University of Waterloo

Faculty of Mathematics

1B Work Term Report

Comparison between languages used for native android app development

Tacit Corporation

Position: Mobile Developer

Toronto, ON

Written by:

Fahad Adnan

20838180

1B Honors Computer Science

July 2020

Memorandum

To: PD11 Evaluators

From: Fahad Adnan

Date: July 7, 2019

Re: Work Term Report: Comparison between languages used for native android app development

---

This is my first work term report for my 1B work term titled: “Comparison between languages used for native android app development”.

I am currently employed by Tacit Corporation as a Mobile Developer Intern where I work as a part of the app development team. With the first stable release of Kotlin in February,2016 (Jetbrains, 2020) many companies have been considering switching to Kotlin for its primary development language. While Java was once the primary language used, now languages such as Kotlin are becoming more popular and with languages such as C++ with a lower memory footprint than Java the problem arises what language should be used for app development and what are the benefits of each. The purpose of this report is to compare the major android app development languages and determine if it is useful for Tacit Corporation and any android developer to consider switching to a more Kotlin centered approach rather than using the universally popular Java language. The report was written entirely by me using cited sources, where teaching assistants in the PD11 course helped me revise the report for possible issues.

From,

Fahad Adnan

20838180

Table Of Contents

Memorandum……...……...……

…….

Figures and Tables……...……...…….

Executive Summary ……...……...…….

1.0 - Introduction……...……...……

2.0 - Analysis ……...……...…….

2.1 Introducing ……...……...……...

2.2 Java ……...……...……...…….

2.2.1 Introducing Java …….……...……

2.2.2 Object Orientation in Java ……...……...……

2.2.4 Advantages of using Java……...……...……

2.2.4.1 Platform Independent & Frequent Use

2.2.4.2 Garbage Collection & Multithreading

2.2.5 Disadvantages of using Java……...……...…….

2.2.5.1 Strongly Typed and Lengthy Syntax

2.3 Kotlin……...……...……...……

2.3.1 Introducing Kotlin for Android Development

……...……...……...……..

2.3.2 Functional Programming in

Kotlin……...……...……..…..

2.3.2.1 Higher Order

Functions…….

2.3.2.2 Lambda Functions…….

2.3.3 Advantages of using Kotlin……...……...……

2.3.3.1 Concise Syntax, Type Inference & Smart casts

2.3.3.2 Null Safety

2.3.3.3 Cross Platform

2.3.3.4 Interoperability with Java

2.3.4 Disadvantages of using Kotlin……..……..……

2.3.4.1 Language was Released Recently

2.5 Performance of different languages ……..……..……

2.6 Migrating Java apps to Kotlin ……..……..……..

3.0 Conclusion……..……..…….

References ……..……..…….

Figures and Tables:

Figure 1: Software Engineering Goals……. 3 Figure 2: Comparison of traditional and functional programming…..…… 4
Figure 3: Java and Kotlin comparisons in compilation……. 3

Executive Summary

Choosing between languages for native app development is an important decision that can have long lasting impacts. While using Java for android development has been the standard for many years creating responsive and performant apps, it can slow down the android teams development process with its unnecessary syntax and large difference in comparison to web and ios development languages such as JavaScript and Swift. Kotlin, a mobile cross platform language alternative, allows companies to develop code for both iOS and android, making code comparisons much easier, while also reducing the amount of syntax required for coding. However, as it is relatively new its introduction comes with the issue of developers having to learn a new language while a large portion of libraries used in android development are written in java. There are also the introduction of languages that can be used alongside Java and Kotlin to aid in development.

This report compares and contrasts Java to new alternatives being brought. It analyses the advantages and disadvantages of each language used for android development before comparing how they hold up against each other in terms of performance and cost. Lastly, it considers the issues companies may have when switching over or incorporating new development languages.

The report finds that Kotlin is becoming more popular for android development and can be used alongside Java code making it the better alternative between the two. It also finds that Kotlin can prove beneficial as it can be used for IOS development alongside Android making it easier to develop for both. It finds that even if a well established project uses Java code incorporating Kotlin code does not require drastic refactoring of previous code and can prove beneficial.

1.0 Introduction

Mobile Applications have been becoming more popular with 2.56 million apps currently in the Google Play Store which hosts specifically Android apps exceeding both the Apple app store and Microsoft app store combined (Clement, 2018) . As the popularity of mobile applications increases, specifically android apps, there have been many updates to the android development platform to help developers make more responsive and useful applications. With all the aid the android studio team gives to developers apps have been becoming more impressive by the day.

However, when it comes to development choosing the language can shape your experience with the platform. Android provides a universal IDE, android studio, but it gives the option for users to choose which language to begin development with. The two main supported languages for development are Java and Kotlin, where traditionally most companies opt to use Java as it is an official language of Android development and is supported by Android Studio, however C++ is also officially supported (Android Developers, 2020d). Java has been an official language longer than Kotlin, and it is also popular outside of Kotlin development for many other purposes (Android Developers, 2020c). Kotlin is a new language for android development officially supported by Android since February 2016, it is also now Google's preferred language of choice, though it is not as widely used outside of Android Studio(Android Developers, 2020a). Apart from the two main languages Android Studio also supports C++ with the use of the Java NDK (Android Developers, 2020b), and various other languages using plugins.

This report will begin by explaining Java, the primary language used for android development and then comparing it to the new popular alternative, Kotlin, which has been gaining traction to see if the benefits of each and whether a company should adopt using a language that differs from the previous norm of using Java for android development.

2.0 Analysis

2.1 Introduction to Android

dAndroid is the largest mobile OS, operating system, based on the Linux Kernel designed for mobile touchscreens devices. Android is open source and is commercially sponsored by Google with the majority of android devices shipped with GMS, google mobile services (Android GMS, 2020). Although Java is based on the linux kernel it has been optimized for mobile devices to keep power consumption to a minimum by suspending operations when they are not in use (Android Memory Allocation, 2020). For example the ART , android runtime,and Dalvik virtual machine use paging and memory-mapping to manage memory with a garbage collector that keeps track of memory allocation and once it determines that a piece of memory is no longer being used it frees it for future use (Android Memory Allocation, 2020). There are many more specific features of Android such as hard limits on how much heap size an app can take. Applications which extend the functionality of android devices are written using the Android SDK, software development kit, which is written in Java. The Android SDK provides tools for developers to make their own apps with a debugger, software libraries, emulators and other tools (Android Developers, 2020b). In late 2014 Google released Android Studio, an IDE, integrated development environment, for app development. Traditionally the language of choice for app development was Java as the Android SDK was developed on the language, but C/C++ can also be used using the android NDK, native development kit, along with GO which is partially supported and Kotlin which is now fully supported.

2.2 Java

2.2.1 Introduction to Java

Java is a general purpose object-oriented programming language built on very low coupling, meaning that it has very few implementation dependencies making it very popular with a WORA, write once and run anywhere, design.(Tech Target, 2002). This means that compiled java code runs on all platforms that support java without a need for recompilation for the specific system. Java was released back in 1995 as a core component of Sun Microsystems and was later relicensed under the GNU general Public License making it open source software (Java, 2020). Later on the tech giant Oracle bought the company and recently released Java8 LTS(Long time support), a more secure version whereas older versions had unresolved security issues (Oracle 2016). The reason Java is so popular in Android development is because Android, built on the Linux kernel, is largely written in Java and the Android SDK, software development kit, uses the Java language as the basis for Android applications, however its bytecode runs on its own virtual machine optimized for low memory devices, specifically mobile devices.(Android Developers, 2019b)

2.2.2 Object Orientation in Java

Java is primarily a class-based OOP (object oriented programming) language meaning that it is based on the concepts of objects which contain data about that object, typically referred to as variables, and methods that you can perform on that object, referred to as methods or functions (Indeed, 2020). This makes Java very accessible to new developers as the top three languages for development in terms of repositories in github have an object oriented foundation (Github, 2019). Also, Java8 introduced Java developers to functional programming with lambda expressions (Friesen, 2018) meaning that although Java may have had limitations in the past it is now even more powerful.

With concepts in OOP such as Inheritance, where one class can inherit methods and properties of another, Encapsulation, where objects can be bundled into objects to stay organized and save space, Polymorphism, which allows multiple implementations depending on the inputs and Abstraction which keeps code simplified, OOP can prove very beneficial for projects (Indeed, 2020).

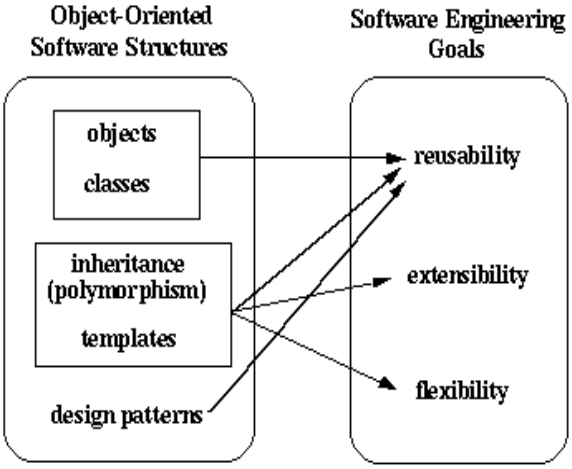


Figure 1: Software Engineering Goals: Shows the goals when creating software using core Object Oriented Programming Principles (Kafura, 1996)

The image in Figure 1, shows common software engineering goals that are accomplished using the core concepts of Object-oriented programming which have real life applications. For example as Tacit Corporation, the company I am currently interning at, makes applications for various companies there are portions of code that are reused and stored in modules to be reused, such as a user authentication module. These modules use OOP principles to remain reusable for multiple projects and easy to implement with low coupling and a level of abstraction so that the module can be reused by someone who doesn't know the specifics of how it works. Low coupling means that the module does not depend on many other modules meaning that the code is organized and can be picked up and reused easily (GeeksforGeeks, 2020). This concept speeds up development which is especially important in a startup such as Tacit making an OOP language such as Java very useful for Android development.

2.2.3 Advantages Of Java

2.2.3.1 Platform Independent & Frequent Use

As mentioned previously, Java is a language designed to run anywhere with a JVM, java virtual machine. As various devices and operating systems are built differently the JVM was created so that any machine with it installed could run and execute java programs (Oracle, 2019). This made Java a very popular programming language and as it was created in the 1990s it has been around for such a long time that many developers have this as their language of choice(Java, 2020). Java has been used extensively in embedded systems, servers, mobile applications, web applications, for big data and much more meaning there are a larger number of java developers compared to kotlin (Team, 2020). As Java is a frequently used language it is typically easier to find developers familiar with Java compared to Kotlin making it a better option in terms of creating a team.

2.2.3.2 Garbage Collection & Multithreading

Along with allowing java code to be run essentially anywhere the JVM comes with additional features such as Garbage collection and Multithreading (Oracle, 2020). A major issue with C/C++ was that when memory was allocated to a value, developers would often forget to deallocate that piece of memory causing leaks, which made software with more bugs and reduced performance(Memory Leak, 2020). Java solved this issue with garbage collection to automatically deallocate memory when it was no longer in use, this was very useful as it drastically reduces the amounts of memory leaks (Oracle, 2020). Multithreaded execution is an essential feature of the Java platform, where threads are independent sets of values and operations for a single core, making Java programs complete multiple tasks at the same time (Oracle, 2020). In Android Studio Java is able to run processes in the main thread to handle UI, user interface, updates and small operations along with making new threads to do other tasks such as network calls making the app more seamless (Android Developers, 2020b). Due to this Java is considered very powerful and is a popular choice amongst developers.

2.2.4 Disadvantages Of Java

2.2.4.1 Strongly Typed and Lengthy Syntax

From my personal experience using the language I know Java is a strongly typed language, meaning that it has stricter conditions when writing code. Although this has the benefit of catching many type errors in the compilation process, it makes writing Java code very lengthy. Java code was originally based on C/C++ code which has very lengthy syntax itself, making it more difficult for beginners to pick up and harder for developers to organize and write code (Java, 2020).

2.3 Kotlin

2.3.1 Introduction of Kotlin in Android Development

Kotlin is a cross-platform general purpose programming language built as an alternative for Java for Android Studio with Kotlin version 1.0 officially released for Android in February 2016 (Jetbrains, 2020). As Kotlin is a cross platform language compiled down to Android for android app development, Native for IOS app development and Javascript for Web application development making it beneficial for developers who learn the language (Jetbrains 2020). Kotlin is also 100% compatible with the JVM, the Java virtual machine, a virtual machine you install which allows you to run your java code on a device, meaning that Kotlin can be compiled and used anywhere Java is (Jetbrains 2020). Also during Google I/O of May, 2019 Google announced the Kotlin programming language is now the preferred language for Android app developers (Lardinois, F. 2019). Kotlin has become so popular that out of the top 1000 apps on the google play store 60% are developed in kotlin meaning that developers have really picked up the language for its benefits in a short span of 4 years since its release (Jetbrains, 2020).

2.3.2 Functional Concepts in Kotlin

Unlike how Java was initially built as an Object Oriented Language, Kotlin was made as a mix of both an object oriented and a functional language as it has capabilities of creating class based objects along with the benefits of a functional language using concepts such as lambdas(Jetbrains, 2020). Functional programming is a programming paradigm where programs are constructed by applying and composing functions.

2.3.2.1 Higher Order Functions

These specific types of functions allow developers to use functions in a similar manner to variables by taking functions as parameters and returning them (Abelson, Sussman, G. J. & Sussman, J, 2010). They can pass a specific function such as 'find the derivative' defined elsewhere and use it on variables within other functions. So for example if you wanted to iterate through each value of a list and apply a function instead of having a for loop and looking up each value in the list by its index you can use a functional programming function, map, that goes through each value in the list and applies a function to it, saving space by not having to store an iterator variable, a concept I learned taking CS136 at the university of waterloo taught by Adrian Reetz.

```
Traditional Imperitive Loop

     val listNum = [1,2,3,4,5,6]
     var sum = 0

     for(var i =0; i< listNum.length; i++){
        if(listNum[i] % 2 == 0){
          result+= (listNum[i]* 10)
        }
     }

Functional Programming (Higher Order functions)

     val listNum = [1,2,3,4,5,6]
     var sum = listNum.filter(n => n%2 == 0)
                      .map(a => a * 10)
                      .reduce((a,b) => a + b))
```

Figure2: Comparison of traditional and functional programming in kotlin created by me to show how functional programming is more concise (Fahad Adnan, 2020)

The simple comparison in Kotlin which adds all even values in the list multiplied by 10 shows how higher order functions can also make your code more concise

2.3.2.2 Lambda Functions

Functional programming also allows developers to use lambda functions, a function definition that is not bound to an identifier, another concept I learned taking CS136. Anonymous functions are often arguments being passed to higher-order functions, or used for constructing the result of a higher-order function that needs to return a function (Abelson, Sussman, G. J. & Sussman, J, 2010). This decreases the amount of code needed to be written and reduces memory required as if a function is only used once you can simply use a lambda function to avoid storing its name, a reference to the function. Although Kotlin provides many benefits as it is functional, the recent release of Java, Java8 also gives developers functional concepts, however Android only supports a subset of features for Java8 while fully supporting Kotlin (Android Developers, 2020b). For example, one functional feature I typically used when developing in Java is the higher-order function, foreach as whenever iterating through a list where index does not matter as it is good practice to not store an unnecessary iterator, Kotlin does have some additional features such as allowing static objects and functions to be defined at the top level of the package without needing a redundant class level (Android Developers 2020a).

2.3.3 Advantages of Kotlin

2.3.3.1 Concise Syntax, Type Inference & Smart casts

Another major benefit of Kotlin as opposed to Java is the concise nature of the code which removes a good majority of the boilerplate code, where boilerplate code simply refers to sections of code that need to be included with little or no altercation (freeCodeCamp, 2017).Kotlin has type inference meaning that some type information can be omitted and inferred by the compiler, a feature of many new languages now such as Typescript (Github, 2019). So instead of declaring a variable as an integer

or a string you can simply set the value and have the compiler set a value to it which saves time and space when writing code. Also, kotlin solves the issue of casting variables with smart casts, which allows developers to avoid explicit casting of values in cases where runtime types are guaranteed to conform to expected compile time types. (Android Developers, 2020c). For reference casting is when you convert a variable of one type to another, such as when you convert a string to an integer. All these features allow Kotlin to reduce its syntax making more readable code, according to Jetbrains, the creator of kotlin, rough estimates indicate approximately a 40% cut in the number of lines of code when switching from java to kotlin code (Android Developers, 2020a). Kotlin has additional features which makes its code more concise including data classes where the getter and setter methods are auto generated instead of needing to be manually made by the developer, small fixes like these make using kotlin very pleasant compared to Java which is very boilerplate heavy (Kotlin, 2020a).

2.3.3.2 Null Safety

Kotlin was originally designed with the issues of Java in mind, one particular issue which was such a problem that it was labeled the Billion Dollar Mistake by its own creator, the null reference (Hoare, 2009). In almost all languages values that are not given a valid value can be assigned a null value to let developers know the value is non-existent. However, this may be useful but it makes it extremely frustrating when a null value is sent as a parameter to a function expecting a type and getting a null value. If the programmer doesn't handle null inputs then any operations on the null value will most likely cause a crash and break the program (Hoare, 2009). This issue is infamous in Java for causing many issues for developers, so Kotlin used null safety which has been accustomed to new languages coming out(Kotlin, 2020b). In Kotlin, the type system separates values that can potentially hold a null reference and gives live errors if you are accessing a possibly null value, which is especially useful considering most null value exceptions go overlooked until testing occurs (Kotlin, 2020b). Kotlin also gives the benefit of allowing you to state that you know a value is not null using '!!' after the variable

name in its usage giving you the benefit of null safety and ignoring the errors in situations where you know the value is not null. (Android Developers, 2020b). This may seem like a small issue, however in my time using Java I would routinely fall into the issue of null pointer exceptions spending hours debugging my code only to figure out I did not check if my parameters were all non-null.

2.3.3.3 Cross Platform

When looking into mobile, Android along with IOS, and web development there are various languages used which makes writing code in one codebase for all three almost impossible. Kotlin solves this issue by being cross platform and having the benefit of being able to share modules across all three platforms (Kotlin, 2020a). Currently I am working on a major project for a food company that is going to be published on IOS, Android, and Web, where these three platforms will share many similar functionality such as ordering. Typically you would have to create an ordering module for each in the specified language, but with Kotlin you can make a shared module used by all three to organize code and simplify logic.(Kotlin, 2020a). Similar to Java, Kotlin has been made for various applications including server-side, mobile development, web development, data science, and Android meaning that its not limited to one application and can be beneficial for developers to learn.(Android Developers, 2020a)

2.3.3.4 Interoperability with Java

The biggest setback people seem to face when picking up a new language is migrating their old projects to the language, but Kotlin solves this issue as it is 100% interoperable with the Java programming language(Android Developers, 2020a) meaning that you can add as little Kotlin as you want to your project. This is especially useful when considering most libraries are written in Java, but with Kotlin you can still have the benefits of a newer language without having to sacrifice the thousands of libraries you already use. Kotlin was designed by Jetbrains with the concept of creating a language that solves issues developers had faced in Java so making it interoperable makes it even

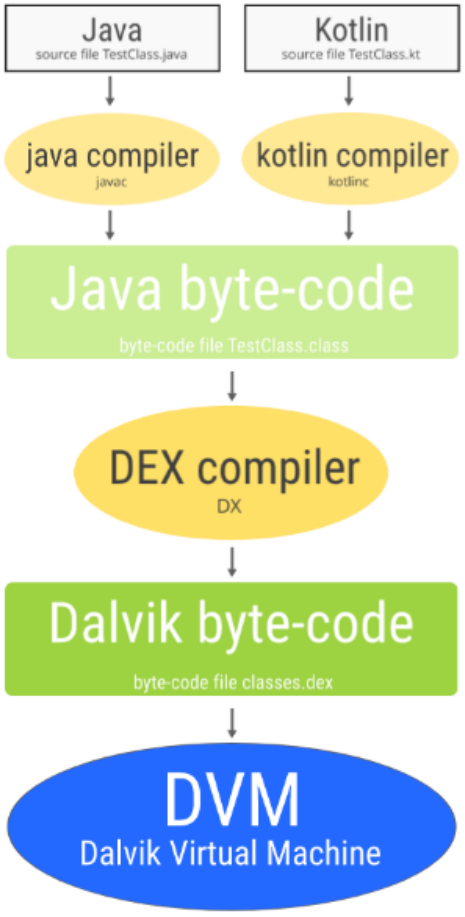
easier to add to your pre existing codebase. Also, as Kotlin was built to work alongside Java code it shares the same benefits in terms of most features including Garbage collection and multithreading.

2.3.4 Disadvantages of Kotlin

2.3.4.1 Language was Released Recently

Kotlin was designed specifically to fix the issues Java developers had with the language and although the language has brought upon many fixes to common problems with Java the issue still remains that Kotlin 1.0, the first version, was released only 4 years ago in February,2016 (Jetbrains, 2020). This means it is going to be harder to find developers that are well equipped using the language and although Kotlin is github's 4th largest language in terms of growth, the fact remains that it will be harder to find Kotlin Developers compared to android. Although the language is relatively new as it was built considering issues within the Java language it was also built in a structure which is easily learnt by Java developers, a feature I noticed first hand when learning the language for my Mobile Developer role.

2.5 Performance of different languages



For Android development the Java compiler converts your code to Java bytecode which is then used by the DEX compiler to become Dalvik byte-code for the Dalvik Virtual Machine used by Android to process code for your android application (Android Developers, 2020b). In a similar manner Kotlin code is converted to code which is interoperable with Java bytecode and then goes through the same process to become machine code for your android device (Markovic, 2020). Although there have been independent tests conducted which show that Kotlin is slightly slower than Java in compiling code, with the recent releases of Kotlin continually making updates to the language this isn't a major difference between the two.

Figure 3: Java and Kotlin comparisons in compilation process to DVM code for your device to run on. (Markovic, 2020)

2.6 Migrating Java Apps to Kotlin

As Kotlin is 100% Interoperable with Java you can use as little or as much Kotlin code in your application as you would like (Android Developers, 2020a). Typically most developers don't want to rewrite modules when switching to a new language which is why languages such as Kotlin, Typescript and Rust that focus on interoperability with pre-existing languages have been gaining traction (Github, 2019). So when migrating Java apps to Kotlin there is no issue regarding having the app down for a certain period of transition as you can incrementally add Kotlin to your project in modules and Kotlin will work with any Java libraries you are already using (Android Developers, 2020a). Typically for companies that are already using Java for their application they typically choose to add

kotlin modules incrementally while newer projects are made completely in Kotlin due to the benefits it provides.

3.0 Conclusion

In conclusion, both languages are good choices to start learning Android app development with Java having a long history of being a performant language with Android along with its many libraries and Kotlin with its improvements on Java and functional features. However, Kotlin is the clear winner when it comes to writing code due to its null safety checks, its type inference, smart casting, and cross platform design.

Whenever the chance presents itself developers should delve into Kotlin as it is likely the future of android development considering its growth over the past four years and with its interoperability with Java. For developers just learning about Android development I would suggest first learning Java as from personal experience almost all the code references and examples online are written in the language, making it easier for newcomers to learn about Android development. However, I believe it is essential to learn Kotlin once developers get basic concepts regarding fragments, activities and threads understood as the language provides much more support in terms of writing cleaner code that avoids many pitfalls with Java such as the infamous null pointer exception error.

Project developers that are currently using Java should consider adding some Kotlin modules to their code so that they don't have to migrate previous code while also getting the benefits of Kotlin. Later on if they decide to migrate the rest of their project they can do so incrementally.

As for experienced Android Developers about to start a new project I would highly suggest having a project that is coded mostly on Kotlin as the benefits it provides over Java are very useful for developers, so much so, that the project I am currently working on for a major food distribution

company has their android team working completely on Kotlin. Previously, older projects were majority Java, but my company has seen the benefits of Kotlin and have decided to migrate their expertise to Kotlin for newer projects.

Kotlin was created to solve issues in Java which is why I think that developers should utilize the language accordingly even though it is relatively new. The support from companies has also been a reassuring factor when releasing the language with Google labeling it the official language of Android in 2017 (Android Developers, 2020a). In conclusion, I believe newer developers should learn the basics of Android in Java due to the larger amount of resources available for learning Android in the language, whereas more experienced developers should learn Kotlin for the benefits it provides.

As for Tacit, throughout my time working at the company they have adopted a more Kotlin focused approach as the benefits of the language became clear with older projects slowly moving to Kotlin as well. This is very reassuring as I get to learn the language with a real life application in a product going to be released soon. Overall I believe that although Kotlin is relatively new it is a language that fixes many issues with the Java programming language and should be considered for new and old projects.

References

Abelson, H., Sussman, G. J., & Sussman, J. (2010). *Structure and interpretation of computer programs*. Cambridge, Mass: MIT Press.

Android Developers(2020) ： Kotlin . (n.d.). Retrieved July 09, 2020, from https://developer.android.com/kotlin

Android Developers (2020) :Developer Guides ： Android Developers. (n.d.). Retrieved July 09, 2020, from https://developer.android.com/guide

Android Developers(2020) Configure your build ： Android Developers. (n.d.). Retrieved July 09, 2020, from https://developer.android.com/studio/build

Android Developers (2020) Documentation :   Android Developers. Retrieved August 08, 2020, from https://developer.android.com/docs

Android GMS, A. (n.d.). Google Mobile Services. Retrieved July 19, 2020, from https://www.android.com/gms/

Android Memory Allocation, A. (n.d.). Overview of memory management :   Android Developers. Retrieved July 19, 2020, from https://developer.android.com/topic/performance/memory-overview

Clement, J. (n.d.). Topic: Mobile app usage. Retrieved July 09, 2020, from https://www.statista.com/topics/1002/mobile-app-usage/

FreeCodeCamp.org. (2017, December 23). What is boilerplate and why do we use it? Necessity of coding style guide. Retrieved July 24, 2020, from https://www.freecodecamp.org/news/whats-boilerplate-and-why-do-we-use-it-let-s-check-out-the-coding-style-guide-ac2b6c814ee7/20

Friesen, J. (2018, October 30). Functional programming for Java developers, Part 1. Retrieved July 09, 2020, from https://www.infoworld.com/article/3314640/functional-programming-for-java-developers-part-1.html

GeeksforGeeks, G. (2020, January 08). Software Engineering: Coupling and Cohesion. Retrieved August 08, 2020, from https://www.geeksforgeeks.org/software-engineering-coupling-and-cohesion/

Haskell, H. (n.d.). Higher order functions. Retrieved July 09, 2020, from http://learnyouahaskell.com/higher-order-functions

Hoare, T. (2009, August 25). Null References: The Billion Dollar Mistake. Retrieved July 23, 2020, from https://www.infoq.com/presentations/Null-References-The-Billion-Dollar-Mistake-Tony-Hoare/

Indeed, O. (n.d.). What Are the Four Basics of Object-Oriented Programming? Retrieved August 08, 2020, from https://www.indeed.com/career-advice/career-development/what-is-object-oriented-programming

Java, O. (2020). History of Java - Javatpoint. Retrieved August 08, 2020, from https://www.javatpoint.com/history-of-java

Jetbrains, K. (2020). FAQ. Retrieved July 09, 2020, from https://kotlinlang.org/docs/reference/faq.html

Kotlin (2020). A Kotlin for cross-platform mobile development. (n.d.). Retrieved July 24, 2020, from https://www.jetbrains.com/lp/mobilecrossplatform/

Kotlin, K. (2020). Null Safety. Retrieved August 08, 2020, from https://kotlinlang.org/docs/reference/null-safety.html

Lardinois, F. (2019, May 07). Kotlin is now Google's preferred language for Android app development. Retrieved July 09, 2020, from https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/

Markovic, B. (2020, January 28) Process of compiling Android app with Java/Kotlin code Retrieved July 24, 2020, from

Memory Leak, L. (2020). JScript Memory Leaks. Retrieved July 24, 2020, from https://web.archive.org/web/20121207132137/http://javascript.crockford.com/memory/leak.html

https://medium.com/@banmarkovic/process-of-compiling-android-app-with-java-kotlin-code-27edcfcce616

Object-Oriented Programming and Software Engineering. (n.d.). Retrieved July 09, 2020, from http://people.cs.vt.edu/~kafura/cs2704/oop.swe.html

Oracle, J.(2016, June 30). Retrieved July 09, 2020, from https://www.java.com/en/download/faq/remove_olderversions.xml

Oracle, O. (2019, October 22). Java Virtual Machine Guide. Retrieved July 24, 2020, from https://docs.oracle.com/en/java/javase/11/vm/java-virtual-machine-technology-overview.html

Oracle, O. (2020). Integrated Cloud Applications and Platform Services. Retrieved July 24, 2020, from https://www.oracle.com/

Team, D. (2020, June 11). 10 Astonishing Applications of Java - Where Java is Used in Real World? Retrieved July 24, 2020, from https://data-flair.training/blogs/applications-of-java/

TechTarget. (2002, May 02). Write once, run anywhere? Retrieved July 09, 2020, from https://www.computerweekly.com/feature/Write-once-run-anywhere

The Most Popular Mobile App Development Languages. (2020, April 13). Retrieved July 09, 2020, from https://buildfire.com/programming-languages-for-mobile-app-development/

The State of the Octoverse. (2020). Retrieved July 09, 2020, from https://octoverse.github.com/