of Java have been working with lambda expressions. In his paper, he also described a few drawbacks which were encountered by the developers during the initial state of development of Kotlin. Such as, compilation of Kotlin was slower, the APK size was increased. Now Kotlin has overcome the issue related to compilation and the Kotlin

codes can be compiled faster. In Kotlin there were no checked exceptions, problems with annotations and wildcard types and performance issues. Niket Keny [6] in his research paper discussed the development of android applications and has compared Kotlin and Java programming languages. He has also mentioned why most of the android developers opt for java code over Kotlin. He has also specified that because of the limitations of Java, it caused a problem with the design of Android API. In his survey, most developers felt that Java was simple and very easy to understand. Java being a verbose language requires writing code which can cause the risk of bugs/errors and requires a lot of memory. Victor Oliveira, Leopoldo Teixeira and Felipe Ebert et al. [3] have conducted the research using the concurrent triangulation strategy. They performed an in-depth analysis of all the questions and problems faced by the developers and to know how they deal with the recent adoption of Kotlin as an official language for Android App Development. Most of the developers said that the use of Kotlin can improve productivity, code quality and readability. Subham Bose, Aditi Kundu, Madhuleena Mukherjee, and Madhurima Banerjee et al. [11] have discussed where Kotlin can overpower Java and tried to brief out on which of the two languages is the best fit for the app development and which was most preferred by developers. In his paper he described the developers' point of view on why Kotlin would be preferred, because of the following reasons: safer code, it helps reduce bugs in the code, and makes Android development much easier.

ANDROID STUDIO: Android is an open-source operating system for mobiles. Android Studio is an IDE that helps in developing applications for an Android OS and is currently the most popular IDE to build an application with, with over 2 million applications being built for Android. Android studio was launched by Google on 16th May 2013 [7]. Android Studio offers developers the benefit of building an android application in any of the following languages they prefer: Java, Kotlin, C, C++, C#, Dart, JavaScript, TypeScript, etc.

KOTLIN: Kotlin being cross-platformed, can run on different platforms and is a general-purpose programming language with type inference and is completely interoperable with Java [6]. Due to type inference (where it can automatically detect the type of the expression) its syntax is more concise. It is a statically typed language where the type of variables is known during run time and its libraries depend on the Java class library.

Here are a few features of Kotlin.

Kotlin is interoperable with Java. Kotlin is 100% compatible with the ecosystem of Java frameworks, libraries and tools. A Kotlin code can call a Java code and vice versa. Applications using Kotlin can even run on older devices with no trouble.

Kotlin codes are more concise. Kotlin has lesser number of lines of codes. One of the best features of Kotlin is that it is null safe and this helps in avoiding common mistakes for developers. In Kotlin the getters and setters get generated automatically which makes coding easier for developers.

JAVA: Java is an object-oriented programming language that was developed by Sun Microsystems in 1995. Java is one of the most favoured language by developers around the globe, especially for Android application development since it is easy to learn and is a brilliant choice for cross-platform apps. Android is built on Java and hence Java is one of the most preferred language. The beauty of the Java compiler (Javac) is that it converts the written source code to byte code which can be run on any platform. Java depends on a VM (Virtual Machine) to understand the bytecode. Android has its own virtual machine since every platform that runs java needs a virtual machine. The older version of this Virtual machine was called Dalvik which used JIT(Just In Time) compiler while the newer version of the virtual machine is known as ART (Android run time) which uses AOT(Ahead Of Time) compiler. The actual role of these Virtual machines is to convert the byte code into machine code for the execution of the program to take place. Java language has high security and is a dynamic language [4,5].

3. COMPARITIVE STUDY BEWTWEEN KOTLIN AND JAVA

EXTENSION FUNCTION: Kotlin has this special feature called as extension function where we can extend the class with new functionality and we do not need to create a new class for doing so. On the other hand, java does not have this special feature. In Java, if you need to extend the functionalities of a class you need to create another class and make use of the inheritance concept. You need to inherit the parent class (here parent class is the already existing class) to the newly created class [10] [11].

LESS CODE: Writing code in Kotlin is much easier and simpler compared to Java and Writing codes in Kotlin can cut off many lines of code (up to 40% to be precise) when compared to writing codes in Java [12]. Kotlin's syntax is much stronger compared to java. Due to smaller code, the developers understand the code with ease and the app can be developed much faster [13] [14].

CONSTRUCTORS: There are two types of constructors in Kotlin. One is the primary constructor and the other is the secondary constructor. The primary constructor should always be referred as the secondary constructor. The secondary keyword should be declared by the keyword 'constructor'. The role of this secondary constructor is to reduce the number of lines of code, which is absent in Java [11] [15].

CHECKED EXCEPTIONS: Checked exceptions are not there in Kotlin but is present in Java. We use the try and catch block to handle the runtime exceptions in Java which makes error handling in Java easy. Kotlin does not have checked exception. Throwable class is inherited by all the exception classes in Java. All the exceptions have a message, an optional cause and a stack trace [16]. The flow of the code or logic can be broken due to checked exceptions. It is still a debate whether checked exception is good or a non-checked exception is good [11] [10].

LAZY LOADING: Kotlin has a function named 'Lazy' which takes a higher-order function or a lambda as a parameter. It returns Lazy<T> as an object [17]. To prevent the unnecessary initialization of an object lazy loading is used. The initialization of the variables does not happen until you use the variable in the code [6]. This is basically done to reduce the wastage of memory. The variable is initialized only once, the next time you use the variable, you get the value from the cache memory. The advantage of this is that it is thread-safe. Initialization happens in the thread only for the first time. The second time we get the value from the cache memory. The variable initialized can be of type var or Val (i.e. they can be mutable or immutable) or nullable or non-nullable [18]. Java does not have this Lazy function.

NULL SAFETY: Null safety is a concept in Kotlin which does not exist in Java. Kotlin-type systems have an inbuilt null safety. Kotlin tries to avoid null values by default in order to avoid null pointer exceptions. Kotlin has a lot of Null safety operators some of them are as follows: Safe call operator (?)

Elvis (?:)

Not-null Assertion (!!)

Safe call with let (?let{..})

The benefit of being null safe is that it eliminates the risk of null reference from code [19].

MODEL CLASSES: To model the data in an application we can make use of the model class. In a model class, we can use the objects to receive or send data. Since Java uses encapsulation all of its properties are declared as private. Java uses getters and setters to access these properties along with a few String methods [20]. Kotlin has data classes for model classes, which have various inbuilt utility methods. The use of these data classes is to directly access the properties.

GLOBAL VARIABLES: Variables that have a global scope are called global variables these variables will be available throughout the program unless shadow them. Group of all the global variables together is called the global environment or the global set. In Java, global variables are declared inside a class with a keyword called static. Due to this, the java developers can access the global variables anywhere without initializing the class as an object. Whereas Kotlin does not use the static keyword instead it provides a companion object. You can implement new features such as extensions and interfacing with a companion object. But the use of static variables is always preferred because they are more reliable as they help keep the codes short and clean.

COMMUNITY SUPPORT: Kotlin is comparatively a new language and its community is much younger compared

to Java. The resources such as blogs, tutorials, videos, books, etc to learn Kotlin is very less as it is a new language. Java is easier to learn because of its resources and it's easier to ask for help because of its huge community [21].

DART: Dart is an object-oriented programming language founded by Lars Bak and Kasper Lund and developed in Google to build server and desktop applications. It is an open-source, class-based, general-purpose, garbage-collected language. Dart was first publicly announced by Google on October-12th 2011. It depends on the source to source compiler to JavaScript, to run in web browsers. Dart had the ability of high-performance implementation and was developed for modern app development. Just like Kotlin is interoperable with Java, Dart is interoperable with JavaScript. Flutter is a plugin that is a UI library that has a set of tools for developers, which helps dart in developing applications.

4. COMPARITIVE STUDY BETWEEN KOTLIN AND DART

TYPE SYSTEM: Both Kotlin and Dart are statically typed languages which means that the types are checked during compile time. Kotlin and dart have different syntax, to begin with for example in Kotlin a function needs to be denoted by 'fun' keyword followed by the name of the function which is then followed by '()' and then the code which is enclosed in a parenthesis '{}', while in dart there is no keyword to denote a function. In Dart, we directly specify the name of the function followed by '()' and then the code which is enclosed within a pair of parenthesis '{}'.

NULL SAFETY: We came across Kotlin's Null Safety feature previously in this paper. While Kotlin has Null safety, Dart has sound null safety. The main aim of null safety in Kotlin was to prevent Null pointer exception unless an explicit call to Null Pointer Exception is made or the '!' operator is in the picture. The benefit of having Sound null safety is to help code run faster and enable more compiler optimization. Sound null safety results in small binaries, fast execution and few bugs.

PARTNER/CORE FRAME-WORKS: Kotlin's partner framework is Android while Dart's partner framework is Flutter. The basic architecture of Flutter is more UI-based than that of Android. The main aim of creating Kotlin was to make a platform more advanced than Java. There are data classes, sealed classes and null safety integrated into Kotlin to make it seamlessly interoperable with Java. Dart shares a lot of resembles with java and is a simple language. To make coding experience similar to mapping JSONs, Dart makes use of the dynamic keyword. Dart stands out from the league of both Java and Kotlin because of the feature called Async-wait [22][23].

TOOLS FOR DEVELOPEMNT: Both the languages have various useful tools. When writing an app on any IDE such as visual studio or IntelliJ, Kotlin is more convenient than Dart this is because Kotlin provides faster coding through features like post-fix completion. The feature of info popups in constructors that rely on field assignment syntax is not available in Dart. Also, the static analysis feature of Dart is a dubious concept [22].

POPULARITY OF THE LANGUAGES: The popularity of any language is based on how user-friendly it is. Google announcing that Kotlin will be the most preferred language for Android development caused the popularity of Kotlin to increase drastically. In India, Kotlin has become the most used language as of 2022. Dart's popularity is still less and is growing when compared to Kotlin [22].

5. FIELD SURVEY

We conducted a survey where we asked professional android app developers to give their perspectives on this narrative. We asked questions to which they answered based on their point of view which helped us greatly and this would enable us to find the best-fit language for android app development.

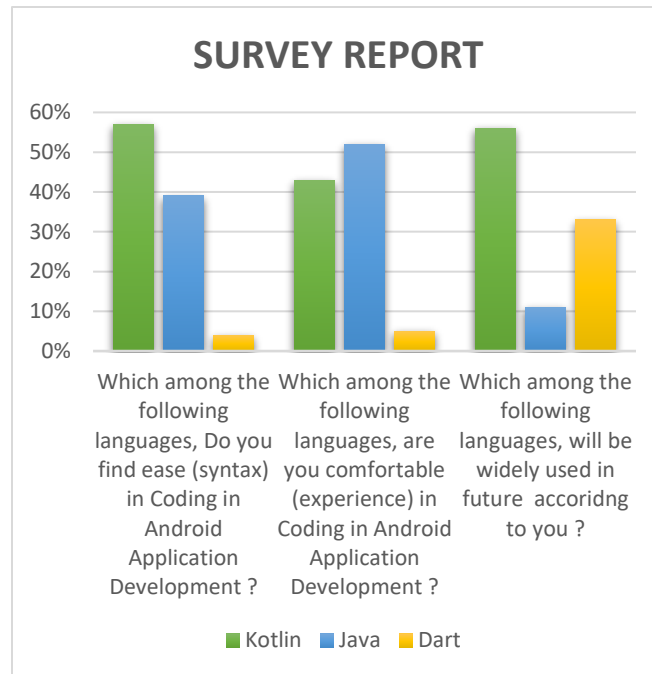


Figure 1: Bar graph showing the accuracy levels of languages.

RESULT: As per the field survey we conducted the results are in fig1. The first set of the bar graph shows that 57%, 39% and 4% accepted Kotlin, Java and Dart respectively for having said that, these languages are easy in terms of having easier syntax or the methodology they used for understanding the language hence a majority of the people find it easier to learn and code in Kotlin as compared to Java and Dart. The second set of the bar graph shows that 43%, 52% and 5% chose Kotlin, Java and Dart as the most comfortable languages in terms of their experience when it comes to working with the language. Here Java is preferred in a higher number as Java is an old language and has a huge developers community and has many resources to develop the developers knowledge on Java. The third set of the bar graph shows that 56% chose Kotlin, 11% chose Java and 33% chose Dart as the future language. It is predicted that in future Kotlin and Dart might be the next popular languages used after Java as Kotlin is gaining popularity among developers and dart which must be used along with flutter is gaining popularity because it is multi-platform language. To answer the question on whether Java has or doesn't have a stand-in android application against Kotlin, one of the developers in our survey expressed his thoughts in the following words "Even though Kotlin is more popular and has an edge over Java, people would still prefer Java because over the past years Java is being widely used in most of the platforms. Thus switching over to Kotlin in all those platforms would take decades and thus Java would still be alive for years before Kotlin took over it."

6. CONCLUSION

Android development is one such domain which has not seen a downfall in the market and would not be seeing a downfall anywhere in the future. Kotlin looks like a promising future language and hence we took it upon us to research on this topic.

From the above comparative study, it is seen that Kotlin, Java and Dart have their own pros and cons. If your application is going to be less complex, easy to maintain and it needs to be developed quickly it's better to use Kotlin. With the help of Kotlin we not only can build android applications but we can also create web applications, native applications, server-side applications and desktop applications. Google has been using a lot of its resources to develop Kotlin. Companies like Netflix, Uber, Pinterest are using Kotlin [28].

But in case your application is complex and large it is better to use Java as it's easier to find Java developers and there are more resources to help with it. Java can be used in various sectors other than android application sector like financial

services, Big Data, Banking[26], Stock Market, Retail, Scientific and Research Community. There is a huge demand for java programmers who takes up the opportunity in the biggest brands of the world like Amazon, LinkedIn, eBay, Twitter, etc for building a solid infrastructure between their backend systems and their web application [27].

And if you want to create cross-platform apps or for client development, such as web applications, we can use the Dart programming language as a replacement for JavaScript. Dart along with flutter has a huge scope in the future due to its multi-platform nature.

The result of the recent survey which we conducted shows that many of the android application developers prefer Kotlin language the most, followed by Java, and then Dart in the current narrative.

Kotlin language overcomes most of the weak points of Java, like its memory usage, number of lines of codes and syntax. On the other hand, Kotlin overpowers Dart due to features like postfix completion.

Thus, we can conclude that there is no perfect language for Android development. It depends on the developer to choose which programming language is comfortable for them to work with based on their requirements.

7. ACKNOWLEDGEMENT

We would like to extend our gratitude to Prof.KANAIYA V KANZARIA for guiding us throughout this paper and to all the developers who shared their experiences.

8. REFERENCES

- [1]. <https://www.venturit.com/post/why-is-android-development-kotlin-first>
- [2]. Matias Martinez and Bruno Gois Mateus. 2020. How and Why did developers migrate Android Applications from Java to Kotlin? A study based on code analysis and interviews with developers. <https://arxiv.org/pdf/2003.12730.pdf>.(2020).arXiv:2003.12730
- [3].V. Oliveira, L. Teixeira, and F. Ebert. 2020. On the Adoption of Kotlin on Android Development: A Triangulation Study. In 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, 206–216
https://www.victorlaerte.com/doc/On_the_Adoption_of_Kotlin_on_Android_Development__a_Triangulation_Study.pdf
- [4]. <https://www.youtube.com/watch?v=sJVenujWGjs>[5]<https://www.youtube.com/watch?v=0J1bm585UCc&t=414s>
- [6]. https://www.ripublication.com/ijaerspl2019/ijaerv14n7spl_04.pdf [ISSN0973-4562]
- [7]. https://en.wikipedia.org/wiki/Android_Studio
- [8]. https://www.researchgate.net/publication/334624041_On_the_adoption_usage_and_evolution_of_Kotlin_Features_on_Android_development
- [9]. <https://ieeexplore.ieee.org/document/9315483> [INSPEC Accession Number: 20361726]
- [10].<https://www.youtube.com/watch?v=9wWgw9smBJs>
- [11]. <https://www.semanticscholar.org/paper/A-COMPARATIVE-STUDY%3A-JAVA-VS-KOTLIN-PROGRAMMING-IN-Banerjee-Bose/b047262d0a95f9d910e5336703c08d1a78902e31>
- [12]. [https://www.programiz.com/kotlin-programming#:~:text=Why%20Learn%20Kotlin%3F-Kotlin%20is%20100%20percent%20interoperable%20with%20Java.,%25%20\(compared%20to%20Java\)\[](https://www.programiz.com/kotlin-programming#:~:text=Why%20Learn%20Kotlin%3F-Kotlin%20is%20100%20percent%20interoperable%20with%20Java.,%25%20(compared%20to%20Java)[)
- [13]: <https://aglowiditsolutions.com/blog/kotlin-vs-java/>
- [14]. <https://medium.com/javarevisited/kotlin-vs-java-which-is-the-best-choice-for-android-app-development-7c9fc782d2c9>
- [15]. https://www.ripublication.com/ijaerspl2019/ijaerv14n7spl_04.pdf
- [16]. <https://kotlinlang.org/docs/exceptions.html>
- [17]. [https://agrawalsuneet.github.io/blogs/lazy-property-in-kotlin/#:~:text=lazy%20is%20a%20function%20defined,of%20Template%20class%20\(%20T%20\)](https://agrawalsuneet.github.io/blogs/lazy-property-in-kotlin/#:~:text=lazy%20is%20a%20function%20defined,of%20Template%20class%20(%20T%20)).
- [18]. <https://www.youtube.com/watch?v=orUgnUuYW44>

- [19]. <https://www.javatpoint.com/kotlin-nullable-and-non-nullable-types>
- [20]. <https://aglowiditsolutions.com/blog/kotlin-vs-java/>
- [21]. Basha, S. M., Poluru, R. K., & Ahmed, S. T. (2022, April). A Comprehensive Study on Learning Strategies of Optimization Algorithms and its Applications. In *2022 8th International Conference on Smart Structures and Systems (ICSSS)* (pp. 1-4). IEEE.
- [22]. <https://www.androiddeveloper.co.in/blog/dart-vs-kotlin/>
- [23]. <https://blog.codemagic.io/dart-vs-kotlin/>
- [24]. [https://en.wikipedia.org/wiki/Kotlin_\(programming_language\)](https://en.wikipedia.org/wiki/Kotlin_(programming_language))
- [25]. <https://www.guru99.com/kotlin-vs-java-difference.html#:~:text=Kotlin%20combines%20features%20of%20both,and%20Java%20supports%20implicit%20conversions.>
- [26]. <https://www.ksolves.com/blog/java/java-in-banking-why-do-banks-use-java-preferentially>
- [27]. <https://www.thetechlearn.com/future-scope-of-java/>
- [28]. <https://thesynergyworks.com/why-how-kotlin-is-the-future-of-mobile-app-development/>

