УДК 004.432

В.В. Наконечний

Д.І. Катєльніков

The benefits of using Java as a universal programming language. Why Python has become more popular in the recent years

Вінницький національний технічний університет

Анотація

У цій доповіді детально розглядаються переваги використання мови Java як універсальної мови програмування, розглядаються переваги, такі як: кросплатформеність, багата екосистема, об'єктно-орієнтованість та безпека. Розглядаються перспективи подальшого використання мови Java. Причини, чому Python обирають як основну мову програмування ШІ попри переваги мови Java розглядаються в цій роботі.

Ключові слова: мови програмування, штучний інтелект, Java, Python.

Abstract

This report delves into the benefits of using Java as a universal programming language, discussing its strengths such as platform independence, rich ecosystem, object-oriented nature and security. The further perspectives of using Java as a universal language are examined. Reasons why Python is chosen over Java for developing AI are discussed.

Keywords: programming languages, artificial intelligence, Java, Python.

Introduction

Nowadays more and more tasks are demanded from various programming languages. The narrow-specialized languages remain the leaders in their fields. Such languages would be: Mathcad, COBOL, Matlab etc. But since such aspects as cross-platform ability, security and high level of abstraction became essential, Java and Python became the undisputed leaders of the market. Java, developed by Sun Microsystems in 1995, has been a widely popular programming language for more than two decades. Its widespread adoption can be attributed to its unique features that make it a versatile and powerful language for various applications. The **aim** of this report is to explore the benefits of using Java as a universal programming language, focusing on its simplicity, portability, scalability, security, and vibrant community, and the reasons why Python tends to be used more and more in the recent years.

Results of the research

Portability

Java is a cross-platform language that allows using the same code on all operative systems. Java's "Write Once, Run Anywhere" (WORA) philosophy ensures that Java code can run on any platform with a Java Virtual Machine (JVM) without the need for recompilation. This was made possible by using bytecode as a medium between Java code and machine code [1]. Javac or Java compiler compiles the source code and the product of such compilation is bytecode (highly optimized none-executable code). The bytecode is then interpreted by the JVM or Java virtual machine which is unique for every platform. JVM translates the bytecode into machine code. Just like Java, Python uses a virtual machine to translate bytecode into machine code. The difference is that Java compiles the code in advance to transform it into bytecode while Python uses a runtime compiler which checks the syntax during compilation and often leads to less stability and errors. Java code must be well-written before compilation and javac will not compile the source code if it contains different sorts of mistakes [2]. Python, on the other hand, is an interpreted language which allows compiling a poorly-written code which can cause trouble when transferring from one platform to another.

Syntax: When comparing the simplicity of learning Java and Python, it is important to consider various factors such as syntax, code readability, learning resources, and the ease of setting up the development environment. Although both languages have their merits, Python is often considered the easier language to learn, especially for beginners. Python syntax is designed to be clean and to resemble English language. Java has a more verbose syntax and requires the detail knowledge of object-oriented principles. Python is easy to learn for beginners in programming while Java is more difficult, but experienced developers tend to choose the more structured syntax, which Java offers. Java code is easy to read, but requires intermediate knowledge of Java core. The minimalistic style and simplicity of Python makes it a popular choice for teaching programming in schools and universities, which further contributes to its accessibility for new learners.

Learning Resources: Both Java and Python have extensive learning resources available online, including tutorials, documentation, and forums. However, due to its beginner-friendly nature, Python often has more resources specifically targeted towards beginners.

Development Environment Setup: Setting up a Python development environment was generally simpler and faster than setting up a Java development environment before IntelliJ IDEA announced their universal IDE. Setting up IntelliJ is generally simple. In addition, IntelliJ provides beginners with tutorials and uses IntelliSense, which predicts the further code and generates basic constructions. Just like Java, Python can be installed and configured with minimal effort, and integrated development environments (IDEs) such as PyCharm and Visual Studio Code offer seamless support for Python development.

Security: Java provides robust security features, making it a reliable choice for applications that require stringent security measures. The JVM offers a sandbox environment, isolating the application from the underlying system and protecting it from malicious code. Additionally, Java offers built-in security features, such as secure class loading, bytecode verification, and access control mechanisms, to ensure the safe execution of code. Java provides the developer with a large set of APIs, implementations and tools for security. This is a list of modules that contain security APIs [1]:

- java.base;
- java.security.jgss;
- java.security.sasl;
- java.smartcardio;
- java.xml.crypto;
- java.security.auth;
- jdk.security.jgss.

Java's security features have been battle-tested in enterprise environments, where security is critical. The language is commonly used in industries such as finance, healthcare, and telecommunications, where stringent security measures are required. Java uses bytecode verification. Bytecode verifier checks the bytecode for malicious code and makes sure only verified bytecodes are executed. Another security feature of Java is the garbage collector which prevents memory leaks. Python offers security features as well. There are numerous libraries for security management but there are existing gaps in security. Java is considered to be safer. Python's security relies more on the developer's understanding of secure coding practices and the proper use of security libraries and tools. While it is possible to write secure applications in Python, it may require more effort and attention to detail compared to Java.

Artificial Intelligence: Java could possibly become the best language for AI [3] but most companies have chosen Python over Java due to its simplicity and readability. Java is faster due to the excellent performance of the JVM but the AI developers have chosen the minimalistic syntax of Python. Python has become the de facto programming language for AI and machine learning, primarily due to its simplicity, readability, and extensive libraries and frameworks. Libraries such as TensorFlow, PyTorch, Keras, and scikit-learn have made it easier for developers and researchers to build, train, and deploy AI models [4]. Furthermore, Python's support for scientific computing through libraries like NumPy, SciPy, and Pandas has further solidified its position as the go-to language for AI. Java has libraries for AI development as well. Here is a list of Java`s largest AI libraries [5]:

- Apache Jena. Apache Jena is an open-source Java framework for building Semantic Web and Linked Data applications. It provides an extensive set of tools and libraries to work with RDF (Resource Description Framework) data and perform various operations, such as parsing, querying, and reasoning over RDF graphs;
- PowerLoom Knowledge Representation and Reasoning System. PowerLoom provides a rich and expressive language for representing and manipulating knowledge, and it supports various forms of reasoning, such as deductive, inductive, and abductive reasoning;
- Tweety. A collection of Java frameworks for developing AI logics;
- Neuroph. Java Neural network creation framework;

- Deeplearning4j. A deep learning library for the JVM.

These Java AI libraries and frameworks offer diverse functionality, making it possible for developers to build AI applications using Java. However, it is important to note that the Java AI ecosystem is not as extensive or widely adopted as Python's AI ecosystem. Unfortunately, it is unlikely that Java will be considered the best language for AI in the nearest future, given Python's current dominance in the field and the support it receives from the AI research community. While Java is a powerful language and has its strengths, Python is expected to remain the preferred language for AI development due to its extensive libraries, simplicity, and widespread adoption.

Enterprise Support and Adoption: Java is widely used in enterprise environments, where stability and reliability are paramount. The language has a proven track record of successfully supporting large-scale applications in industries such as finance, healthcare, and telecommunications. Additionally, its popularity and widespread adoption have led to strong support from major technology companies, ensuring its longevity and continued development.

Conclusion

Java's simplicity, portability, scalability, security, and vibrant community make it an ideal choice for a universal programming language. Its widespread adoption and support from major technology companies further establish its position as a reliable, versatile, and powerful language for developers to create diverse applications. By using Java as a universal programming language, developers can ensure they are working with a language that is easy to learn, adaptable to various platforms, and capable of supporting even the most demanding applications. Python is an object-oriented language as well. It is gaining popularity due to its simplicity and readability. Numerous libraries provide Python developers with powerful tools to work with. Python's enterprise support and adoption have grown significantly due to its simplicity, versatility, extensive libraries and frameworks, and strong community backing. Python has become a popular choice for many enterprises, and its continued growth and improvement make it a reliable option for businesses of all sizes.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The language environment. [Електронний ресурс]. — Режим доступу: https://www.oracle.com/java/technologies/architecture-neutral-portable-robust.html (дата звернення: 31.03.2023). — Назва з екрана.
2. Python vs Java: key differences and code examples. [Електронний ресурс]. — Режим доступу: https://www.imaginarycloud.com/blog/python-vs-java/ (дата звернення: 31.03.2023). — Назва з екрана.
3. Java: Is It the Best Language for Artificial Intelligence? [Електронний ресурс]. — Режим доступу: https://www.sam-solutions.com/blog/java-is-it-the-best-language-for-artificial-intelligence/ (дата звернення: 31.03.2023). — Назва з екрана.
4. Python libraries for AI. [Електронний ресурс]. — Режим доступу: https://www.unite.ai/10-best-python-libraries-for-machine-learning-ai/ (дата звернення: 31.03.2023). — Назва з екрана.
5. Java AI. [Електронний ресурс]. — Режим доступу: https://www.baeldung.com/java-ai (дата звернення: 31.03.2023). — Назва з екрана.

Наконечний Влас Володимирович – студент групи 3ПІ-22б, Факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: vlas.nak.05@gmail.com

Катєльніков Денис Іванович – к.т.н., доцент кафедри програмного забезпечення, Факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, e-mail: fuzzy2dik@gmail.com.

Vlas Nakonechnyi Volodymyrovych – student of 3PI-22b group, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: vlas.nak.05@gmail.com

Katielnikov Denys Ivanovych, PhD, Associate Professor of Software Engineering Department, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: fuzzy2dik@gmail.com.