UDC 004.9 **doi: 10.26906/SUNZ.2023.3.093**

T. Dmytrenko, T. Derkach, A. Dmytrenko

National University «Yuri Kondratyuk Poltava Polytechnic», Poltava, Ukraine

USING JAVA AND C # PROGRAMMING LANGUAGES FOR SERVER PLATFORMS AND WORKSTATIONS

Abstract. The paper analyzes the features of the use of known programming languages Java and C #, and their use in software development for server platforms and workstations. Their main advantages and disadvantages are investigated, some differences in their syntax are compared. Based on the existing dynamics of popularity, the prospects of Java and C # programming languages are shown. By comparing a simple synthetic test, their speeds were compared on the Java virtual machine version 1.8 and the Microsoft .NET Framework version 4.5. As a result, recommendations for further use of Java and C # programming languages are offered.

Keywords: Java, C #, JVM, .NET, programming.

Introduction

Nowadays, two popular and rapidly developing competing programming languages are Java and C#. Even though C# appeared much later than Java, they have a lot in common, but as they say, nothing is perfect.

This case is not an exception, and in this article, an analysis of the advantages and disadvantages of each of them is carried out. When comparing C# and Java, we must start with their origin stories. What is the reason for this? Well, mainly because of their similarity, which explains the many similarities you face today between these languages. C # and Java languages appeared at different times. The Java language was created long before C #. Under the name Oak Java was developed by Sun Microsystems in 1990, and in 1995 the first beta version of Java was released. The creation of C # was announced in 2000, and in 2002 the first version of the .NET platform supporting C # was released. Thus, if Java was created based more on the experience of Objective C and C, then for C # such support was C ++ and Java itself. And, despite its name, C # turned out to be closer to Java than to C ++. From the developer's point of view, Java and C # are very similar. Both languages are strictly typed and object-oriented. Both incorporate much of the C++ syntax, but unlike C++, are easier to learn for beginners. Both borrowed from C a set of basic keywords and service symbols, including curly braces for delimiting blocks. Both languages come with rich library collections. But languages also have their features and differences, strengths, and weaknesses. C # took into account many shortcomings of Java and corrected them in its implementation. But Java does not stand still, evolving in parallel with C #. Kick Redek of Microsoft considers C # to be a more complex language than Java. According to them, "*Java was built to keep a developer from shooting himself in the foot*", and "*C # was built to give the developer a gun but leave the safety turned on*" [1].

The purpose of the article is analysis of Java and C # programming languages' functionality. Considering the advantages and disadvantages of each language, determine the best option for software development.

Presenting main material

C# is a product of Microsoft and the main direction of this language is products for the operating system of the Windows family, both for desktop computers and for mobile devices. The Unity 3d game engine, on which many games are developed, is also gaining popularity, primarily due to its cross-platform nature (the ability to run on different operating systems) [2]. Using the Xamarin platform allows you to use C # to develop applications for various operating systems, including Android. Although the system is quite new and at first glance very promising, it is not known exactly how long it will take root. This is due to the fact that Microsoft sometimes closes its products if it sees their futility. This can be easily verified by installing their main tool for developing Microsoft visual studio software. If you compare their versions, you can see that older versions contain such tools for program development, which are excluded in more modern versions of Microsoft visual studio, such as J # [3]. In this approach, Microsoft has both advantages and disadvantages. The advantages are that to maintain the relevance of products, it is necessary to spend company resources that can be used to develop more promising products. The main disadvantage is that the programmer will have to learn another language and start all over again, which is not good for him, as well as for the company's reputation in his eyes. The strictness of the language helps to avoid many obvious mistakes. Java, according to statistics, ranks first in popularity. The scope of this language is very large and it supports the ability to run programs almost anywhere: almost all operating systems, personal computers, servers, mobile phones (using Java mini), Android operating system, GPS navigators, VCRs, satellite systems, pressure meters, heart rate monitors, even bracelets to measure the quality of human sleep, etc. According to official developer data, Java is used on more than 3,000,000,000 (three billion) devices [4]. The approach of Java developers in the field of stability is completely opposite to Microsoft. New Java components are tested long before they are fully marketed and are not removed after they are released. Undoubtedly, this method has a significant disadvantage, the novelty of the product is introduced very slowly. Despite this, nowadays, the latest version of the Java language is almost not inferior in innovations to C# [5].

Regarding the comparison of the performance of programs written in Java and C#, we can say that this comparison is a synthetic test, and its purpose is to compare how long it will take programs written in C# (net 4.5) version "Release" and Java (1.8) to complete tasks in the Windows 10 operating system. For example, consider the process of creating the same class a given number of times, for example, 1,000,000,000 times, in which the same mathematical functions are performed. Programs were run not from editors, but from created files, which were created not as Debug versions, but as Release.

As a result of the test, you can clearly see that to perform the same operations, the program in C # took more time than the program in Java, ie a program in Java performs the same operation several times faster than a program in C # [3].

There is a set of libraries for both languages, which allows you to build a user interface for desktop applications. In the case of Java, these are the Swing and SWT multiplatform libraries, as well as the platform JavaFX, which allows you to create RIA applications.

For C# on the Windows platform, the main platforms for developing desktop graphical applications are Windows Forms and WPF. There is a special WinRT platform for development under Windows 10. For other platforms, the gtk # library, created as part of the Mono project, is used. Developments that are originally based on Windows are usually built on Windows Forms, their transfer to another platform becomes time-consuming [4]. C# development in the Mono environment using gtk# is ported, but much less. There is no implementation of the WPF platform within the Mono project, so WPF programs are not portable to Linux–based operating systems.

C #, along with Java, is gradually becoming popular on several Linux and BSD-based operating systems. The implementation of the Mono project was a legally painless process, as the CLR and C # languages are standardized by Ecma and ISO, and anyone can implement them without worrying about the legal side of the case. At the same time, it should be noted that a program written in a Windows environment can have significant problems running under another OS [1]. For server software development, these two programming languages are closest to being considered competitors. Java with its J2EE (Java (2) Enterprise Edition) platform and C# with its ASP.NET are at odds when it comes to creating dynamic web content and applications. Both programming languages are widely used and supported in this market, along with a set of tools and related products created in JavaEE and .NET [2].

Since the creation of C #, it is constantly compared with Java. There is no denying that C # and its CLR control shell owe much to Java and its JRE (Java Runtime Environment). It is arguable that C # development is the result of Microsoft's recognition that the Java-led control code shell has many benefits in a growing networked world, especially with the advent of the Internet on non-personal devices and the growing importance of network security. [3]. Prior to the creation of C #, Microsoft modified Java (creating J ++) to add features that only work on Windows, thus violating the Sun Microsystems license agreement. The development of these two programming languages, as well as their APIs, binaries, and execution environments, is governed differently. C # is defined by ECMA and ISO standards, which specify the syntax of the language, the format of the executable modules (known as CLI), and the Base Class Library. The standards do not include many new libraries, such as libraries for databases, GUIs, and web applications. To date, no component of the Java environment is standardized by Ecma, ISO, ANSI, or any other standards organization. While Oracle retains unlimited exclusive legal rights to modify and license its Java trademarks [4].

Conclusions

Today, Java is more popular than C#, and considering the popularity graphs, the situation will not change in the near future. C # attracts primarily because you can write anything on it, thanks to the .NET platform and a huge number of frameworks. However, despite its affinity with C ++, it is quite friendly to beginners. The syntax of C# is more concise because signs are used instead of long words. Its special advantage is the ability to divide classes into separate parts and files with the instruction "partial". In terms of software development speed, C# is once again the leader – it is possible to write a simple program using GUI (graphical user interface) faster than Java [5]. If the program must be cross-platform, there are no options when choosing between C# and Java – Java will be better in this case. In the speed of the conducted synthetic test, Java won by a significant margin. Thus, we can conclude that if you need stability, cross-platform, and speed of the program itself, it is better to write in Java, and if you need to write a program that will run on Windows then it will be more profitable to choose C #. In general, do not worry too much about this choice. C # was originally created as an analog of Java, so the languages are quite similar. After all, a programming language is just a tool, and C # and Java are like two tools that are the same in purpose but from different companies. Therefore, the main thing is to learn how to use this tool, and which one you have specifically is not so important.

References

1. Jeffrey Richter CLR via C #. Programming on the Microsoft.NET Framework 4.5 in C #: Print2print. – 2020. – 896 p.
2. Andrew Troelsen, Philip Jepix C # 6.0 programming language and .NET 4.6 platform: Apress. – 2016. – 1440 p.
3. Joseph Albahari, C# 10 in a Nutshell. The Definitive Reference: O'Reilly. – 2022. – 1058 p.
4. Herbert Shieldt Java 9. Complete Guide: Науковий світ. – 2023. – 752 p.
5. Robert Laforet Data Structures and Algorithms in Java: Print2print – 2016. –704 p.

Received (Надійшла) 18.05.2023
Accepted for publication (Прийнята до друку) 13.09.2023

Використання мов програмування Java та C# для серверних платформ та робочих станцій

Т. А. Дмитренко, Т. М. Деркач, А.О. Дмитренко

Анотація. В роботі проведено аналіз особливостей використання відомих мов програмування Java та C#, та використання їх у розробці програмного забезпечення для серверних платформ та робочих станцій. Досліджено їх основні переваги та недоліки, проведено порівняння деяких відмінностей у їх синтаксисі. На основі існуючої динаміки популярності, показана перспективність мов програмування Java та C#. Шляхом простого синтетичного тесту було проведено порівняння їх швидкостей на платформі Java virtual machine версії 1.8 та Microsoft .NET Framework версії 4.5. В результаті запропоновано рекомендації, щодо подальшого використання мов програмування Java та C#.

Ключові слова: Java, C#, JVM, .NET, програмування.