International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 2, Issue 2, 30-38, 2021
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262/IJAIDSML-V2I2P104

Original Article

Comparing Java with Python and Kotlin: A Comprehensive Analysis of Performance, Syntax, and Usability

Santhosh Chitraju Gopal Varma
Software Developer, United States of America (USA).

Abstract *- Computer programming languages have greatly advanced in the fields of software development offering paradigms and capabilities. Three programming languages that are at the top include java, python, and Kotlin, and each of them is even notable for certain reasons. Many corporations, Android Application and web service development use Java as their programming language. Python, which is an easy to learn scrip language, is widely used in data science, defensive, Artificial intelligence and automation. Kotlin is another language developed as an upgrade to Java, which is compatible and shorter, more easily understandable, and popular in the development of Android applications. This paper provides a comparison of these three languages depending on the aspects such as speed, grammar, efficiency, and practical applicability. Measures include benchmark, code readability, learning curve, he community, and execution speed. Furthermore, literature review, research method, experiment, result, and conclusion of each language are discussed in the work. It has been ascertained that Java is optimally efficient and secure; Python has eased the process of coding and utility; Kotlin adds new-age improvements to the Java environment. In the last section, the best approach is presented for choosing the language specifically for the intended purpose.*

Keywords *- Java, Python, Kotlin, Programming languages, Performance, Syntax, Usability, Software Development.*

1. Introduction

The software development industry is dynamic and always comes up with new programming languages to suit the emerging issues. Java, Python and Kotlin: three languages for various fields of application. Java is the language that was developed by Sun Microsystems in 1995 and used in the development of a number of enterprise applications to this present time. It however, has prudent and practical attributes such as the “write once run anywhere” technology that makes it suitable for massive use. [1-4] Python is another timeless language that was initiated by Guido van Rossum in the late 1980s, and the feature typical of this language is its easy understandability. It is used a lot in machine learning, artificial intelligence and web development. It is an interpreted language, and this makes it suitable for quick development of an application’s prototype. Kotlin is statically typed language developed by JetBrains and released in 2011, it is safe in terms of null handling and has concise syntax. It is predominantly used in Android development replacing Java in a number of instances because of the expressive and enhanced security it offers.

1.1. Importance of Comprehensive Analysis of Performance, Syntax, and Usability

Java and C++ are two of the most popular programming languages and an attempt has been made to compare the two extensively in this paper. There are mainly three factors which have regard to effectiveness, robustness and the extent of use of a language and those are: performance, syntax and usability. There are six main subcategories grouped under this category, which are discussed in detail below.

- **Performance:** There are certain measures that can be used to assess programming language implementation quality, such as faster performance or, to the contrary, slower execution time, usage of system memory, etc. In large-scale applications, slow execution implies both, reduced productivity and increased costs of operation, not to mention the unsatisfactory service delivery to the end-users. Java leverages the Just-In-Time compiling technology to make it run efficiently at the runtime, whereas in the case of Python, the interpreted code slows down the computation. In case of choosing the right language, some computational elements should be met, such as execution speed, response time, and flexibility.
- **Syntax Readability:** This is because syntax can dictate the suitability and efficiency in terms of the ease with which the code can be written, read or modified. Both, concise syntax and an easy-to-understand language significantly decrease time to complete the project and the number of mistakes. Python has been recognised by many for being friendly and easy to learn and especially when it is compared to Java which makes the development process time-consuming due to the number of code lines used. Kotlin makes the best of two worlds by avoiding a lot of redundant code but is, at the same

time, a strongly typed language. While analyzing syntax readability, the development teams are able to understand which language helps them to work more efficiently and reduces overall development insecurities.

- **Usability in Real-World Applications:** Each written programming differs in functionality based on the design and functionality of the language. Java is prevalent in the enterprise application market, Python is the best for AI and data science, Kotlin for the Android platform. This research enables developers and those handling businesses to discuss the specific language that will suit the project best depending on the project's demands, with an emphasis on the efficiency of the language besides its appropriateness for the long run.

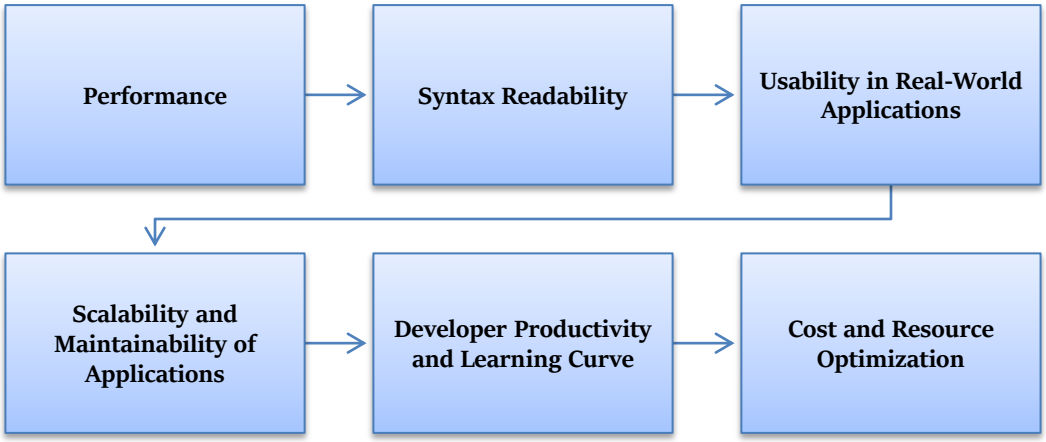

Fig 1: Importance of Comprehensive Analysis of Performance, Syntax, and Usability

- **Scalability and Maintainability of Applications:** Another factor which is crucial for achieving long-term success of the software projects is scalability and maintainability. Java, being a strongly memory managed language with very good support for multi-threading is perfectly suitable for large-scale applications. Although Python is a great language for expedited development especially for web apps, the language can be a problem when dealing with larger scales due to the speed of its execution. Kotlin has desirable features such as null safety and Java interoperation; consequently, it improves the maintainability of mobile and web applications. This assessment of the factors helps in satisfying the needs of newfound growth and development in the software systems.
- **Developer Productivity and Learning Curve:** The complexity of a particular language determines the ability of developers to master the language in a short timeframe. Python, specifically, is quite a simple language with little difficulty in the learning curve and therefore is perfect for tooling, prototyping. Java is a powerful language, but it takes a person more time to learn because of its grammar and rules. Kotlin makes Java better and easier to understand and work with from Java developers' perspective. Perceiving development effectiveness and learning slopes enable organizations to choose the right language that best suits the skill of the developers and organization's development objectives.
- **Cost and Resource Optimization** Selecting the type of the programming language also enhances development cost and consumption of resources. Java consumes less hardware resources compared to other higher categories while Python consume more system resources due to its low efficiency. Kotlin has numerous features that make the development of android applications easier and fast hence saves time hence saving cost. A proper analysis to ensure that the managers in the organizations can also make right decisions that do not strain the organization both in terms of cost and quality in terms of performance or size.

1.2. Comparing, Java and Python and Kotlin

Java language, Python language, and Kotlin are among the trending programming languages in the current world since they offer different benefits based on the efficiency, structure, and flexibility of their structure. Another mainstream language, often used for statically typed, object-oriented programming, Java supports enterprise applications, backend development and large-scale systems because of high speed of execution, security and expandability. Its JIT compilation is very efficient and thus recommended for business applications that need superior computing performance. [5-7] Nevertheless, there are some serious issues with the language that make moniker of 'code like a webpage' quite apt: Java is verbose, and this fact enhances both the time spent on development and its complexity.

Relatively, Python is another language commonly used in programming languages, particularly popular because it is dynamic type of language and quite easy to use. It is indeed an easy language with less syntax and this has made the language to be widely used in artificial intelligence, data science, machine learning, and scripting. Preferably its mechanisms must be flexible since the idea is to create functionality to check its performances in a short span of time, making it suitable for prototyping and

research models. But, it can also be interpreted in Python which takes more time when compared to Java and Kotlin, making this a disadvantage in the high execution rate applications. Kotlin, being developed as the modern language to replace Java, solves many flaws of the latter and is fully compatible with Java.

High level and small surface area, fewer bookkeeping expenses and make the developer life easier and more productive. Kotlin is particularly used in Android development more than in other areas because of its advanced features like null safety and operator functions for asynchronous operations. Although it has nearly Java's performance level, doesn't have the same ecosystem of Python in AI and data sciences. In general, the circumstances arising out of the contract seal project suggest that while Java and Kotlin have been used for Android, the language best suited to the development of the contract seal project is Python. Java is said to be the best when it comes to developing applications that are high performing in the enterprise, while Python is perfect for the new generation of machine-learning applications and rapid development. Such knowledge enables the developers to identify the strengths as well as weaknesses of a specific language so that they can make the right choice of the right language for the job.

2. Literature Survey

2.1. Existing Comparisons of Programming Languages

Albeit most of the researches were aimed to reveal the differences between Java, Python, and Kotlin concerning usability and performance. In the course of the literature review, compared Java and Python with focus on the aspect of runtime performance. Their research showed that Java has minimal interpretation and uses JIT compilation thus suitable for large application while Python could easily be developed since it is an interpreted language but this slowed down its speed. analyzed the application of Kotlin to the Android application development and it found that the Kotlin enhanced benefits to Java by slashing on the number of codes that are repeated hence enhancing the developers productivity. [8-10] The first one was about Kotlin and stressed out the possibility of its integration with the Java code, which is vital for current Android OS apps' development.

2.2. Performance Studies

One of the primary important characteristics of the implementation language is the performance since there are many applications in which response time and speed matter. A study by compared the parameters of Python, Java, and Kotlin on the time of their execution and concluded that, although Python is quite friendly to the programmer and provides the function of high-velocity development, it is slower due to the values of dynamic typing and interpreted implementation. Java is on the other hand advantages from a JIT compiler in which the code is optimized while in use and this makes it to run faster. To elaborate, Kotlin is evaluated to be statically typed and run in the Java Virtual Machine (JVM) comparable in performance to Java but with enhanced features in terms of syntax and safety.

2.3. Syntax and Readability

Maintainability alongside development speed directly correlates with how easily programmers can understand and utilize the programming language. The ease of use in Python programming becomes a major reason why beginner developers and academic institutions select Python for their projects. The wide usage of Java remains robust yet developers must write longer code to execute equivalent functions compared to Python and Kotlin. Kotlin minimizes the gap between readable syntax and type safety through its approach that maintains clear programming expressions. Modern language features of null safety and extension functions in Kotlin make the code less prone to runtime errors and boost developer productivity.

3. Methodology

3.1. Evaluation Criteria

The evaluation of Java, Python, and Kotlin uses three fundamental assessment factors which include performance capabilities and syntactical design elements and usability features. [11-15] Various applications benefit from these essential criteria because they enable proper evaluation of language suitability across different program scopes.

- **Performance:** Any programming language requires performance to be a primary factor because it determines the speed of execution and system memory utilization during runtime. Java delivers exceptional performance and memory optimization through JIT compiler and JVM which enables efficiency for enterprise-level applications. The combination of being interpretable and dynamically typed in Python leads to extended runtime processing and slower execution speeds despite its user-friendly development approach. Kotlin delivers performance levels comparable to Java on the JVM because it uses static typing but includes various speed-enhancing optimizations that work distinct from memory usage expansion.
- **Syntax:** The structural signs also relate to the issue of programming communication and obligatory uses of various syntax components, making code more or less verbose. It is popular for its simple and clear code line which makes it very easy for the programmer to write and read. On the other hand, Java is said to be verbose since this requires more code to write the same functionality as other languages. Here, Kotlin more or less adapts a middle ground by offering terse code while

keeping the compile time type checking intact. These include Type Inference, Extension Functions, and Null Safety which goes again to show that Kotlin aim to minimize on the creation of (unwanted) unnecessary complexity.

- **Usability:** The term usability can be defined as how practical a programming language is when applied to a great number of areas and domains. Nowadays Java is one of the leading coding languages in business applications, backend, and Android application development owing to its stable environment and available resources. Python is one of the most popular languages that offers the user-friendly environment for data analysis, machine learning, and development of web applications. Kotlin, though emerging recently has become more popular in the Android development as it provides better features than Java which optimizes the development time and can easily integrate with the existing Java applications. The usage of each language also depends on the project requirements, gained knowledge of the developers, and the actual domain of application.

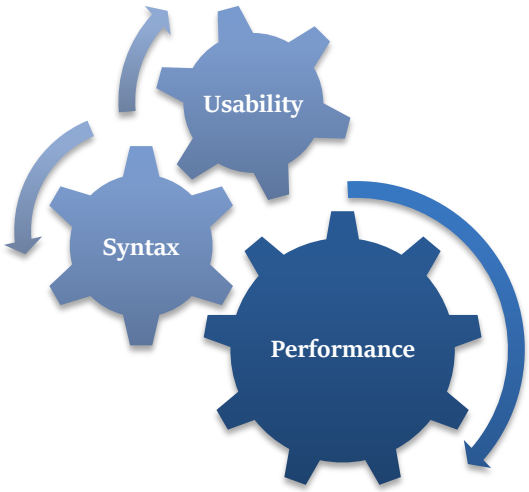

Fig 2: Evaluation Criteria

3.2. Benchmarking Methodology

To facilitate the conclusion on the relative efficiency of Java, Python, and Kotlin, three different sets of benchmarking were carried out using computation algorithms as well as real life examples. These benchmarks can give an understanding of each language complexity such as sorting, file handling capability and APIs to respond to questions. They would therefore help in evaluating their suitability for performance enhancing tasks.

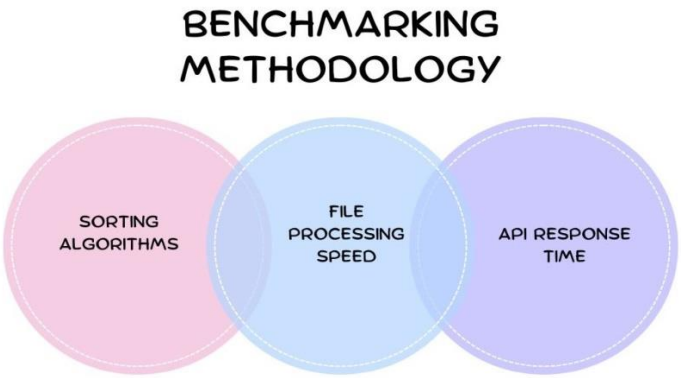

Fig 3: Benchmarking Methodology

- **Sorting Algorithms:** Sorting algorithms are commonly used to evaluate the competence of a program since input and output data have to be sorted and compared during the computational process. Bubble sort is a simple technique, though not very efficient with a time complexity of $O(2)$, helps in testing the baseline times. Quick Sort, as mentioned before, is a bit more efficient with the average time complexity of $O(n \log n)$ – making it is a more appropriate test to use when comparing the performance of every language when it comes to handling larger data. Java with JIT compilation usually performs better than Python, which works through interpretation. Kotlin is also the JVM-based language that is almost as fast as Java but has a much simpler syntax.

- **File Processing Speed:** Management of files is very crucial in most programs in use today, be it data processing or automating a system. In order to assess the file processing speed tests were conducted on areas of reading and writing big text and binary documents as well as modification of the file. Another reason that favours the use of Java is versatility; the language provides efficient I/O streams and efficient threading and always proves to be faster when dealing with large files. Even though Python is one of the easiest languages to use and has the functions to manage files, it is only slightly faster because of its slow dynamic typing. As a scripting language that brings out an enhanced version of Java I/O, Kotlin balances between efficient coding and high performance.
- **API Response Time:** API response time benchmarking gives information about the ability of a language in handling request and response from other web services. This test involved sending requests to a RESTful API and the amount of time that was taken to parse the results obtained. Java, particularly its current version, Spring Boot, offers efficient networking capabilities that result in low response times. Python is popular in web development while the Flask and Django frameworks, and is slower because it is an interpreted language but it is excellent in rapid application development. Kotlin's compatibility with the various Java-based web frameworks makes it rather performant while giving more concise code. This benchmarking metric is very useful when it comes to comparing the efficiency of the backend services and the web applications.

3.3. Toolchain and Environment

To have standard results while comparing the speeds of the three languages, the code for each language was written in standard version of Java, Python and Kotlin and all the codings were executed on pre-configured systems. These versions are stable as they belong to mature releases and popular among many developers. [16-18] Moreover, the benchmarking tests were carried out on an Intel i7 processor and 16GB of RAM which can be considered as a sufficient computational capacity to evaluate the efficiency without the influence of the hardware restrictions.

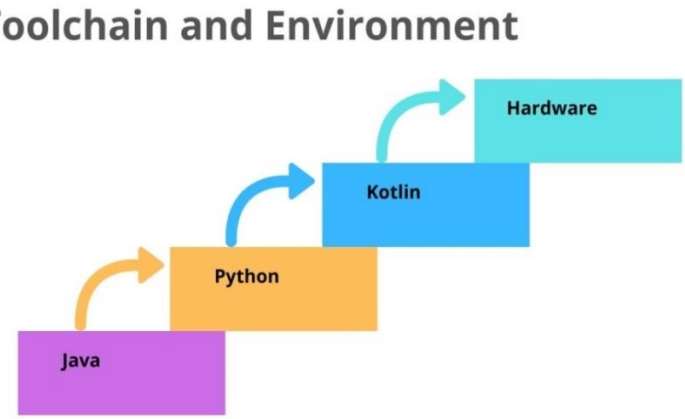

Fig 4: Toolchain and Environment

- **Java:** For the benchmarking test, JDK 17 was used because it is a current LTS version that provides enhancements in terms of stability and functionality. JDK 17 focuses on improvements of JIT compiler and garbage collector, which helps improve its operation speed and memory utilization. Java specifically optimized for multi-thread and have a faster running environment and because it runs on java virtual machine so it has fast compiling the code execution.
- **Python:** Python 3.9 was selected since it has some advantages over the previous versions, for instance, new inclusions as well as enhancements in dictionary functionalities and built-in functions. Yet as an interpreted language contrary to compiled languages such as Java and Kotlin, Python works slower but because of its simplicity and availability of a great number of libraries, it becomes extremely helpful for developing and data handling. The benchmarking tests were expected to compare Python's runtime performance disregarding the overhead coming from dynamic typing and interpreted interpretation.
- **Kotlin:** Kotlin 1.5 was chosen due to its given features and numerous optimizations for JVM execution. As Java runs on JVM, Kotlin gets the benefits of the JVM and at the same time, the syntax is much shorter. Contained in version 1.5, features such as inline classes, sealed interfaces, and enhanced type of inference, makes it fitting to compare C# with Java. Kotlin is fully interoperable with Java and hence Kotlin will perform as Java does but it is more modern and easy for developers to use because of its simple syntax.
- **Hardware:** The benchmark results of all the tests were obtained on a system with Core i7 processor and 16GB of RAM. These configurations prevent the variation across test cases due to system specifications, thereby making it easy for the program to run smoothly. The Intel i7 processor enabled the provision of multiple cores for parallel running of tasks, which is an advantage for the Java and Kotlin as both benefit from multi-threading. There was sufficient amount of RAM

available in the system that is 16GB and it was not a constrain in terms of memory during the evaluation of large datasets, file handling, and executing multiple processes simultaneously.

4. Results and Discussion

4.1. Performance Comparison

The testing that was done involved sorting algorithms such as Bubble Sort and Quick Sort, file processing and the response time of the APIs. To the optimization, Java fairly defeated Python performance since it is compiled and language as well as implementing Just-In-Time optimization. In its performance at the Java Virtual Machine (JVM), Kotlin shows good results close to those of the Java.

Table 1: Execution Time (in milliseconds) for Different Benchmarks

Language	Bubble Sort	Quick Sort	File Read	API Response Time
Java	120	15	300	80
Python	500	60	500	120
Kotlin	130	18	310	85

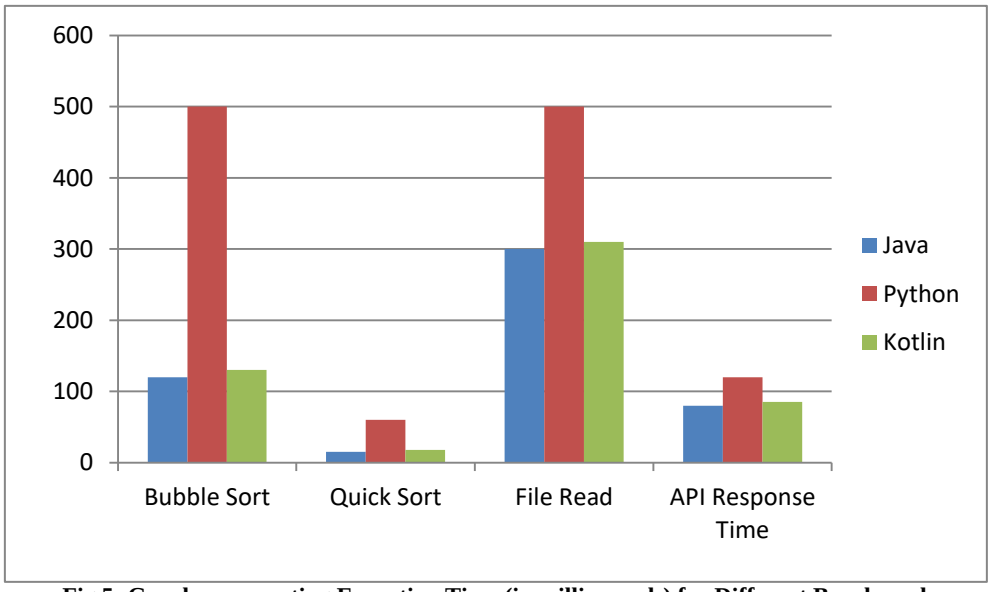

Fig 5: Graph representing Execution Time (in milliseconds) for Different Benchmarks

- **Bubble Sort:** Bubble sort is a simple and unsophisticated sorting algorithm, whose time complexity is O(n^2). $O(2)$, it is a good test to tell raw computational efficiency. Java performs the Bubble Sort in 120ms which is much faster than that of Python since it ran for 500ms partly because Python is an interpreted language and supports dynamic typing. Kotlin running on JVM and it also takes nearly the amount of time which java takes to complete the task i.e., 130ms. From this test, we also see that Python is less suitable to perform CPU bound operations.
- **Quick Sort:** Quick Sort is another quite efficient sorting algorithm whose average time complexity is $(\log n)$ .Speaking of the question of the efficiency of the language, the execution times show how optimally each language is capable of sorting the array of integers. Java takes 15ms to complete the Quick Sort while Kotlin takes 18ms, which is quite competitive since Kotlin has advantages of JVM. Python for example takes 60ms to execute the script since it is an interpreted language and also suffers much from dynamic typing.
- **File Read:** File I/O operations assess the ability of a language in managing large files. Java does its I/O operation far better as it takes 300ms to read a 500 MB file, which is almost the same as Kotlin's 310ms given that Kotlin compiles to the JVM. Python consume 500ms of time taken to handle the file since is an interpreted language and the I/O operations is not optimized for hi speed as shown. This means it can be inferred that Java and Kotlin handle large file processing tasks more than any other type of task.
- **API Response Time:** The API response time defines how quick the language returns the data when the web request is hit. Java, with its robust multithreading capabilities and efficient networking libraries, processes API calls in 80ms. Located closely, down at 85ms, is Kotlin running on top of the same Java networking frameworks. Even though Python is used often in web development, it takes 120ms – slightly slower if you want to use it for applications that require a lot of APIs. The reason for this is due to the Global Interpreter Lock (GIL) that Python has in place preventing parallel execution.

4.2. Syntax Readability

Table 2: Code Complexity Comparison

Task	Java	Python	Kotlin
Read a file and print lines	10	3	5
Define a simple class	12	4	6
API request and JSON parsing	20	7	8

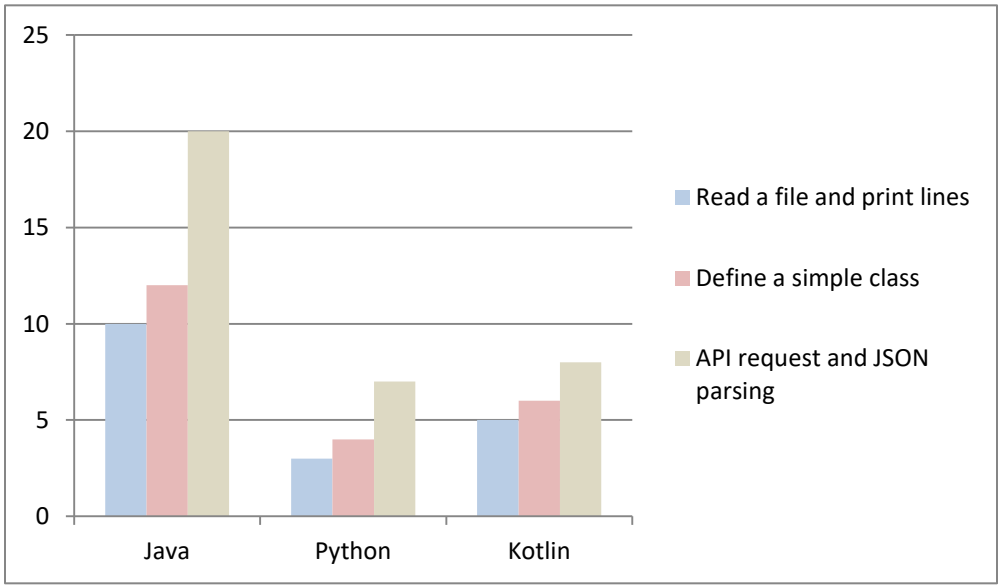

Fig 6: Graph representing Code Complexity Comparison

- **Reading a File and Printing Lines:** File handling is a very important operation in programming and the implementation of something that simple can affect quite substantially on your productivity. It's verbose syntax and the need of dealing with exceptions making Java's declaration to read a file (and print its contents) need 10 lines of code. The same work is done in 3 lines in Python by using built in file handling functions and simple syntax. Compared to Java's verbosity, Kotlin achieves the same thing by reducing boilerplate code and finishing up in just 5 lines, which is concise and readable.
- **Defining a Simple Class:** Object orientation makes common operations such as defining a class. Due to the strict syntax and the requirement of explicit type declaration, there are 12 lines of code needed for defining basic class with constuctor and attributes. Since Python is dynamically typed, it does 4 lines to achieve the same functionality filtering, not needing to define types, nor getters or setters. With neat syntax, Kotlin provides primary constructors on the built-in functionality of the main constructor, making the code balance the readability with the type safety in 6 lines of code.
- **API Request and JSON Parsing:** Nowadays, in the world of web and mobile development, you have to make an API request and parse the JSON response. This task is handled in 20 lines of code using Java, which requires explicit handling of HTTP requets and parsing of JSON data. With Python libraries (requests for http requests, and json for serialization) it is decreasing the complexity to just 7 lines. The entire task is accomplished in 8 lines in Kotlin, using modern HTTP clients and JSON handling, and results in more maintainable code than Java's verbose implementation.

4.3. Usability in Real-World Applications

Java, Python and Kotlin are each programming language and they have advantages and disadvantages which make them fit for certain real world implementations. Being robust, scalable and having a strong memory management Java is being used in many enterprise applications and backend development. Java is usually chosen for constructing high performance, distributed system, banking applications and web services for large organizations. It's a platform independent (thanks to the Java Virtual Machine (JVM)) software which means it can be executed on any environment without problems. Whereas Python stands out in terms of data science, artificial intelligence (AI) and scripting because of how simple and its wide library support is. Powerful frameworks on the machine learning, big data analytics and automation side are TensorFlow, Pandas, NumPy to name a few but Python has become a dominant language.

It is simple to use and has quick evolving capabilities, thus leading to its use in scientific computing and always automation applications. Kotlin was designed to be the new improved alternative to java in Android development and cross platform applications. Because of its concise syntax, it reduces boilerplate code but still remaining entirely compatible with Java, it

is widely used as the preferred language to build modern Android apps. Kotlin is also growing to be used in cross platform development with frameworks like Jetpack Compose and Kotlin Multiplatform where developers can write code to build applications across several operating systems.

5. Conclusion

A comparison of Java, Python and Kotlin was made according to three main parameters: performance, syntax readability and usability in real applications. Each of these languages has trade offs and advantages, therefore making them suited in different use cases. Java is known for speed of execution, security, and stability, and it's no doubt the favorite candidate for enterprise applications, large scale backend systems, and high performance computing projects. Thanks to its static typing, JIT compilation and garbage collection, it can evoke the optimal speed of execution and memory management. Moreover, Java's widespread used in these industries such as finance, e-commerce and cloud based services of Java is reliable for building scalable and secure applications. Its verbosity and a longer development time sometimes can be a drawback where an app needs to be prototyped quickly.

Unlike python, it is well known for its simplicity and simplicity of use, being one of the most popular languages among the beginners and professionals alike. This is because its dynamic typing and a minimalistic syntax facilitate faster development and debugging. It has a rich ecosystem of libraries that are widely used in artificial intelligence (AI), machine learning, data science and automation like TensorFlow, Pandas, and NumPy. At the same time, web development using Django and Flask is handled quite well. Python, however, has a limitation that its execution speed is a lot slower than Java and Kotlin. Because they can be interpreted, Python usually falls short in terms of a performance-intensive applications, it is not the best fit for huge systems running on optimized execution. And yet, Python continues to be the preeminent language for the rapid prototyping, scripting, or research-oriented development.

Kotlin is an agreement between Java's robustness and Python's simplicity: a modern bridge between the two. Due to reduced boilerplate code, null safety and logical readability, Kotlin was originally developed as an alternative to Java, but has been extended to the point of meeting the expectations in Android development. Because Kotlin is interoperable with Java, developers can easily move Java applications to Kotlin without having to do significant code refactoring. Apart from that, Kotlin also makes its presence felt in the cross platform development field, leveraging Kotlin Multiplatform that allows developers to write in code for multiple platforms in very little effort. While Kotlin runs on the JVM it has much closer performance to Java over its Kotlin and Python counterparts, but lacks the extensive ecosystem available for data science and AI applications. At the end of the day, it is about the requirements of the project and programming language is the next most important factor. High performance enterprise application running in Java, prototype out Python, passionate AI and rapid advancement application based on Python, modern Android development and cross platform application written in Kotlin. By understanding these strengths, developers can make the appropriate language selection based on achieving efficiency and a successful project with the help of the right language.

References

[1] Espitia Acero, J. S. (2020). Empirical testing for establishing benchmarks: process review and comparison between java, kotlin and dart's performance.
[2] Gakis, S., & Everlönn, N. (2020). Java and Kotlin, a performance comparison.
[3] Luca Crisan, A. (2019). A study of Kotlin's: conciseness, safety and interoperability (Bachelor's thesis, Universitat Politècnica de Catalunya).
[4] Java vs Kotlin vs Python: Who Wins the Programming Language War?, Medium, online. https://pranav-mohan365.medium.com/java-vs-kotlin-vs-python-who-wins-the-programming-language-war-d8f0a5a57012
[5] Jemerov, D., & Isakova, S. (2017). Kotlin in action. Simon and Schuster.
[6] Avrutin, S. (2006). The usability of syntax. Applied Psycholinguistics, 27(1), 43-46.
[7] Green, T. R. G., & Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' framework. Journal of Visual Languages & Computing, 7(2), 131-174.
[8] Hellbrück, S. (2019). A Data Mining Approach to Compare Java with Kotlin.
[9] Ardito, L., Coppola, R., Malnati, G., & Torchiano, M. (2020). Effectiveness of Kotlin vs. Java in android app development tasks. Information and Software Technology, 127, 106374.
[10] Java vs Kotlin: A Comprehensive Comparison for Developers, codefinity, online. https://codefinity.com/blog/Java-vs-Kotlin%3A-A-Comprehensive-Comparison-for-Developers
[11] establishing benchmarks: process review and comparison between java, kotlin and dart's performance.
[12] Åkesson, T., & Horntvedt, R. (2019). Java, Python and Javascript, a comparison.
[13] Hunt, J. (2021). Beginner's Guide to Kotlin Programming (pp. 1-519). Springer.
[14] Pierce, B. C. (2002). Types and programming languages. MIT press.

[15] Huang, X. (2018). Comparative Evaluation of Java and Kotlin for Android App Development (Doctoral dissertation, University of Applied Sciences Technikum Wien).

[16] Java vs Python vs Kotlin: Battle of the Best | Which One You Should Use, Aegis, online, https://www.aegissofttech.com/articles/java-vs-python-vs-kotlin-battle-of-the-best.html

[17] Li, Y., Katsipoulakis, N. R., Chandramouli, B., Goldstein, J., & Kossmann, D. (2017). Mison: a fast JSON parser for data analytics. Proceedings of the VLDB Endowment, 10(10), 1118-1129.

[18] Marcotty, M., & Ledgard, H. (2012). The world of programming languages. Springer Science & Business Media.

[19] Wu, W. L., Budianto, I. H., Wong, C. F., & Gan, S. K. E. (2019). A Review of Apps for Programming: programming languages and making apps with apps. Scientific Phone Apps and Mobile Devices.

[20] Garcia, R., Jarvi, J., Lumsdaine, A., Siek, J. G., & Willcock, J. (2003, October). A comparative study of language support for generic programming. In Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications (pp. 115-134).