International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 05 | May - 2021 ISSN: 2582-3930

Kotlin for Android Application Development

A1: ATHIRA ANIL

M.C.A Student, Dayananda Sagar College Of Engineering

CO-AUTHOR: RAKSHITHA KIRAN
Assistant Professor, Department of MCA, Dayananda Sagar College Of Engineering

---

Abstract

Java programming language has been utilized to foster native Android applications for quite a while. In the previous few years, numerous organizations and freelancers have begun utilizing Kotlin either totally or in part. In 2017 Google announced Kotlin, a programming language that combines functional and object-oriented features as an official programming language. Kotlin is a pragmatic programming language, Kotlin runs on the Java Virtual Machine(JVM) and Android.

The paper aims in analysing Kotlin and Java with Android by leading a Literature Review on the development process. The main issue that is analysed in this paper is how Kotlin contrasts from Java and on various features of Kotlin and Java along with their drawbacks.

Key Words: Android, Kotlin, Extended Functions, Checked Exception, Lazy loading

1.INTRODUCTION

JetBrains announced Project Kotlin, a new language for the JVM that had been in development for a year, in July 2011. Kotlin v 1.0 was released to public on 15 February 2016. The developers at JetBrains needed a tool that was easier to use than Java, they needed something to work on their product called IntelliJ IDEA that is their main product. The folks looked into other options but were unsatisfied, so they decided to create a language of their own, a language that was simpler than Java and something that works along with java. That was the objective which they thought of when creating Java, they wanted a powerful tool that is used along side Java in native android application development and a language that works anywhere that Java is supported. Despite the reality that the syntax isn't consistent with Java, Kotlin is designed to work with Java code and is reliant on Java code from the current Java Class Library, such as the collections framework.

The developers at JetBrains started with the development of Kotlin during the year 2010, after six years in early 2016 that is during February the first verion of Kotlin (ie., version 1.0) was released. Kotlin was announced as an official language for Android development at the Google I/O annual conference in 2017. Kotlin is a statically typed programming language that may be compiled to JavaScript source code or used with the LLVM compiler framework. It operates on the Java virtual machine.

Kotlin is accumulated to Java byte-code, which implies that it is interoperable with Java, i.e., Kotlin code is able invoke code written in Java and the other way around, both running on the equivalent basic JVM. Because of this interoperability, engineers of Java-based Android applications can: i) adding new Kotlin code and keeping up the current Java code, and/or ii) migrating a few parts of theirs applications written in Java code to Kotlin. Duolingo, is an application which allows its users to learn a new language along with pronunciations and practice speaking, this app took nearly 2 years to migrate to Kotlin, in the time between complete migration the application had a few functionalities in java and Others in Kotlin. In this paper we mainly focus on how java is the foundation language for android applications and an comparison analyses among Java and Kotlin, as well as the areas where Kotlin may be able to outperform Java.

2. LITERATURE REVIEW

In a paper by Prof.,Patel and Prof.,Panchal, they have done a comparative study where they have done a the detailed study on the works that are done in Java for android and also focused on Kotlin benefits and how it can be used instead of Java. As a result this can be developed to create Kotlin applications.[3]

On its YouTube channel, Android Authority posted a video titled “10 Reasons to Try Kotlin for Android Development”.[4] The video demonstrates the features of Kotlin and how it can improve android application. Some of the advantages discussed in this video include on how null pointers are handled.

In their work, M.M.Katti and S.Holla, have analysed the security issues that is faced in android applications.[5] The paper was mainly focused on the risk that comes with the app store and how we can make the devices more safer without getting infected by viruses.

In another work by Asst. Prof. Madhurima Banerjee, Madhuleena Mukherjee, Shubham Bose, and AditiKundu, conducted a comparative analysis, in which the authors have looked at the differences between Java and Kotlin.[6] They

essentially centered around Kotlin since it is easier to create and consumes less time.

3. ANDROID.

Android is a Linux-based open-source mobile operating system that was developed by Google. Android has user interface, a middleware and an operating system. Android is unquestionably the most popular operating system for smartphones. Android was designed by keeping mobile devices that has touch-screen in mind, they can either be smartphones or tablets, Google has also created a television called Android TV which uses android. There are additional devices such as wrist watched called Wear OS, vehicle named Android Auto that uses Android framework.[7] Gaming consoles, computers, digital cameras and some other electrical devices also uses android. Android comes in different renditions. When we see of different android versions, they were named after some dessert. If we look at the first version it was named ac Cupcake, even the later versions were named on deserts. But recently they were just named on the version like Android 10 and Android 11. These versions are just given a number and not the name of any deserts, although they do have some sweet names which is used internally for the development. Android versions may include Jellybeans, Lollipop, Oreo, Marshmallow etc., The latest version of Android is Android 11 which was released in September 2020. Android OS has a few security issues that are ignored by many and taken into consideration by a couple of clients.

Till 2016, we could download hundreds of Android applications and there was no restriction in posting their applications in Android market or play store. There were no strict security checks that was done on the android market, if anyone wanted to post their applications they could have done it, as a result the market became main target to cybercriminals. This became a concern to many users to protect the clients from getting infected by viruses and becoming a victim to these cybercriminals Google established a project, Play Protect. With Google Play Project, all apps could not be posted and the apps that are posted has to go through strict security checks, which solved the issue.

4. KOTLIN.

Let's see about Kotlin, Kotlin is a programming language that is statically typed and it is open-source that helps developers to develop applications without having to purchase any additional software or license. Kotlin community has grown rapidly and is growing still. Google has even officially announced Kotlin as second official language for android app development. Kotlin can also be used as a cross platform language, in some cases developers use Kotlin for general purpose programming as they use Java. It can be run in JVM ie., Java virtual Machine or in JavaScript. The team that developed Kotlin from the JetBrains team were based in Russia. (Kotlin got its name from Kotlin Island which is situated in Russia).

Feature of Kotlin.

- *Interoperability.*

  The most remarkable feature of Kotlin is its interoperability; you can call Java code from Kotlin and the other way around. There is no compelling reason to change over the whole project into Kotlin from the earliest starting point.

- *Freedom from Null Pointer Exceptions.*

  Freedom from Null Pointer Exceptions. We need not fear Null Pointer Exceptions, in light of the fact that NULL value checks are moved from runtime to compile time in Kotlin, which implies that, getting rid of null pointer exception that is Null safety, is in the framework we don't have to include external files, it is a part of the actual framework. All the variables in Kotlin language does not contain null values that means the variables have non-null values.

- *Easy to Learn.*

  Most of the Android developers are familiar with Java and ad Kotlin is an improvement of Java and the creation of class is similar to in Java so Kotlin is easier for most of us to understand. Kotlin is simpler when compared to Java that makes it easier for us to learn Kotlin. When we compare Java and Kotlin we notice that Kotlin code is simpler and easy to code. As the size of code is smaller than Java its easier to understand and the code is readable. The syntax of the Kotlin is concise and, crisp, that gives an advantage to developers when compared to Java.

- *Smart Casts.*

  A coder need not stress over explicitly casting operators, in light of the fact that Kotlin's compiler embeds casts automatically any place required.

Kotlin drawbacks.

Despite the fact that Kotlin appear to be the most encouraging one it likewise have a few blemishes.

- *Sluggish compilation.*

  The compilation speed which is genuinely not as much as its competitive languages is a minor disadvantage of Kotlin .

- *Small developer community*

  For the present Kotlin actually has a little developer community notwithstanding its fast selection among coders.

- *Larger package size*

  The bundle size of Kotlin is greater in size when contrasted with Java.

- *Java to Kotlin*

Around 80% of the Java code converts to Kotlin may be a help consistently. The leftover 20% of the code gets completely mixed and can be too drawn-out to even consider settling.

5. JAVA

Java is the language in which most Android apps are based on and as Android developers are used to working in Java, it is the most loved language, especially for android developers. Java is a language that runs in different system with the same code, the program written in Java is compiled into bytecodes that run different OS let it be MacOS, Windows or Linux, it runs on all platform with the same byte code.

Feature of Java.

When we see the features of the Java, there are many features a few of them are that Java is a language that can be easily learnt, as it uses objects it is easy to comprehend. Java is an Objct-Orinted language where the objects are used. Another feature of Java is that it is flexible, it means that a java program works in any browser or any virtual machine. When it comes to cros-stage application Java is the decent selection to it. In case of Android, Java is the main part that an android application depends on its mainly because the SDK of Java contains numerous libraries that helps to run the application. When we take security into account Java stands out as it gives us a virus free software.

Drawbacks of Java..

No language is perfect or without any drawbacks, similarly Java too has its limits which sometimes can effect the Android API design. We can say Java as a verbose language, as it's a verbose language there are risk of creating errors or bugs, the boiler plate for Java is bigger there are chances for developers to make mistakes in the syntax. . It is more slow when contrasted with numerous different languages. It requires a great deal of memory.

6. OBJECTIVE

The motivation of this paper is to mainly to constract Android applications written in Java and Kotlin. This paper attempts to consider the advantages of both Java and Kotlin along with the drawbacks.

7. RESEARCH METHODOLOGY

This is an illustrative study that explains the functionality on Java in Android and the difference in Java vs Kotlin and it spotlights on Kotlin as a replacement to Java in Android Applicatioin Development. The data was gathered from various sources that was published previously.

8. COMPARATIVE STUDY BETWEEN KOTLIN AND JAVA

- **Null Safety:**
  In Java whenever a null value has to be returned it gives an exception called NullPointerException(NPE), this is not just the case of Java; many other programming languages will return similar exception. In Kotlin there is no exceptioin for null return it just return null not an exception. To cope with NullPointerException, Kotlin has Null safety, which doesn't throw NPE. In some case if the developer wants to throw NullPointerException then the developer can explicitly call them, until they are called explicitly null safety is ensured.

  An example of Java code is shown below

```java
public class App {
    public static void main(String[] args) {
        String exampleString= null;
        System.out.println(exampleString.length());
    }
}
```

  The Output for the above code is:

```
Exception in thread "main" java.lang.NullPointerException
```

  Now lets see the same code in Kotlin

```kotlin
fun main(args: Array<String>) {
    var exampleString: String? = null // can be set null
    println(exampleString?.length)
}
```

  Output:

```
null
```

  The Null Pointer error does not stop the flow of the code in Kotlin, whereas it does in Java. It returns NULL as a result.

- **Extension Function:**
  In the event where we want to add additional features to an application in most programming languages we have to add a new class, but in Kotlin we can use the extension functionality which helps to add an additional feature to the applocation where the function is defined outside of a class but it is a membr function of that class.
  Below is an example code to add a new feature to the main method to remove or eleminate the first and last charcters in a string. Java doesn``t have this feature.

```
fun String.removeFrstLstChar():
    String = this.substring(1, this.length - 1)
fun main(args: Array) {
    val myString= "Hello World"
    val result = myString.removeFrstLstChar()
    println("First character is: $result")
}
```

- **Constructors:**

Constructors are special methods that are used in intializing data in a program. Java has constructors where we cwn even overload the constructors and pass different constructors, whereas in Kotlin there are primarly two types of constructor one is primary and another one is secondary constructor. In Kotlin we have to use the keyword constructor to declare a constructor. In Kotlin one constructor can call a constructor inside another constructor. Primary constructor in Kotlin is a prt of the header: it is usually at the end of class name.

The primary constructor can't contain any code.Secondary constructors are not in any way regular in Kotlin language. We principally utilize Secondary constructor when we want multiple constructor and want to initialize the fields in multiple ways.

Below is the code in Kotlin for constructor:

```
class Stud {
    val sname: String
    val sroll: Int
    val smarks: Int
    private var selec = false
    constructor(sname: String, sroll: Int, smarks: Int)
    {
        this.sname = sname
        this.sroll = sroll
        this.smarks = smarks
    }
    constructor(sname: String, sroll: Int, smarks: Int, selec: Boolean) : this(sname, sroll, smarks) {
        this.selec = selec
    }
}
```

From the above code we can see that the first constructor that is the primary constructor has 3 parameters, that is name, roll and marks. In the secondary constructor there are 4 parameters and as first 3 parameters are same as they are in primary constructor, the primary constructor is called in secondary constructor.

Java constructor overloading code is given below:

```
class Stud {
    String sname;
    Int sroll;
    Int smarks;
    Int selec;
    Stud (String sname, Int sroll, Int smarks) {
        this.sname = sname;
        this.sroll = sroll;
        this.smarks = smarks;
        this.selec = 0;
    }
    Stud(String sname, Int sroll, Int smarks, Int selec){
        this.sname = sname;
        this.sroll = sroll;
        this.smarks = smarks;
        this.elec = elec;
    }
}
```

Java constructor doesn”t have the feature of secondary constructor. As we can see from the code with the help of secondary contructor we can reduce the code lins.

- **Checked Exception.:**

When it comes to handling exceptions Java utilizes try and catch blocks. This handling is checked exception. Whenever an exception occurs it has to be thrown and, caught and handled in a method that is known as checked exception.

The syntaxt for exception handling in Java is as follows:

```
try {
    // some code where exception may occur
} catch (e: SomeException) {
    // handler to handle exception
} finally {
    // finally block which is optional
}
```

In Java we need to have atleast a catch block and one or more finally block. When an exception occurs it has to be caught and a message has to be displayed either on the console or a required action has to be taken so that we can identify a checked exception has occurred. The catch block catches the excetion which that has occurred in a code inside the try block. Kotlin is a language which does”t have checked exception. In java whenever we need to implement exception handling then we need to extend the throwable class, but in Kotlin all classes are child of Throwable class so there is no need to extend a class to catch excetions. Each and every exceptin in Kotlin has a messge, optional cause and a stck trace. With the help of throw expression we can handle excption in Kotlin.

```
fun fail(message: String):
    Nothing {
        throw IllegalArgumentException(message)
    }
val s = person.name ?: throw IllegalArgumentException("Name required")
```

- **Lazy-Loading.:**

  Lazy loading is utilized in computer, a program language which is utilized to vary initializations of an object, until which the point is required. The lazy loading time varies the loading time of any program. Kotlin gives us the feature of Lazy loading yet Java doesn't support Lazy loading which prompts a great deal of non-required content is stacked during the startup of the applications.

9. CONCLUSIONS

To conclude, Java acquires a higher ground when considering for general purpose programming as it is easy to understand and comprehend. On the other hand a rising number of engineers and assoiciations are now embracing Kotlin for rapid app development in Android. When contrasting Java and Kotlin both have their weaknesses and benefits on one another.

REFERENCES

1. https://en.wikipedia.org/wiki/Kotlin_(programmin g_language)
2. R.K.Panchal and, A.K.Patel, 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
3. https://www.javatpoint.com/features-of-java
4. S.Holla and M.M.Katti, ANDROID BASED MOBILE APPLICATION DEVELOPMENT and, its SECURITY, in International Journal of Computer Trends and Technology, 2012, Vol 3, Issue 3, pp 486- 490
5. https://clearbridgemobile.com/java-vs-kotlinwhich-is-the-better-option-for-android-appdevelopment/
6. Shubham Bose,Madhuleena Mukherjee,AditiKundu, and,Asst. Prof. Madhurima Banerjee A Comparative Study: Java vs. Kotlin Programming in Android Application Development, in International Journal of Advanced Research in Computer Science
7. https://www.ijiter.com/wpcontent/uploads/papers/2016/A-comparative-studyJava-Vs-kotlin-Programming-in-Android.pdf
8. https://www.techopedia.com/definition/25106/androidoperating-system
9. https://antonioleiva.com/extension-functionskotlin/