International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 5 Issue 4, May-June 2021 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

A Review Paper on Kotlin Programming Language

Siddhi Sanjay Shinde, Prof. Pratibha Adkar

MCA Department, Modern College of Engineering, Pune, Maharashtra, India

ABSTRACT

There are many alternative programming languages like C, C++, JAVA, Python, etc. Every artificial language has its advantages and downsides, features. What's Kotlin? Why the android developers choose this is often that Kotlin is to enable mixed-language projects. Kotlin also introduces improved syntax, still as concise expressions and abstractions. Using Kotlin with Java reduces excessive boilerplate code, which could be a huge win for Android developers. Kotlin has been included as another to the quality Java compiler. Many concepts in Kotlin helps programmers to begin with Kotlin is meant to be an industrial-strength object-oriented language, and a "better language" than Java, but still be fully interoperable with Java code, allowing companies to form a gradual migration from Java to Kotlin. Some benefits of Kotlin are that semicolons are optional as an announcement terminator; in most cases, a newline is sufficient for the compiler to deduce that the statement has ended. Kotlin variable declarations and parameter lists have the information type come after the variable name (and with a colon separator), the same as BASIC, Pascal, and Typescript.

KEYWORDS: *Android, Architecture, Feature, Application of Java and Kotlin*

How to cite this paper: Siddhi Sanjay Shinde | Prof. Pratibha Adkar "A Review Paper on Kotlin Programming Language" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-5 | Issue-4, June 2021, pp.1182-1185, URL: www.ijtsrd.com/papers/ijtsrd42529.pdf

IJTSRD42529

1. INTRODUCTION

Kotlin languages, including (butnot limited to) Java, Scala, C# and Groovy. One in all the foremost ideas behind Kotlin is being pragmatic, i.e., being a linguistic communication useful for day-to-day development, which helps the users get the task done via its features and its tools. Thus, many design decisions were and still are influenced by how beneficial these decisions are for Kotlin users. Kotlin may well be a multiplatform, statically typed, general-purpose linguistic communication. Currently, as of version 1.4, it supports compilation to the following platforms.

- JVM (Java Virtual Machine)
- JS (JavaScript)
- Native (native binaries for various architectures)

The type system of Kotlin distinguishes at compile time between nullable and not-nullable types, achieving null-safety, i.e., guaranteeing the absence of runtime errors caused by the absence burning (i.e., null Value). Kotlin also extends its static type system with elements of gradual and flow typing, for better interoperability with other languages and straightforward development. Kotlin is an object-oriented language which also incorporates lots of functional programming elements. From the object-oriented side, it supports nominal subtyping with bounded parametric polymorphism (akin to generics) and mixed-site variance. From the functional programming side, it's first-class support for higher-order functions and lambda literals. This specification covers Kotlin/Core, i.e., fundamental parts of Kotlin which should function mostly the identical way irregardless of the underlying platform.

2. APPLICATION DEVELOPMENT:

The platform organizations needed to develop, deploy and manage mobile apps are made from many components and tools which allow a developer to write, test and deploy applications into the target platform environment.

A. Front-end development tools

Front-end development tools are focused on the user interface and user experience (UI-UX) and provide the following abilities:

- UI design tools
- SDKs to access device features
- Cross-platform accommodations/support

B. Back-end servers

Back-end tools pick up where the front-end tools leave off, and provide a set of reusable services that are centrally managed and controlled and provide the following abilities:

- Integration with back-end systems
- User authentication-authorization
- Data services
- Reusable business logic

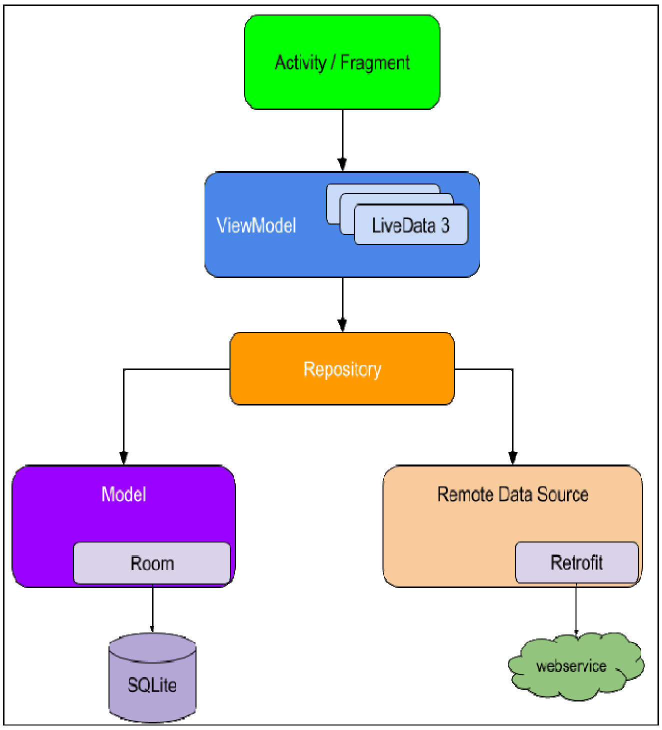


Figure 1: shows how all the modules should interact

3. WHAT IS KOTLIN?

It is an open-source statically typed artificial language designed by JetBrains. Yep, the identical guys who created IntelliJ IDEA. You will say "IntelliJ is that the best IDE for JAVA, those people know the way to try to to the things, this Kotlin can be a very good tool" – and you're right! I mean, if you wish JAVA and think it's a cool language then Kotlin is even cooler. Kotlin is object-oriented and supports functional programming features. It's designed for the JVM (Java Virtual Machine). What does that mean? That's fully compatible with JAVA. Actually, to be honest, Kotlin is Java with some extra pieces of stuff. What does that mean? Those languages are often employed in parallel within the same project. For instance, you'll be able to easily use JAVA libraries in Kotlin's project, but Kotlin gives you more out of the box. It's focused on interoperability, safety, clarity, and tooling support. You'll use it to a code Back-end server, Android Apps or maybe iOS app using Kotlin Native. Another interesting fact is that you simply can compile Kotlin to JavaScript.

OK, that sounds good, but why is it better than JAVA? The primary argument which involves my mind is Kotlin's syntax. When working with Kotlin, we are able to reduce the code by plenty of lines. Such code are more transparent and straightforward to grasp. JAVA could be a very nice language, but because the project expands, the amount of code lines increases logarithmically. Eventually, you'll need a major amount of your time to seek out what you're searching for in your code. Developers of Kotlin tried to induce eliminate this disadvantage, and that they did so o.k. i'll not show you each difference in syntax because it might take way more space than a blog article ever should, but I can explain the essential idea in simple cases. What number DTO objects have you ever written? All of them looks almost like getters, setters, hash Code, equals, to String, and constructors? For easy objects like as an example Student together with his id first Name, surname, age, country it'd be over 50 code lines.

4. KOTLIN ARCHITECTURE:

Kotlin could be an artificial language and has its own architecture to allocate memory and produce a high quality output to the tip user. Following are the various scenarios where Kotlin compiler will work differently, whenever it's targeting different other reasonably languages like Java and JavaScript. Kotlin compiler creates a byte code which byte code can run on the JVM, which is precisely adequate to the byte code generated by the Java .class file. Whenever two byte coded file runs on the JVM, they will communicate with one another and this is often how an interoperable feature is established in Kotlin for Java.

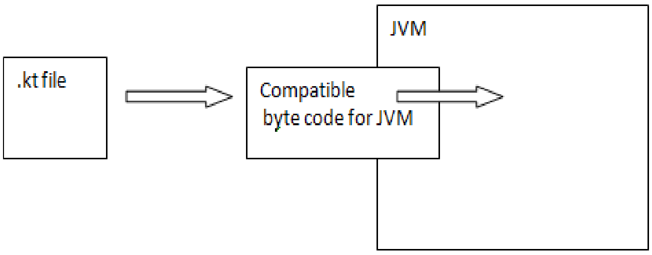


Figure 2: Kotlin Architecture

5. FEATURES OF KOTLIN:

Kotlin compiles to JVM bytecode or JavaScript Like Java, Bytecode is that the compiled format for Kotlin programs also. Bytecode means Programming code that, once assembled, is run through a virtual machine rather than the computer's processor.

Kotlin can use all existing Java Frameworks and Libraries

Kotlin are often learned quickly, and it's approachable It are often rapidly learned by simply reading the language reference. The syntax is clean and intuitive(easy to use and understand).

6. JAVA ARCHITECTURE :

Java Architecture is a collection of components, i.e., JVM, JRE, and JDK. It integrates the process of interpretation and compilation. It defines all the processes involved in creating a Java program. Java Architecture explains each and every step of how a program is compiled and executed.

Java Architecture can be explained by using the following steps:

- There is a process of compilation and interpretation in Java.
- Java compiler converts the Java code into byte code.
- After that, the JVM converts the byte code into machine code.
- The machine code is then executed by the machine.

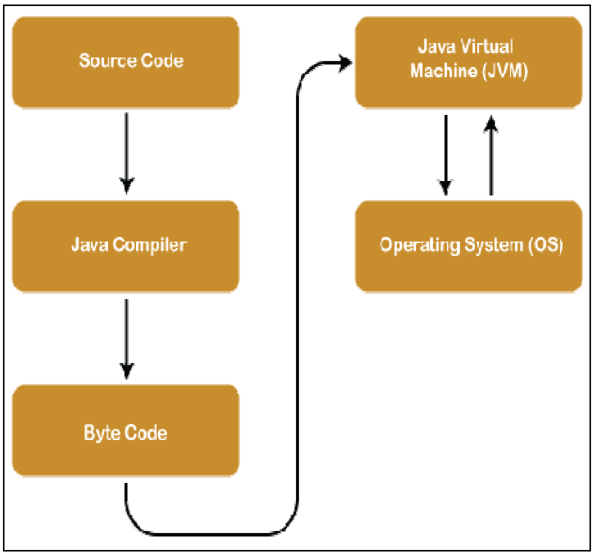


Figure 3: Java Architecture

7. FEATURES OF JAVA:

In Java, everything is an Object. Java may be easily extended since it's supported the item model.

Java is intended to be easy to find out. If you understand the essential concept of OOP Java, it'd be easy to master.

Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

8. THE PROBLEMS MANY COMPANIES FACE FOR USING JAVA FOR APPLICATION DEVELOPMENT:

Java comes with limitations prefer it causes problems with Android API design.

- Being a verbose language, Java requires writing more code because it carries a far better risk of bugs and errors.
- This linguistic communication is slower compared to many other languages since it requires many memory.
- The syntax that Java uses are going to be a small amount complicated or cumbersome while comparing to other languages like C + or Python.
- With Java, it's impossible to access certain content which is able to be incompatible with the device or equipment getting used.
- It isn't easy to access the new Java enhancements in mobile development.
- There are times when Java causes problems with Android API design.
- Test-Driven Development for Java requires writing more code and carries some way higher risk of programming errors and bugs
- Java could also be a small amount slower with relation to other programming languages and takes many system memory.

9. FUNCTIONS IN KOTLIN:

Function may perhaps be a gaggle of inter related block of code which performs a particular task. Function is employed to interrupt a program into different sub module. It makes reusability of code and makes program more manageable. In Kotlin, functions are declared using fun keyword. There are two types of functions counting on whether it's available in standard library or defined by user.

Standard library function

Kotlin Standard library function is built-in library functions which are implicitly present in library and available for use.

For example

```
fun main(args: Array<String>)
{
 var number = 25
 var result = Math.sqrt(number.toDouble())
 print("Square root of $number is $result")
}
```

User defined function

User defined function is a function which is created by user. User defined function takes the parameter(s), perform an action and return the result of that action as a value.

Kotlin functions are declared using the fun keyword. For example:

```
fun functionName() {
// body of function
}
```

10. CLASSES AND OBJECT IN KOTLIN:

Kotlin Class

Kotlin class is similar to Java class, a class is a blueprint for the objects which have common properties. Kotlin classes are declared using keyword class.

Syntax of Kotlin class declaration

```
class className { // class header
                   // property
                   // member function
                  }
```

Kotlin object

Kotlin object is created in two steps, the first is to create reference and then create an object.

var obj1 = className()

11. CONSTRUCTOR:

Kotlin has two types of Constructors primary as well as secondary constructor. Constructor is the keyword is used to declare the secondary constructor and it should always refer to the primary constructor.

Secondary constructors are not at all common in Kotlin language. We mainly use Secondary constructor is when we need to extend the class which provides multiple constructors that initializes the class in different ways.

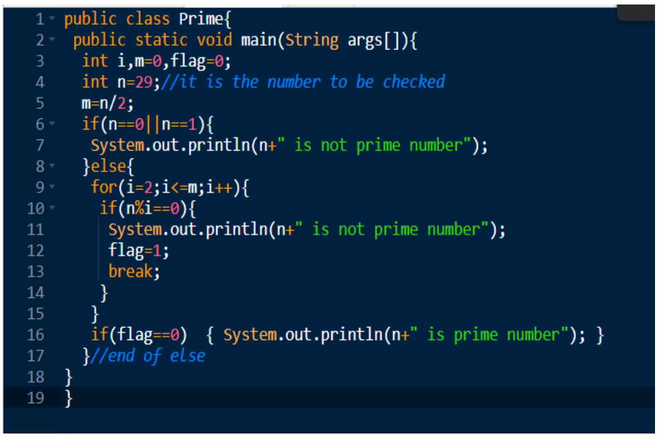
Following is an example of Secondary Constructor:

```
Class Login {
         constructor(data1: String) {
// some code to be written
}
constructor(data1: String, number: Int) {
//some codeto be written
}
}
```

This Secondary constructor feature is missing in Java as the main objective of Secondary constructor is it reduces the lines of code.

12. CODE DIFFERENCES:

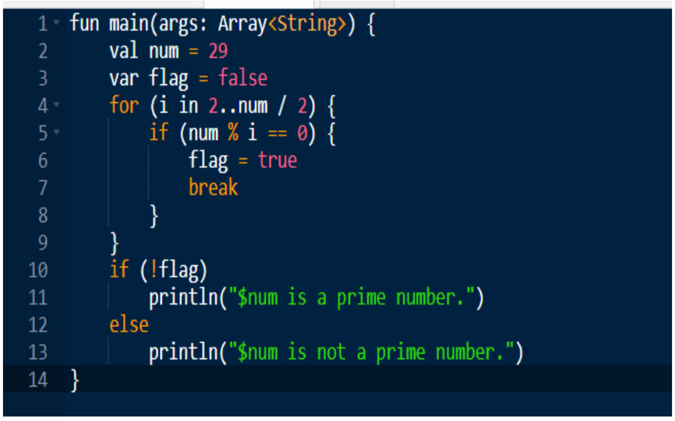
Developers now praise Kotlin for being concise. This was the quality which Java is not used to, however, java is more readable over concision. The Java code is big and time-consuming as compared to Kotlin as the Java code takes more number of code and as compared to Kotlin takes very less amount of code so that's why the developers are switching to Kotlin to save time and energy.



```
public class Prime{
 public static void main(String args[]){
  int i,m=0,flag=0;
  int n=29;//it is the number to be checked
  m=n/2;
  if(n==0||n==1){
   System.out.println(n+" is not prime number");
  }else{
   for(i=2;i<=m;i++){
    if(n%i==0){
     System.out.println(n+" is not prime number");
     flag=1;
     break;
    }
   }
   if(flag==0)  { System.out.println(n+" is prime number"); }
  }//end of else
}
}
```

Figure 4: Java Code

Above is the simple prime function which is written in Java. As seen in the above diagram the Java code takes 19 lines of code for a simple prime program. The code written is very lengthy and time-consuming as compared to Kotlin code which is shown in the diagram below.



```
fun main(args: Array<String>) {
    val num = 29
    var flag = false
    for (i in 2..num / 2) {
        if (num % i == 0) {
            flag = true
            break
        }
    }
    if (!flag)
        println("$num is a prime number.")
    else
        println("$num is not a prime number.")
}
```

Figure 5: Kotlin Code

It may not seem like much Java code but the Kotlin code of this calculator can be written with less code of lines of the code which took in Java code. The Kotlin code took only around 14 lines of code which is half of the Java code for a simple calculator.

Writing large and huge projects becomes easier when we write code in Kotlin as compared to Java. The Kotlin

Syntax is concise and readable.

13. ADVANTAGES OF KOTLIN:

Maximize the productivity

Kotlin may be a programming language that's supported Java. Kotlin can easily get obviate aggravations furthermore as obsolescence of Java. Kotlin could be a clear compact and dynamic language. Kotlin can maximize the productivity of the developer's team because it takes little time to write down and also you'll deploy it pretty fast.

Works with existing Java Code

Kotlin is interoperable with Java codes. Kotlin is persistent with Java and then many other related frameworks and tools. So switching to Kotlin is far easy. If the merchandise you're building can't only be written in Kotlin, you'll be able to use the opposite one and both will be used together easily.

Less Buggy

As we all know Kotlin could be a clear and compact codebase so it doesn't leave any space for creating any mistakes, it also helps to produce more stable codes in production. The compiler will identify every possible mistake at a compile-time span sans any tumult.

14. DISADVANTAGES OF KOTLIN:

Different from Java

Though Kotlin and Java have too many resemblances but still, there are some differences too. App developers just can't make the switch if they need inadequate knowledge about Kotlin.

Compilation Speed

In some cases, Kotlin works even faster than Java while performing incremental builds. But it should be remembered that Java will still stay incomparable after we speak about clean building.

Less Kotlin professionals

Kotlin has the sky-high popularity in developer's community still only few programmers are available during this field. the very fact about Kotlin is developers have to have in-depth knowledge about this language but nowadays it's really hard to search out experienced experts within the domain of Kotlin.

15. 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.

16. REFERENCES:

[1] 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

[2] Kotlin (Programming Language) - https://en.wikipedia.org/wiki/Kotlin_(programming language)

[3] AndroidDoc (2018a) Array map android developers. URL https://developer.android.com/reference/android/support/v4/util/ArrayMap, [Online; accessed17-July-2018]

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

[5] Atomic Kotlin is the book and associated learning resources by Bruce Eckel and Svetlana Isakova.

[6] Schildt, the Complete Reference Java, Seventh Edition, Chapter 1, page 9

[7] https://en.wikipedia.org/wiki/Kotlin_(programming_language)

[8] https://www.javatpoint.com/features-of-java

[9] https://www.udacity.com/course/kotlin-bootcamp-for-programmers--ud9011

[10] https://kotlinlang.org/