INTERNATIONAL JOURNAL OF TECHNOLOGY AND COMPUTING (IJTC)
ISSN-2455-099X,
Volume 3, Issue 8 August 2017

Review Study on New Era of Android Kotlin

[1]Jahanvee Narang, [2]Er.Salony Tuli

[1]B.Tech (CSE), 2Assistant Professor (CSE)

[12]CGC Technical Campus, Jhanjeri, Mohali[12],

[1]jahanvee.narang@gmail.com, 2er.salony@gmail.com

Abstract: In the New Era of Technology, Google declared a new-fangled popular language for android applications and Software's i.e. KOTLIN. KOTLN is a programming language which includes the targets java standards. Kotlin is best fitted for android as it brings all the advantages of new era modern pros into it and that to be without limiting any of its techniques. Kotlin was written by company named JETBRAINS, which also developed in the android studio. As we have already discussed that the language is similar like java programming language. Kotlin name also came from an island named St. Peterberg named Kotlin Island. Software Company, Jetbrains found that the java has long congested program to work with which results in less production. Kotlin is designed in such a way that it can be interoperate with java code and vice-versa. This means that we can easily convert java code existing file into kotlin file. Kotlin includes functional programming which makes program more readable and includes null point exceptions.

Keywords: Jet Brains, JVM, Kotlin.

I. INTRODUCTION

Kotlin is statically type language ie. Variable types declared are stable or static in which once we set a variable to a type, then we cannot switch it. This programming language runs on Java virtual machine (JVM) similarly like java but kotlin syntax is not congruous with java code. It can also be compiled with Javascript source code which again a bonus to this language as JS is used as a foundation of programming languages such as Node.JS and many server side technologies. Kotlin gives more focus on object as compared to java as it offers functional programming with object oriented programming, thus making it more exciting for programmers to code. With all other new applications, kotlin also introduces some complications.

The elementary motto of kotlin is to contribute to less of congested codes and more of productivity and a safer alternative programming language to java. In future kotlin, may be supportive for software companies and developers to bring out their code and programs more smoothly. Most prevalent areas where kotlin has been used are:

- Backend web applications
- Building mobile applications
- Using intel multi operating system engine one can run kotlin code on Mac devices.
- Provides great assimilation with libraries that support programming paradigm.

As we learn in the above passage that the kotlin can also work on functional programming, which is new to many users. Thus learning it would not be easy task but in the beginning we have to be patient and go calmly. The concept of functional programming is to write a program with functions and store it as variable and pass them as parameter from other functions. Functional code looks more effective and compact which is a bonus to this language. Also working with it gives more absorption or abstraction of the code which endorse to skip the duplication in code. Functional programming has a benefit of easier testing that lets one person to check the code more easily and also these functions will be tested in concealment which do not requires a lot system code. Kotlin never enforces to program in functional code. Thus, due to this it is able to solve a lot of headache problems facing by developers and programmers while working with java.

Now as Google has announced kotlin officially for their android applications, so it would be a tough completion between java and kotlin[2].

Similarly, here also java scope would be get much in future. As kotlin has also java features and with a plus point it has overcome all the java problems which are faced by developers. So, you just have to push up yourself and explore yourself to more things and don't just stuck up yourself with java and c++. There a lot more languages to learn you can go for it. Kotlin is mostly a statically-typed programming language that runs on the Java Virtual Machine and also can be compiled to JavaScript source code. Its primary development is from a team of JetBrains programmers based in Russia.[2] While the syntax is not compatible with Java, Kotlin is designed to interoperate with Java code and is reliant on Java code from the existing Java Class Library, such as the collections framework.

II. NARRATIVE OF KOTLIN

The journey with Kotlin started over 6 years ago with the aim to create a language that would be in line with the same principles that drive our tools – create something that helps developers with the tedious and mundane tasks, allowing them to focus on what's truly important. And of course make the process as enjoyable and fun as possible. Within a small time period , Kotlin has been successfully adopted by major companies Previously they had experience with Kotlin, but on a much smaller scale converting some parts of apps to a new language or trying it on pet projects. However, developing a commercial application in a new programming language. Kotlin is being promoted by google android and was announced as an official Android development language at Google I/O 2017. It became the third language fully supported for Android, in addition to Java and C++. Google has long had a close relationship with JetBrains. Android Studio, Google's official integrated development environment **(IDE),** is based on JetBrains IDE. So, it comes as no surprise that Google announced at Google I/O that it would deliver "first-class support for Kotlin ". Kotlin is an excellent fit for Android not only because it gives developers what they want, but also because it matches the spirit of Android. Kotlin is a great fit for developing Android applications, bringing all of the advantages of a modern

language to the Android platform without introducing any new restrictions. One of the obvious applications of Kotlin is Android development. As the android platform was stuck on Java 7 for a while and Kotlin introduces many improvements for programmers such as null-pointer safety, extension functions and infix notation. Accompanied by full Java compatibility and good IDE, it is intended to improve code readability, give an easier way to extend android classes and speed up the development. This is a great step for Kotlin, and fantastic news for Android developers as well as the rest of our community. Android developers are thrilled with the opportunities this opens up. For them, Kotlin support is a chance to use a modern and powerful language, helping solve common headaches such as runtime exceptions and source code verbosity. . Kotlin is easy to get started with and can be gradually introduced into existing projects as it is also designed to have a gentle learning curve for Java developers. Java developers should find that most of the Kotlin syntax feels familiar. Kotlin aims to be an enhancement to Java, rather than a complete rewrite, so many of the skills have acquired and honed throughout in Java career. Programming languages are just like human ones, the more people speak a language, the better they become use to it. First-class support on Android will likely bring more users to Kotlin, and we expect the community to grow significantly. This means more libraries and tools developed in/for Kotlin, more experience shared, more Kotlin job offerings, more learning materials published, and so on. Kotlin can help make your life as an Android developer a lot easier. Kotlin is on pretty solid footing.

III. EMINENCE OF KOTLIN

Following are the Eminence features of Kotlin:

A. *Extra Runtime Size`*

The Kotlin Standard Library and runtime will increase the size of your **.apk**. While this only equates to around 800KB, if your application is already on the large side then that extra 800KB may tip it over the edge and make users think twice before downloading your app.

B. *Initial Readability of Code*

Although Kotlin's syntax is one of the language's greatest strengths, you may initially find some Kotlin difficult to decipher, simply because there's so much going on in such a small amount of code. Java may be more verbose, but the upside is that everything is clearly spelled out, which means unfamiliar Java code tends to be easier to decipher than unfamiliar Kotlin. Also, if used incorrectly, Kotlin's operator overloading can result in code that's difficult to read.

C. *Lack of Official Support*

Kotlin may have excellent support in Android Studio, but it's worth remembering that Kotlin isn't officially endorsed by Google. Also, Android Studio's auto-complete and compilation tends to run slightly slower when you're working with Kotlin, compared to a pure Java project.

D. *Smaller Community and Less Available Help*

Since Kotlin is a relatively new language, the Kotlin community is still fairly small, particularly compared to the community surrounding more established languages like Java.

IV. BENEFITS OF KOTLIN:

A. Less code

At least 20% less code was written in kotlin than compared to 22 years old java. Kotlin's architecture was created from scratch, causing the absence of layer to layer architecture.

B. Less crushes occur on Kotlin.

Kotlin JVM has much fewer issues like the issue of Null Pointer Exception is preventable there. Null Pointer is yet present in every programming language, but the ways of working with it may differ**.**

C. Kotlin is a type-safety language

Again, in Kotlin every class is a function, and vice versa; more to this, same as Swift for iOS, Kotlin for Android has Optional types, which help with all the safety checkups.

D. Kotlin saves your precious time.

As it was already said, less code almost guarantees less bugs and, respectively, less time is tracked on the whole coding process. Kotlin would be the Game Challenger for the Android Platform Development. While it's true that the majority of Android apps are written in Java, when it comes to Android development, Java isn't your *only* option but a new JVM-compatible programming language that's really caught the attention of the Android community is Kotlin, a statically typed programming language from JetBrains. Kotlin is simple, clean and removes a lot of the code bloat from Java**.** Kotlin also adds some needed features that Java doesn't yet support in Android.

V. WHAT KOTLIN HAS THAT JAVA DOES NOT

- Kotlin has proper function types, as opposed to Java's SAM-conversions
- Kotlin does not have checked exceptions.
- Java has checked Exception but Kotlin does not have this.
- Java have checked exceptions but not in Kotlin.
- Static members are used in Java but Kotlin does not have.
- Extension functions
- Null-safety
- Smart Casts
- String Templates
- Properties
- Primary Constructors
- First-class delegation
- Type inference for variable and property types
- Singletons
- Declaration-site variance & Type projections

- Range expressions
- Operator overloading
- Companion objects
- Data classes
- Separate interfaces for read-only and mutable collections

VI. FIVE THINGS WHICH MAKE KOTLIN INTERESTING

I. It Like Swift: Kotlin predates Swift so it's technically more accurate to say that Swift resembles Kotlin. it means that the syntax for Kotlin, like Apple's Swift, is clean and modern. This is an advantage in any case, but when it's the alternative to a language like Java – a language that, for all its strengths, is not known for its syntactical beauty – syntax becomes a killer feature.

II. It's Java Compatible: While Kotlin breaks with Java on multiple fronts, its ability to both leverage the JVM and interoperate and intermingle with Java code is enormously important.

III. Its tools are top notch: This is particularly notable in cloud platforms; for all the meteoric growth, the typical development experience for most cloud developers is more rudimentary that might be expected. Kotlin's tooling story is different, to the point that the enthusiasm in this description by Yegge is actually common:

IV. It's an Officially Sanctioned Android Language: Swift's growth was, of course, driven by iOS. Prior to its introduction, Objective-C was that platform's supported development language. If the goal is to compare the opportunity in front of Kotlin to Swift, then, the obvious question is about size of target. While there are obviously other factors to consider – Java and Objective-C have very different user populations, for one – consider the public numbers.

V. Its Supporters**:** it's one thing to use a technology because it gets the job done or because it's ubiquitous; it's quite another for it to also be likeable. By all accounts, regardless of its other strengths, Kotlin is a language that is adored by users in a fashion that is rare.

REFERENCES

[1]. https://en.wikipedia.org/wiki/Kotlinhttps://blog.mindorks.com/should-i-learn-kotlin-or-stick-to-java-3a73c3580ac3

[2]. https://www.lynda.com/Android-tutorials/Learning-Kotlin-Android-Development/550456-2.html

[3]. https://kotlinlang.org/docs/reference/comparison-to-java.html

[4]. https://www.quora.com/What-is-the-future-of-Kotlin-programming-language

[5]. https://blog.mindorks.com/should-i-learn-kotlin-or-stick-to-java-3a73c3580ac3

[6]. https://blog.jetbrains.com/kotlin/2017/05/kotlin-on-android-now-official/

.