International Journal of Computer Applications Technology and Research
Volume 13–Issue 03, 07 – 09, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1303.1002

Kotlin Programming Language: A Comprehensive Overview of Definition, Applications, Advantages, and Limitations

Mohammed Saad Talib
College of Administration and Economics, University of Babylon,
Babel – Iraq

Alyaa Abdul-Moneim Al-Najar
College of Administration and Economics, University of Babylon, Iraq

Ali Huseein Hassan
College of Administration and Economics, University of Babylon,
Babel – Iraq

Zainab Saad Talib
College of Science, University of Kerbala
Kerbala, Iraq

Abstract: Kotlin is a relatively new, fast-growing programming language known for its brevity, safety, and interoperability with Java. Kotlin is a statically typed, object-oriented programming language that is designed to interoperate fully with Java and run on the Java Virtual Machine (JVM) along with other platforms. The language is aimed at being more concise, expressive, and safe than Java.This scientific research paper endeavors to explore various aspects of Kotlin, such as its definition, key features, its drawbacks, its Fundamental and its applications.

Keywords: Kotlin Programming Language, Java Interoperability, Null Safety, Android, IOS.

1. INTRODUCTION

Kotlin is a programming language that is becoming increasingly popular in mobile application development. It offers developers a robust, concise syntax and a wide range of features that allow them to write high-quality code quickly and efficiently. Kotlin, a newly developed programming language (v1.0 released in 2016), was officially adopted by Google in 2017 [1][2][3] for use in creating Android applications. So that, the Kotlin programming language has allowed Android developers to create Android apps [4]. Kotlin is relatively modern programming language. Kotlin was announced by JetBrains in 2011 for Android mobile devices. Kotlin program language is regarded as being native languages for mobile app development for their respective platforms[5]. This is thus important and popular language in the economically-booming mobile app market.

As is the trend for modern languages, Kotlin is classed as being multi-paradigm language [6]. Kotlin is considered to be an excellent choice for creating apps for Android devices due to its interoperability with Java, allowing developers to easily integrate existing Java libraries into their applications [1].

Programming language Kotlin mixes functional and object-oriented features, some of which aren't available for Android development or aren't present in Java [4]. Furthermore, Kotlin is also easy to learn and use, meaning developers can get up-to-speed on creating mobile apps using this language quickly. Using the powerful capabilities of Kotlin, developers are able to create rich, dynamic mobile applications with fewer lines of code than other languages [7].

This helps them reduce development costs and timeframes while still offering users the reliability and performance they need in their applications.

Also, providing interactive demonstrations and games that people can play with minimal effort can help with the rapid prolif- eration of new Game AI ideas and is clearly desirable. Kotlin is a powerful modern programming language and can provides a great starting point for writing cross-platform games [8].

2. FUNDAMENTAL CONCEPTS OF KOTLIN

Kotlin has several fundamental concepts which includes classes and objects, functions and lambdas, control flow, strings and regular expressions, and collections. The class is a blueprint or template that defines the state and behavior of an object. Functions and lambdas offer a more concise way of defining behavior[6].The development of Kotlin/JS was straightforward. The idea of storing functions instead of Algebra references was derived from the Java bytecode backend in which the generated function is assigned a similar constants array storing both constant values of the expression and function references [9].

Control flow determines the order in which statements are executed. Strings and regular expressions are used to manipulate text[4]. Collections are used to hold groups of related objects[10][5]. Also, unary Operation Function and binary Operation Function companion functions were added, which both return an object of the Kotlin function type instead of the value of the operation [9]. Kotlin provides a scripting engine for compiling expressions on-the-fly [9].

3. KOTLIN'S KEY FEATURES

Table 1: Features of Kotlin programming language [11][12]

Feature	
Lambda expressions and Inline functions	YES
Extension functions	YES
Checked exceptions	no
Null-safety and smart casts	yes
Primary constructors	yes
First-class delegation	yes
Type inference for variable and property types	yes
Wildcard types	no
Declaration-site variance and Type projections	yes
Range expressions	yes
Operator overloading	yes
Ternary operator	no
Data classes	Requires less code compares with other languages such as java
Static members	no
Coroutines	yes

Kotlin offers several features that make it a popular programming language explained as follows [7][13][14][15].

1. Concise syntax: Kotlin has a concise syntax that makes it easy to read and write. It is less verbose than Java, which makes it easier to write code faster. Kotlin has also eliminated some of the boilerplate code that is required in Java.
2. Null safety: Kotlin has a nullable type system that allows developers to write safer code. This means that developers can prevent null pointer exceptions, which can be a major source of bugs in many programs.
3. Interoperability: Kotlin is interoperable with Java, which makes it easy to use with existing Java code. This means that developers can use Kotlin and Java together in the same project without any major issues.
4. Performance: Kotlin has comparable performance to Java, which means that developers can write code that is just as fast and efficient as Java.

Benefits:

1. Improved productivity: Kotlin allows developers to write code faster and more efficiently. Its concise syntax and null safety features help reduce the amount of code that needs to be written, which saves time and improves productivity.
2. Better code quality: Kotlin's null safety and type inference features help developers write code that is less error-prone. This means that the resulting code is of higher quality and is less likely to contain bugs.
3. Improved team collaboration: Kotlin's interoperability with Java means that teams can work together more easily. Developers can use Kotlin and Java together in the same project, which means that team members with different skill sets can collaborate more effectively.
4. Future-proofing: Kotlin is a modern programming language that has been designed with the future in mind. It is continuing to evolve and improve, which means that developers can use it with confidence knowing that it will continue to be supported and updated.

4. DRAWBACKS

Kotlin offers some drawbacks that explained as follows[16][17]:

1. Lack of backward compatibility: One of the significant drawbacks of Kotlin is its lack of backward compatibility with earlier versions. This means that developers must update their code to the latest version of Kotlin to avoid compatibility issues and to leverage new features.
2. Slow build times: Another significant limitation of Kotlin is its slow build times compared to other programming languages. This is caused by its complex type inference system, which requires more time to build than simpler languages.
3. High learning curve: Kotlin has a steeper learning curve than other programming languages such as Java and Python. This is due to its complex syntax and type inference system, which may take some time for developers to master.

5. APPLICATIONS OF KOTLIN

Kotlin is widely used in developing mobile applications, backend web applications, and server-side applications. It is a popular choice for developing Android applications due to its interoperability with Java and its concise and expressive syntax. Kotlin also supports multi-platform development, which allows developers to write code that runs on multiple platforms like Android, IOS, and web platforms.

There are many uses for similarities programming languages such as Kotlin and Java as shows in Table I.

6. APPLICATIONS OF KOTLIN

Table 2: Data type similarities between Kotlin and Java [11]

Data Type	Kotlin Java	Kotlin Java
Integer	Integer Byte,	Integer Byte,
Floating point Boolean	byte, short, int,	byte, short, int,
Alphanumeric character	Short, Int,	Short, Int,
Alphanumeric string	long	long
Null object Base class	Long	Long
Data Type	Float, Double float, double Boolean	Float, Double float, double Boolean
Integer	boolean	boolean

7. CONCLUSION

Kotlin is a modern programming language that offers many advantages and benefits over other programming languages. Its concise syntax, null safety, interoperability, and performance make it an ideal choice for many software development projects. As Kotlin continues to evolve and improve, it is likely to become even more popular in the future. Kotlin also has some drawbacks and limitations that developers should consider before using it. These limitations include the lack of backward compatibility, slow build times, and high learning curve. Despite these limitations, Kotlin remains a popular choice for many developers due to its many advantages..

8. REFERENCES

[1] S. Bose, A. Kundu, M. Mukherjee, and M. Banerjee, “A COMPARATIVE STUDY : JAVA VS KOTLIN PROGRAMMING IN ANDROID APPLICATION DEVELOPMENT,” Int. J. Adv. Res. Comput. Sci., vol. 9, no. 3, pp. 41–45, 2018.

[2] A. PELEŠ, M., JEVREMOVIĆ, S., SIMOVIĆ, A., & HADŽIĆ, “Possibilities for developing and implementing a mobile application for recognizing the shape of the environment, text, and reading QR codes using the Android CameraX framework and the Machine Learning Kit,” Ann. Spiru Haret Univ. Econ. Ser., vol. 21, no. 4, pp. 163–179, 2021.

[3] P. M. L. e Silva, “SKot: a Web-based Structured Code Editor for Introductory Programming in Kotlin,” 2022.

[4] M. Martinez and B. G. Mateus, “Why did developers migrate Android applications from Java to Kotlin ?,” in IEEE Transactions on Software Engineering, 48(11), 2021, pp. 4521–4534.

[5] V. Oliveira and F. Ebert, “On the Adoption of Kotlin on Android Development : A Triangulation Study,” in IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2020, pp. 206–216.

[6] S. Bonev and J. Galletly, “Functional programming features supported by Kotlin and Swift,” Electrotech. Electron. (E+ E), vol. 54, no. E+E, pp. 98–104, 2019.

[7] G. Hecht and A. Bergel, “Quantifying the adoption of Kotlin on Android stores : Insight from the bytecode,” in IEEE/ACM 8th International Conference on Mobile Software Engineering and Systems (MobileSoft), 2021, pp. 94–98.

[8] S. M. Lucas, “Cross-Platform Games in Kotlin,” in IEEE Conference on Games (CoG), 2020, no. 242, pp. 774–775.

[9] I. Postovalov, “Compilation of mathematical expressions in Kotlin,” arXiv Prepr. arXiv2102., vol. 07924, pp. 1–6, 2021.

[10] P. Sommerhoff, Instructor ’ s Manual for Kotlin for Android App Development Instructor ’ s Manual, no. c. 2019.

[11] N. Dimitrijevic, Nemanja Zdravkovic, and Vladimir Milicevic, “An Automated Grading Framework for the Mobile Development programming language Kotlin,” Int. J. Qual. Res., vol. 17, no. 2, pp. 313–324, 2022.

[12] G. Piskachev, F. Iem, and F. Iem, “To what extent can we analyze Kotlin programs using existing Java taint analysis tools ? (Extended Version),” 2022.

[13] N. Everlönn, S. Gakis, and A. Nilsson, “Java and Kotlin, a performance comparison,” 2020.

[14] R. Calegari, G. Ciatto, E. Denti, A. Omicini, and G. Sartor, From Objects to Agents, vol. 2706. 2020.

[15] R. Terra, E. Cirilo, and R. S. Durelli, “Are you still smelling it ?,” in Proceedings of the VII Brazilian symposium on software components, architectures, and reuse, 2018, pp. 23–32.

[16] D. Stepanov, M. Akhin, and M. Belyaev, “ReduKtor : How We Stopped Worrying About Bugs in Kotlin Compiler,” in 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2019, pp. 317–326.

[17] M. Peters and G. L. Scoccia, “How does Migrating to Kotlin Impact the Run-time Efficiency of Android Apps ?,” in IEEE 21st International Working Conference on Source Code Analysis and Manipulation (SCAM), 2021, no. September 2021, pp. 36–46.

International Journal of Computer Applications Technology and Research
Volume 13–Issue 03, 07 – 09, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1303.1002