

Discussion-1 What are the advantages of developing Android applications using Kotlin compared to other programming languages

Discussion Topic:

Please choose one of the following questions to discuss in your initial post:

- What are the advantages of developing Android applications using Kotlin compared to other programming languages?
- How does Android Studio help developers in the Android application development process?
- What are the key differences between Kotlin and Java in terms of syntax and features?

My Post:

Hello Class,

For this discussion, I chose the following topic: What are the advantages of developing Android applications using Kotlin compared to other programming languages?

When developing an Android app, one of the first decisions a developer faces is choosing a programming language. Kotlin, a relatively new language, has quickly become the preferred choice for many. Kotlin has been endorsed by Google since 2017, meaning that it is the official programming language for Android application development (Samojlo, 2024). Google's parent company, Alphabet, owns Android, and Google manages it as a subsidiary. Additionally, over 95% of the top 100 Android apps use Kotlin (Kotlin, n.d.). This post explores the advantages of developing Android applications using Kotlin over other programming languages.

Kotlin is described as the most succinct language, therefore making the least error-prone (Horton, 2019). While Android SDK is largely written in Java, Kotlin is fully interoperable with Java, meaning that Java libraries and frameworks can be integrated into Kotlin, and Java projects can be migrated to Kotlin. Additionally, Kotlin has the following features:

- It is an Object-Oriented Programming language (OOP).
- It includes null safety, preventing null pointers.
- It allows function extensions, that is adding functionality to existing classes without modifying their source code.
- It allows data classes which are classes primarily used to hold data.
- It can implement coroutines, making asynchronous programming much easier such as handling network requests.

When compared to C++ and Java, Kotlin has significant advantages for Android development, the table below lists some of these advantages.

Table

Kotlin's advantages over C++ and Java

Advantages Over Java	Advantages Over C++
<ul style="list-style-type: none"> - More concise syntax <p>Reduces boilerplate, more readable, and more maintainable code. (Example: "Hello World" comparison). Less verbose than Java.</p>	<ul style="list-style-type: none"> - Easier to use <p>Generally easier to learn and use, especially for Android development. Simpler syntax and modern language features reduce the learning curve. No pointers.</p>
<ul style="list-style-type: none"> - Null safety: <p>Built-in null safety in the type system prevents <i>NullPointerExceptions</i>, a major source of crashes in Java. Explicit nullable types (?) force handling of potential null values.</p>	<ul style="list-style-type: none"> - Less Complexity: <p>Automatic memory management (garbage collection) eliminates manual memory management.</p>
<ul style="list-style-type: none"> - Extension functions <p>Allows adding functionality to existing classes without inheritance.</p>	<p>C++ has mechanisms like free functions and operator overloading, but they are not directly comparable to Kotlin's extension functions in terms of Android integration.</p>
<ul style="list-style-type: none"> - Coroutines support <p>Built-in support for asynchronous programming simplifies handling background tasks and improves UI responsiveness.</p>	<p>C++ has libraries for concurrency, but Kotlin's coroutines are more tightly integrated with the Android ecosystem and generally easier to use for common Android tasks.</p>
<ul style="list-style-type: none"> - Data Classes <p>Automatically generates common methods (equals(), hashCode(), toString(), copy()) for classes primarily holding data, reducing boilerplate.</p>	
<ul style="list-style-type: none"> - Jetpack compose support: <p>Kotlin is the preferred language for Jetpack Compose which is an Android's recommended modern toolkit for building native UI.</p>	<ul style="list-style-type: none"> - Better Android integration <p>Tighter integration with the Android ecosystem (Android Studio, libraries, frameworks). Better tooling and support for Android development.</p>
<ul style="list-style-type: none"> - Faster development <p>Concise syntax and a fail-fast compiler speed up development cycles. Early error detection reduces debugging time.</p>	<ul style="list-style-type: none"> - Generally faster development <p>The combination of ease of use, reduced complexity, and better Android integration often leads to faster overall development compared to C++.</p>
<ul style="list-style-type: none"> - Scripting <p>Can be used as a scripting language for tasks like build automation.</p>	<p>C++ can be used with scripting languages, but Kotlin itself can act as a scripting language.</p>
<ul style="list-style-type: none"> - Java Interoperability: <p>Allows using existing Java libraries, gradual migration from Java, and mixed Java/Kotlin projects.</p>	<p>Kotlin's interoperability is specifically with Java.</p>

Note: The table lists several advantages that Kotlin has over C++ and Java. From several sources (Kotlin n.d.; Horton, 2019; Samojsło, 2024; Ramos, 2023, Rkumaraj, 2023, Developers, n.d.; Berga et al, 2019; Lotarev I. 2024; Zaręba, 2024)

In conclusion, Kotlin has several advantages for Android development over other languages OOP such as more concise syntax, null safety, coroutines support, interoperability with Java, Jetpack Compose support, and function extensions; moreover, it is endorsed by Google as the official programming language for Android application development. Overall Kotlin provides a more modern, safer, and more productive approach to developing Android applications than other programming languages.

-Alex

References:

Berga, M., Figueiredo, R., & Franco, T. (2019, April 19). *Kotlin vs Java: the 12 differences you should know*. Imaginary Cloud. <https://www.imaginarycloud.com/blog/kotlin-vs-java>

Developers (n.d.). *Android's Kotlin-first approach*. Developers. <https://developer.android.com/kotlin/first>

Horton, J. (2019). *Android programming with Kotlin for beginners*. Packt Publishing. ISBN: 9781789615401

Kotlin (n.d.). Kotlin for Android. *Kotin overview*. Kotlin. <https://kotlinlang.org/docs/android-overview.html>

Long, M. (2023, June 28). *Can C++ be used for Android app development?* Groove Technology. <https://groovetechnology.com/blog/software-development/can-c-be-used-for-android-app-development/>

Lotarev, I. (2024, January 16). *Kotlin vs Java: Which is better for Android app development?* Adapty. <https://adapty.io/blog/kotlin-vs-java/>

Ramos, M. (2023, September 18). *Kotlin vs Java for mobile and Web development*. Kinsta. <https://kinsta.com/blog/kotlin-vs-java/>

Rkumaraj. (2023, May 8). *Why Kotlin is the best choice for Developing Android Apps*. Medium. <https://medium.com/@rkumaraj5694/why-kotlin-is-the-best-choice-for-developing-android-apps-7ce306fcc6ec>

Samojsło, G. (2024, November 26). *Pros and Cons of Kotlin for Android App Development*. Netguru. <https://www.netguru.com/blog/kotlin-pros-and-cons>

Zaręba, G., Zarębski, M., & Smołka, J. (2024). C++ and Kotlin performance on Android – a comparative analysis. *Journal of Computer Sciences Institute*, 30, p. 21–25. <https://doi.org/10.35784/jcsi.5299>