DOI: http://dx.doi.org/10.26483/ijarcs.v9i3.5978

ISSN No. 0976-5697

Volume 9, No. 3, May-June 2018

International Journal of Advanced Research in Computer Science

CASE STUDY

Available Online at www.ijarcs.info

A COMPARATIVE STUDY: JAVA VS KOTLIN PROGRAMMING IN ANDROID APPLICATION DEVELOPMENT

Subham Bose
Student, BCA
The Heritage Academy
Kolkata,India

Aditi Kundu
Student, BCA
The Heritage Academy
Kolkata,India

Madhuleena Mukherjee
Student, BCA
The Heritage Academy
Kolkata,India

Madhurima Banerjee
Asst. Prof., BCA
The Heritage Academy
Kolkata,India

Abstract: The purpose of this paper is to compare and conclude between Java and Kotlin in android application. We have taken different fields and compared it with Java and Android. This paper attempts to study the various features of both Java and Kotlinand in the end concludes which programming language fits the developers.

Keywords: Android; Kotlin; Checked Exception; Lazy Upload, Extended Functions

I. INTRODUCTION

In the advancing world of technology, mobile applications are a rapidly growing segment of the global mobile market. Mobile applications are evolving at a meteor pace to give users a rich and fast user experience. In this paper, we have discussed how java is used as the programming platform of Android application. New technology like Kotlin is upcoming promising area of Android technology and the comparison between Java and Kotlin and where Kotlin can overpower Java.

II. ANDROID

Google launched Android as an open-source mobile phone operating system which works on Linux platform. It consists of the operating system, middleware, and user interface and application software [12]. Certainly, Android is widely used as operating system on mobile phones. It was primary built for touch-screen mobile devices such as smart-phones and tablets.

In addition, Google has further developed Android TV for televisions, Android Auto for car and Wear OS for wrist watches [14], each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics. Android comes in various versions. Each version of the Android has been named after a dessert. The first named version of Android is called Cupcake [16]. Marshmallow, Nougat, Oreo, Lollipop, Kitkat, Jellybeans are some of the other Android versions. According to [15], most recent Android version in February 2018 is Android Oreo, though; Android Nougat is more popularly used.

Android comes with certain security issues, that few users take into account.

Till 2016, we could download thousands of applications for Android, anyone could upload their programs without having to submit them to careful security checks in Android Market. This makes Android a prime target for computer criminals. Google has launched a Play Protect Project to overcome the above issue [13].

III. OVERVIEW OF JAVA

Java is a programming language first released by Sun Microsystems back in 1995. It can be found on many different types of devices from smart-phones, to mainframe computers. We can use it on your desktop PC and even on the Raspberry Pi. Java doesn't compile to native processor code but rather it relies on a "virtual machine" which understands an intermediate format called Java bytecode [1][25]. Each platform that runs Java needs a virtual machine (VM) implementation. On Android the original VM is called Dalvik. Google has also started previewing its next generation VM called ART. The job of these virtual machines is to interpret the bytecode which is really just a set of instructions similar to the machine code found in CPUs, and execute the program on the processor. The VMs use a variety of technologies including just-in-time compilation (JIT) and ahead-of-time compilation (AOT) to speed up the processes.

This all means is that we can develop Android apps on Windows, Linux or OS X and the Java compiler converts the source code into bytecode. This in turn is executed on the VM built-in to Android. This is different to the model used by iOS which uses a native compiler to turn Objective-C into ARM machine code. [2]

According to [17] Java is used for Android development because, it is a well known language amongst the developers, it does not have complications of pointer arithmetic. Since it runs on VM, so it is not required to recompile the code for every device the code it used on. Though speed is a issue for JAVA, yet its popularity and advantages overweighs over the speed.

IV. KOTLIN

Kotlin (Android_and_Kotlin) is a statically-typed programming language that runs on the Java Virtual Machine and also can be compiled to JavaScript source code. It was was released to the public in February 2016. Its primary development is from a team of JetBrains programmers based in Saint Petersburg, Russia (the name comes from Kotlin Island, near St. Petersburg). In fact, the name Kotlin comes from Kotlin Island in Saint Petersburg[18].

In May, 2018, Google Android team announced that Kotlin is now the official Language for Android development. Developers have been using Kotlin to build Android apps in previous years but Google just announced a first class support for it[18].

Both Kotlin and Java can be used to build Android apps.
Now the question is why should there be a switch in the programming languages?

V. LITERATURE

In a comparative study: Prof. Panchaland, R. K and Patel, A.K have studied many works in android with java, so they have studied with Kotlin instead of java. This work can be further enriched to achieve Kotlin related mobile apps. [18]
On 26th June, 2017, Android Authority published a video on 10 Reasons to try Kotlin for Android development It shows what new features Kotlin brings to android application development [19]. Adding Extensions, Handling Null pointers are some of the benefits discussed in the paper.
Holla, S. and Katti M.M. has discussed security issues of Android application and future scope in their paper [20]. The issues like android store's risk of getting virus affected application. They have discussed the future scopes like addition of more sensors to the future devices which will make Android more secured.

VI. OBJECTIVE

The purpose of this paper is to compare and conclude between Java and Kotlin in android application. We have taken different fields and compared it with Java and Android. This paper attempts to study the various features of both Java and Kotlin and in the end concludes which programming language fits the developers.

VII. RESEARCH METHODOLOGY

This study is an explanatory study that leads to understandings of role of java in android development and focuses on Kotlin as replacement of java in android application. It is a qualitative study and the secondary data has been gathered from different source published in electronic media.

VIII. COMPARATIVE STUDY BETWEEN KOTLIN AND JAVA

A. *Extension Function:*

If it is required to add some extra features in a class, in most programming languages, a new class is derived. An extension function is a member function of a class that is defined outside the class.

Extension function can be elaborated with the example given in [21].

As per the example, we need a function String class has to return a new string with first and last character removed; this method is not available in the String class. The extension function declared outside class creates a functionality of the specified class extending the predefined functionalities. The function can be extended as follows:

```
fun   String.removeFirstLastChar():   String   =
this.substring(1, this.length - 1)


fun main(args: Array<String>) {
    val myString= "Hello Everyone"
    val result = myString.removeFirstLastChar()
    println("First character is: $result")
}
```

"Kotlin provides the ability to extend a class with new functionality without having to inherit from the class or use any type of design pattern such as Decorator. This is done via special declarations called extensions. Kotlin supports extension functions and extension properties."[4]

This Extension function is not present in java. For availing the functionality of extension functions, Android frameworks are commonly used.
But Android framework sometimes makes things difficult. Java provides only one solution, that is to create wrappers.. [5]
But Kotlin has given us the advantage of extension function which would eliminate the difficulties posed by Android framework.

B. *Checked Exception:*

JAVA uses try...catch block to handle runtime exceptuions. It mainly uses checked exceptions.
A checked exception is a type of exception that must be either caught or declared in the method in which it is thrown.

Following is the syntax of JAVA try...catch block [22].

```
try
{ // some code }
catch (e: SomeException)
{ // handler }
finally
{ // optional finally block }
```

There can be zero or more catch blocks and one or no finally block. Either a catch block or a finally block is required.
In JAVA, if some code within a method throws a checked exception, then the method must either handle the exception or it must specify the exception using *throws* keyboard.

Kotlin does not have checked exception. All exception classes in Kotlin are descendants of the class Throwable. Every exception has a message, stack trace and an optional cause. [6]

Using throw expression can be used in Kotlin to handle exceptions [22]:

```
fun fail(message: String): Nothing
{ throw IllegalArgumentException(message) }

val s = person.name ?: throw IllegalArgumentExcepti
on("Name required")
```

It is a topic of debate whether a checked exception is better than non checked exception. Some programs find way to implement checked exceptions in Kotlin, but here we state the advantages of not using a checked exception.

Checked exceptions can break the logic or flow of the code [23]. Specially in codes with lot of callback methods, using checked exceptions can generate lost flow of code.

Secondly, In case of large softwares, checked exceptions lead to less productivity and little or no increase in code quality [22].

C. Constructors:

Kotlin can have a primary constructor as well as a secondary constructor [8].
"constructor" keyword is used to declare a secondary constructor. It should always refer to the primary constructor[7].

Following example of secondary contructor:

```
class Student {
   val name: String
   val roll: Int
   val marks: Int
   private var elective = false

   constructor(name: String, roll: Int, marks: Int) {
      this.name = name
      this.roll = roll
      this.marks = marks
   }

   constructor(name: String, roll: Int, marks: Int, elective:
Boolean) : this(name, roll, marks) {
      this.elective = elective
   }
```

In the above example, the primary contructor has 3 parameters and the secondary contructor has 4 parameters.
In case of constructor overloading in JAVA, the code would have looked like:

```
class Student {
        String name;
        Int roll;
        Int marks;
        Int elective;
        Student (String name, Int roll, Int marks)
        {
                this.name = name;
                this.roll = roll;
                this.marks = marks;
                this.elective = 0;
        }
        Student (String name, Int roll, Int marks, Int elective)
        {
                this.name = name;
                this.roll = roll;
                this.marks = marks;
                this.elective = elective;
        }
  }
```

This secondary constructor feature is absent in Java. The utility of secondary constructor is that it reduces the lines of code.

D. Null Safety:

One of the most common pitfalls in many programming languages, including Java, is that accessing a member of a null reference will result in a null reference exception. In Java this would be the equivalent of a NullPointerException or NPE for short [3][9].

Kotlin uses the feature called Null Safety to handle the NULL pointer situation. Unless it is explicitly called for, Kotlin does not throw a NullPointerException. [9]
Following is a code in Java

```
Public static void main(String args[])
{
        String name= null;
        System.out.println(name.length());
        }
```

Output: Null Pointer Exception

Following is a code in Kotlin-

```
Fun main(args: Array<String>)
{
        var name: String? = null
        println(name?.length)
}
```

Output: Null

While coding in Kotlin, the Null Pointer exception does not break the flow of the code unlike in JAVA. It gives the output as NULL.

E. Lazy-Loading:

Lazy loading is used in computer programs to defer initialization of an object until the point it is needed. Thus, the lazy-loading feature decreases the loading time.
Kotlin gives us the feature of lazy loading unlike JAVA.
In case of java, there is no such feature as lazy-loading, so a lot of the non required content is loaded during start up of the application thus loading the application slower.

IX. CONCLUSION

While analysing we deciphered that both Java and Kotlin have their own advantages and disadvantages.
From beginners' point of view, Java is a better option due to the following reasons-
Jaava is a very popular language which is used extensively amongst developers. Android development is only a drop in the ocean of development happening across the world. Thus

being a beginner, knowing JAVA is more advantageous than Kotlin widening the spectrum of opportunities.

Secondly, there is a huge community of Java programmers, which means we find answers to critical programming issues when we are stuck. This is very important because, as a beginner, facing technical problems ia a common scenario and we might not know where to head when we are stuck. When we search about Java problems, we are bound to get answers; the same cannot be said for Kotlin, which is still an upcoming programming language.

There are also more tutorials, books, and courses out there, both free and paid, which can teach us Android development with Java, same cannot be said for Kotlin.

Thinking from developers' point of view, Kotlin would be preferred due to the following reasons [10]:

1. Language and environment are mature

The Kotlin release has gone through many stages before release of the final 1.0 version unlike other programming languages.
This might means that all the probable issues commonly arising in other programming languages have already been taken care of during the evolution to the final stage. [5]

2. It makes Android development much easier

Kotlin makes programming easier and Android apps better. Kotlin is a modern programming language. It opens the window for a large number of possibilities for android app developers, i.e it makes developers more productive. [5]

3. Kotlin helps reduce errors and bugs in the code

The Kotlin compiler aims to fail-fast whenever possible which greatly facilitates searching for bugs. To avoid runtime errors and reduce the cost and effort required in fixing errors the Kotlin compiler performs many checks.[11]

4. Kotlin code is safer

Common programming mistakes by design can be easily prevented by using Kotlin, resulting in fewer system failures and application crashes. This proves Kotlin code is inherently safer than any other programming code. [11]

5. Kotlin is more concise

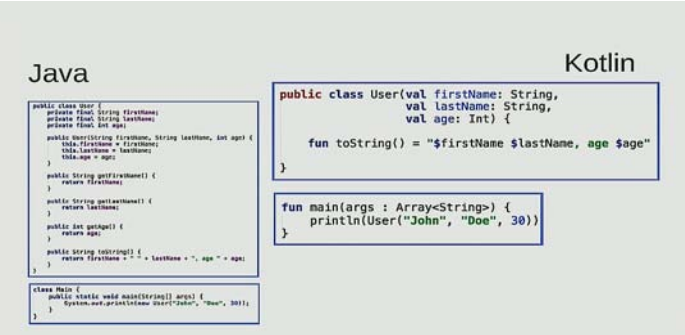
Kotlin is way more concise than any other programming languages in many cases; it lets us solve the same problems with fewer lines of code. This improves code maintainability and readability, which means engineers can write, read, and change code more effectively and efficiently. [11]

As Android gives the support to convert a project to Kotlin a developer is always free to switch.

Following is the screen shot of Android studio that allows conversion of Android code written in JAVA to Kotlin.

Kotlin is desirable as it increases productivity. A class which takes 50 lines of code in Java can be written in very few line in Kotlin. Boiler-plate code can be avoided, e.g. we don't need to specify setters(), equals(), hashCode() or toString() methods. All these can be generated by Kotlin.

Number of lines of codes in Kotlin is much less than the number of lines of code in JAVA for achieving the same output. An example of this is taken from [24] is given below:



X. REFERENCES

[1] E. Obugyei, 2016, "Kotlin for Android: An Introduction", viewed on 25th April, 2018 from https://www.raywenderlich.com/132381/kotlin-for-android-an-introduction.

[2] C. Singh, n.d., "Introduction to Java Programming", viewed on 1st May, 2018 from https://beginnersbook.com/2013/05/java-introduction/

[3] A. Sinicki, 2018, "An Introduction to Kotlin for Android development", viewed on 25th April, 2018 from https://wwwhttps://www.androidauthority.com/introduction-to-kotlin-for-android-775678/

[4] "Kotlin", viewed on 1st May, 2018 from https://kotlinlang.org/docs/reference/extensions.html

[5] A. Leiva, n.d. , "Extension functions in Kotlin: Extend the Android Framework (KAD 08)", viewed on 1st May, 2018 from https://antonioleiva.com/extension-functions-kotlin/

[6] ChikeMgbemena, n.d., viewed on 7th May, 2018 from https://code.tutsplus.com/tutorials/kotlin-from-scratch-exception-handling--cms-29820

[7] "Kotlin", viewed on 8th May, 2018 from https://kotlinlang.org/docs/reference/classes.html

[8] D. Odalodic, June, 2017, "Dusan Odalodic @ JVM", viewed on 8th May, 2018, from https://odalinjo.wordpress.com/2017/06/25/primary-and-secondary-constructors-in-kotlin/

[9] M. Daga, May, 2018, "Java vs Kotlin: Which Programming Language Is Better for Android Developers?", viewed on 8th May, 2018 from https://dzone.com/articles/java-vs-kotlin-which-programming-language-is-bette

[10] "What are the advantages of Kotlin over Java? ", viewed on 8th May, 2018, from "https://www.quora.com/in/What-are-the-advantages-of-Kotlin-over-Java

[11] P. Sommerhoff, January, 2018, "Kotlin vs. Java: 9 Benefits of Kotlin for Your Business", viewed on 7th May, 2018 from https://business.udemy.com/blog/kotlin-vs-java-9-benefits-of-kotlin-for-your-business/

[12] G. Suite, n.d., "Android", viewed on 9th May, 2018 from https://www.engineersgarage.com/articles/what-is-android-introduction

[13] I. Majocha, December, 2017, "Report: Top Android Security Problems in 2017", viewed on 9th May, 2018 from https://dzone.com/articles/report-top-android-security-problems-in-2017

[14] Developers, n.d., "About the platform", viewed on 9th May, 2018 from https://developer.android.com/about/

[15] Fossbytes, n.d., "Most Popular Android Versions In February 2018 (Always Updated List)", viewed on 9th May, 2018 from https://fossbytes.com/most-popular-android-versions-always-updated/

[16] Turbofuture, April, 2016, "Android Version Names: Every OS from Cupcake to Marshmallow", viewed on 9th May, 2018 from https://turbofuture.com/cell-phones/Cupcake-Donut-Eclair-Froyo-Gingerbread-Honeycomb-Android-OS-Version-Codenames-and-Why

[17] "Why does Android use Java?", viewed on 9th May, 2018 from https://stackoverflow.com/questions/3560963/why-does-android-use-java

[18] R.K. Panchal, and, A.K. Patel, 2017, A comparative study: Java Vs kotlin Programming in Android , in International Journal of Innovative Trends in Engineering & Research, September 2017, vol 2 Issue 9, pp 4 – 10.

[19] Android Authority, 2017, "10 reasons to try Kotlin for Android development", viewed on 9th May, 2018 from https://www.youtube.com/watch?v=LEi1ecigDFE

[20] S. Holla and M.M. Katti, 2012, ANDROID BASED MOBILE APPLICATION DEVELOPMENT and its SECURITY, in International Journal of Computer Trends and Technology, 2012, Vol 3, Issue 3, pp 486- 490

[21] Programiz, n.d., "Kotlin Extension Function", viewed on 9th May, 2018 from https://www.programiz.com/kotlin-programming/extension-functions

[22] "Kotlin", viewed on 8th May, 2018 from https://kotlinlang.org/docs/reference/exceptions.html

[23] E. Petrenko, n.d., "Catching exceptions with less code in Kotlin", viewed on 9th May, 2018 from http://jonnyzzz.com/blog/2017/02/15/catchall/

[24] Hype.codes, 2017, " Kotlin Vs Java", viewed on 9th May, 2018 from https://hype.codes/kotlin-vs-java

[25] Schildt, The Complete Reference Java, Seventh Edition, Chapter 1, page 9

[26] https://code.tutsplus.com/tutorials/learn-java-for-android-development-introduction-to-java--mobile-2604, viewed on 9th May, 2018

