Imperial Journal of Interdisciplinary Research (IJIR)
Vol-3, Issue-8, 2017
ISSN: 2454-1362, http://www.onlinejournal.in

Kotlin – The unrivalled android programming language lineage

Manish Jangid
Student, Shaheed Sukhdev College of Business Studies, University of Delhi, India

Abstract*: Time constraint and productivity are two major things kept in mind when working on any Android project, which could not be easily achieved by the boilerplate code and innumerable exception handlers in Java. Thus, it was time to introduce a new and more mature language, which has been effectively done by JetBrains by introducing Kotlin. The paper means to address the significant portions where Java lacked as an Android programming language, and why Kotlin is going to take over Java real soon.*

1. Introduction

The most popularly adopted and prevalent language in the sphere of Android programming has always been Java. It's been all around the programming world so predominantly that the term "Java" automatically springs up in the minds of coders whenever they hear about programming. There's no doubt in the fact the Java took over the programming market prematurely and became unshakable. Rather it is the number one language that any coder would opt for his career.

However, being popular or being widely used doesn't necessarily conclude that it remains unsurpassed throughout the eras. With time coders eventually realized Java's predicament, about its verbosity and its error prone nature. Well its true that almost all the Android applications are shaped with Java, but don't worry, that's not the only option in hand.

Nonetheless, there's no need to be concerned about these issues stated with Java because Kotlin is here to back you up. In simple words, no more crying out whenever you see a Null Pointer Exception popping up, or lamenting on seeing your clustered Java code. Kotlin comes with a whole bunch of mind blowing features that would prove out to be a boon for the coders.

This paper addresses the various ways where Kotlin gains the upper hand over Java and why exactly Java is getting pushed out of the frame of Android programming real soon. Parallely, it suggests coders, why is it the most perfect time to get started with Kotlin to code modern and more pragmatic programs.

2. Why is Java amiss?

It is a really cumbersome task to get acquainted to a challenging programming language and unanticipatedly coming to know about a new language in the market. Java being a platform independent language has been loved by everyone. It is being used almost everywhere and for many purposes but everything has a drawback or the other. It seems like every other day another security loop hole pops up in Java that everybody cries on.

2.1. Java has gone rather decrepit

It's true that Java was one of the best programming languages, but time flies by real soon. In the modern world, no coder can imagine coding without the integration or support of lambda functions & streams, and no programmer wishes to hassle with plugins to achieve the same. Even though Android Nougat introduced new features in Java 8 using the Jack compiler, but that can only be put into use if working with a minimum SDK Version of 24 or higher. And going by today's trends, updating of Android versions by respective users takes place on a very slow pace. Thus, much could not be benefitted by this new introduction by Android Nougat.

2.2. It's verbose

Ever got a headache on seeing heaps of code on your computer screen after finishing up with a project? The answer by most of the programmers would be a big Yes. Who doesn't want clean and simple code, which is self understandable and easily read by others also, but Java makes it cumbersome to organize code in a concise way. Writing huge sections of code even for the smallest of tasks, is one of the biggest drawbacks of Java that every coder faces. And at the end, it turns out to be a pain for the programmers to manage multiple files with innumerous lines of codes in it.

2.3. Java – an unsafe language

One of the well known drawback of Java is the Null Pointer Exception, also referred to as "THE BILLION DOLLAR MISTAKE".

"I call it my billion-dollar mistake. It was the invention of null reference in 1965… This has led to innumerable errors, vulnerabilities, and system crashes, which have probably caused a billion dollars of pain and damage in the last forty years" – Sir Charles Antony Richard Hoare

Not even a single application now today is free from the Null Pointer Exception, because Android uses NULL vastly over and over again, which makes it one of the most major drawbacks of Java when it comes to efficiently handling Null Pointer Exceptions. Explicit handlers have to be used for each instance of the problem which makes it too tedious for a task to be done over again and again.

Per Brinch Hansen argued in 1999 that Java's implementation of parallelism in general and monitors in particular to not provide the guarantees and enforcements required for secure and reliable parallel programming. While it is possible for a programmer to establish design and coding conventions to, say, only access thread-global variables in a controlled fashion, the language and compiler make no attempt to enforce that controlled access, i.e. the programmer may mistakenly allow uncontrolled access to thread global variables, and the compiler will not detect it. [1]

3. Kotlin – for the coder's liberation

Kotlin is a statically-typed programming language that runs on the Java Virtual Machine and also can be compiled to JavaScript source code or uses the LLVM compiler infrastructure. Its primary development is from a team of JetBrains programmers. While the syntax is not compatible with Java, Kotlin is designed to interoperate with Java code. [4]

"We believe 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." – Google Developer Product Group

To reduce the misery of coders, Kotlin is pragmatic and a more mature language which comes with a whole bunch of amazing features. The reason why Kotlin stands out in the sea of programming languages is that it improves over Java's limitations and positively affects day to day development workflow. It is incredibly powerful and has a handful of things which would attract any coder.

3.1. 100% Interoperability

The first and the most outstanding feature in Kotlin's magic box is the interoperability. One can call a code written in Java to Kotlin and vice-versa smoothly. So, no need to convert your entire project into Kotlin from the beginning. Just tap into your PC, and start writing it into your existing Java program. One thing to keep in mind is that the syntax of Kotlin is not compatible with Java, it's that Kotlin is designed to interoperate with code written in Java.

3.2. Freedom from Null Pointer Exceptions

You don't need to fret anymore if you are afraid of Null Pointer Exceptions, since Kotlin takes NULL value checks from runtime to compile time. This means that Null safety is a part of the system itself. In Kotlin, all variables are non-null. Thus, to declare a nullable variable, just add a '?'

For example, if you try declaring a variable as shown in Figure 1, the compiler will generate an error "Null cannot be a value of non-null type string"

```
var name : String
name = null
```

Figure 1.

Thus, you need to explicitly mention the null variable, i.e. as shown in Figure 2 with a '?' mark.

```
var name : String?
name = null
```

Figure 2.

3.3. Lesser verbosity

No one likes writing loads of lines of code for small functions. To improve the situation, Kotlin helps you write concise and crisp code to help save ample time and decrease the clustering & boilerplate code. Kotlin understands the code and can infer the type of variable declaration as well as getters / equals / hashcode generated by the compiler. It helps the programmer to get rid of hassling task of putting semicolons after every statement. This saves time as well as helps in increasing productivity. Figure 3

shows a simple example to prove lesser verbosity of Kotlin over Java.

```
System.out.println();  //Java

println()  //Kotlin (without any semicolons)
```

Figure 3.

3.4. Smart Casts

No more worrying about explicitly casting operators because Kotlin's compiler inserts casts automatically wherever needed. To check whether an object conforms to a given type at runtime, one can use the 'is' or '!is' operator.

For example, let us consider a function 'MethodX' as declared in Figure 4. Here 'obj' could be any class if it isn't null. But no worrying about null checks or type checks, here's where Smart Casts come in handy.

```
fun methodX(obj : Any?){
//lets imagine that we need to check for a string
if (obj == null || obj !is String)
{ return }
// from now onwards, obj would be considered as a String and not Null, which saves us from the hassle of checking for null or do any explicit casts
val length_string = obj.length
}
```

Figure 4.

3.5. Destructuring Declarations

Destructuring simply means treating a single object as a set of multiple variables, i.e. declaring multiple variables for an object in one go.

For example, let us consider the name, age and gender of a person. With the statements shown in Figure 5, we have created 2 new variables.

```
val person = person("Manish", 21)
val (name, age) = person

// the 2 variable we created
println(name)
println(age)
```

Figure 5.

For the compilation part, a Destructuring declaration is compiled as shown in Figure 6. Also, Destructuring declarations can be used for lambdas as well as loops.

```
val name = person.component1();
val age = person.component2();
```

Figure 6.

3.6. Awesome IDE and Plugin support

Kotlin being a JetBrains product leaves nothing to worry for the programmer, since it's one of the best companies who specializes in creating the finest IDE's. In order to get Kotlin, just install a simple plugin in Android Studio or even in Eclipse (Yes! It's that easy).

Another amazing plugin contains "Convert Java file to Kotlin" which converts your existing Java code to Kotlin without having you worry about it. This feature comes in handy if you have lots of Java code and you find it difficult to mix it with Kotlin in between. So there you go, Kotlin is to the rescue once again.

4. Is Kotlin perfect?

The question that would pop up in your mind right now after reading Kotlin's rescuing features to make programming hassle-free would be "Wow, that looks great. But is it perfect now?". The simple answer is no. It may seem to be the most promising language but even the better things have some flaws.

4.1. Sluggish compilation

A minor drawback of using Kotlin is the compilation speed which is fairly less than its competitive languages.

4.2. Small developer community

Despite its rapid adoption among coders, Kotlin still has a small developer community for the time being. This directly means hassling to find solutions if you stick at any point, and there are lesser sources to learn this new language from. For example, searching for Kotlin on StackOverflow returns around 3600 posts while there are more than 1000,000 posts related to Java.

4.3. Larger package size

Another issue with Kotlin is that the end product is heavier in size than its predecessors. As compared to Java, the package size of Kotlin is bigger in size.

4.4. Java to Kotlin

The converting of existing Java code to Kotlin might be a boon but it converts only around 80% of the code seamlessly. Rest 20% of the code gets thoroughly scrambled which can be too tedious to resolve.

5. Conclusion

Kotlin is overall a programming language that every coder has ever dreamt of, when working tediously on Java. It makes Android programming lot more interesting and easy to learn. It reduces complexity of code and helps you take care in managing your code in a crisper manner.

Taking everything into account, if you are an Android developer, you got to give Kotlin a try. Its automated syntax converter that detects Java code as soon as you paste it in your current file, is a boon since it converts most of the Java code seamlessly. What else would you want a programming language to be like?

In a nutshell, Kotlin is the next big thing that you should totally switch to, if you are a hardcore Android lover. Even after some of the drawbacks that it still has, it helps you save ample amount of time in a much neater and helpful way.

So go ahead, grab on your PC, download the Kotlin plugin for your Android Studio and get started. There won't be any better time to learn Kotlin since it's newly arrived and not many people have a major experience with it.

6. References

[1] Criticism of Java - https://en.wikipedia.org/wiki/Criticism_of_Java

[2] Kotlin reference - https://kotlinlang.org/docs/reference/

[3] Java vs. Kotlin: Should you be using Kotlin for Android Development? by Jessica Thornsby - https://code.tutsplus.com/articles/java-vs-kotlin-should-you-be-using-kotlin-for-android-development--cms-27846

[4] Kotlin (Programming Language) - https://en.wikipedia.org/wiki/Kotlin_(programming_language)

[5] Kotlin - The Good, the Bad, the Ugly by Keepsafe Engineering - https://medium.com/keepsafe-engineering/kotlin-the-good-the-bad-and-the-ugly-bf5f09b87e6f

[6] Kotlin in the Real World by Philippe Breault - Published January 26, 2017 in Technology - https://speakerdeck.com/pbreault/kotlin-in-the-real-world

[7] Kotlin + Android First Impressions by Keyhole Software, April 20, 2016 - https://www.javacodegeeks.com/2016/04/kotlin-android-first-impressions.html