

**Dušan Tošić**

Faculty of Mathematics, University of Belgrade

## ROLE OF PROGRAMMING LANGUAGES IN DIGITALIZATION

**Abstract.** The first high-level programming language was FORTRAN and was created in 1957. Since then, a large number of programming languages have been created. The time of creation of some of them, as well as their mutual relationship, can be described by a genealogical tree. On the genealogical tree that we used, 4 periods in the development of programming languages were distinguished, and in each period several of the most important languages were chosen. 14 most significant programming languages were selected (as chosen by the author of this work), and the main criterion was: popularity and influence on development of other languages. For each of the selected languages, some characteristics are briefly mentioned, as well as their importance for digitalization.

**Keywords.** programming languages, digitalization, genealogical tree.

### Introduction

Today, digitalization is present in all human activities. What do we mean by digitalization? According to wikipedia[1] "digitalization may refer to: digital transformation, the adoption of digital tools to create new or modify existing products, services and operations". Digital transformation is based on digitization - the conversion of analog information into a digital format. There is a difference between digitalization and digitization. By digitization, content from any medium is transferred into a digital form, usually expressed by binary digits 0 and 1 [2]. Conversion does not change the content downloaded from the medium, it is just written differently. Digitalization means to change business processes over to use digital technologies. For digitalization, it is important to operate with data in digital form, which is done through digital transformation. Digital transformation is realized by electronic (digital) computers, often using a programming language. So far, a large number of programming languages (several thousand) have been created, many of which are dead (no longer used). However, a large number of programming languages have been successfully used in the digitalization process, and a significant number of them are still in use today. Following the development of programming languages from their origin until today, we will consider the impact of some of the most important (in the author's opinion) on the digitalization process.

### Beginning of digitalization

It is difficult to say exactly when digitalization started. First steps towards digitalization were made in the 1940s, when digital computers supplanted analog ones. In that period, one cannot talk about digitalization as it is understood today, that is [3] "the ongoing integration of digital technologies and digitized data across the economy and society". Computer equipment is essential for digitalization. The first computers were very expensive and could only be owned by large companies and institutions. In this period, there were no conditions

for digitalization even in developed countries. With the advent of mini computers, the situation improved, computers became cheaper and could be purchased by smaller companies and organizations. The use of mainframes (IBM/360, S370, Unisys, Cyber, ...) and minicomputers created better conditions for digitalization. Now we can already talk about the beginning of digitalization. However, digitalization flourished with the appearance of microcomputers, especially PCs. This is the period when every individual can get a computer and use information from the digitalization process.

### **Digitalization and languages**

Language is a set of rules for communication between two or more subjects. Language is used to present and transmit information. All languages can be classified into two categories: natural and artificial. Natural languages are used for communication between people in spoken or written form. Artificial languages are created for use in certain areas where the use of natural languages is not suitable. Artificial languages include programming languages. Programming languages serve primarily for communication between humans and computers (sometimes it is also used for communication between people!) There is no strict (mathematical) definition of a programming language. However, there are various descriptive definitions. Thus, we can define a programming language as a set of syntactic and semantic rules used for the description (definition) of computer programs. What impact does digitalization have on languages, in general, and especially on programming languages? There is a large number of works on the relationship between language and digitalization. We will mention only two here. In the paper [4], the effect of digitalization on language and culture is considered. In the paper [5] the impact of digitalization on natural languages is considered. It is concluded that the impact is significant, but also different for different age categories. This could be expected because of the emergence of new technology regularly significantly affects natural languages (for example, the appearance of the automobile has influenced natural languages to be enriched with a number of new terms.) However, programming languages are directly involved in the digitalization process, so the impact of digitalization on programming languages is much more pronounced.

**Programming languages.** In order to make it easier to understand the terminology used here, we will list some commonly known terms related to programming languages. For a more detailed introduction to these concepts see [6] and [15]. A programming language can also be defined as an artificial language for describing constructions that can be translated into machine language and executed by a computer. Electronic computers can only execute constructions expressed using the binary alphabet, i.e. 0 and 1. A language created over a binary alphabet is called a machine language. Different types of computers use different machine languages. Machine languages are not suitable for humans, so languages similar to natural languages were created for writing programs. These are programming languages and all of them can be classified into 2 categories: machine-dependent and machine-independent languages. Machine-independent languages are also called high level programming languages. The first digital computers used exclusively machine-dependent languages. After the appearance of the first high level programming language, machine-dependent languages are used less and less, and a whole series of new high level languages have been created. In the further text, we will consider only high-level programming

languages (some authors make a distinction between classical programming languages (FORTRAN, Pascal, Java, ...) and script languages. However, we will treat these languages in the same way).

Programming languages differ based on their characteristics (simplicity, expressiveness, uniformity, portability, efficiency, ...), but all programming languages are unique (unambiguous), unlike natural ones. In order to make a programming language with the best features, a large number of programming languages have been created. New languages are made by using previously created ones and taking their good features. Genealogical trees (Fig. 1.) are used to display the time of origin, as well as the mutual connections between languages. Views of the connections between programming languages for different authors are different and therefore there are various genealogical-trees [7]. In order to highlight the most important programming languages, we will use the diagram from the site [8]. We distinguish 4 periods in the development of programming languages (see Fig. 1.) the period of

- first programming languages,
- programming languages of the Middle Ages,
- programming languages created at the end of the 20th century and
- future programming languages.

In each of the mentioned periods, several of the most important programming languages (according to the author's opinion) have been singled out and these languages are colored in different colors (yellow color is used for languages of the first generation, red for languages of the second period, etc.)

**Script and markup languages.** A scripting language is a programming language used for writing scripts. A script is a list of commands that can be executed without user intervention, that is, without interaction with the user. A script is a program written in a scripting language. Script languages are most often used to connect components within an application. They are often used in graphic environments. Some of the scripting languages are: JavaScript, PHP, Perl, Python, AutoLISP, VBScript.

An important role for the digitization process is played by markup languages, such as: SGML, HTML, XML. In parallel with the development of markup languages (especially XML), specialized programming languages have been developed for various processing related to markup languages. More about for markup languages, see [9]. We will not deal with markup languages here.

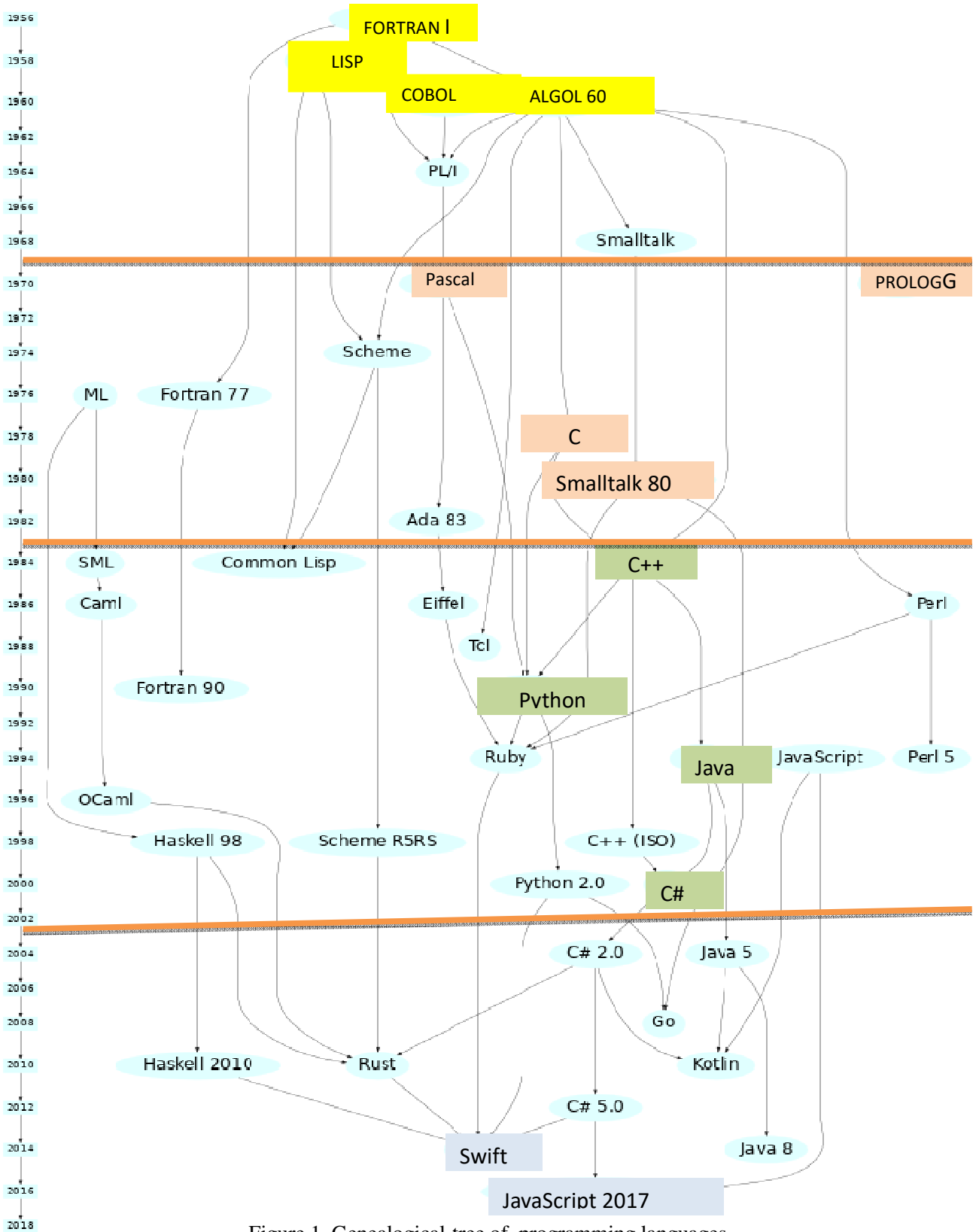


Figure 1. Genealogical-tree of programming languages

### **First programming languages and digitalization**

In the period when the first high-level programming languages were created, there were no conditions for digitalization (insufficient number of computers, absence of internet). Therefore, programming languages from this period could not support digitalization explicitly. However, through them, the ground was prepared for the future digitalization process, and therefore they deserve to consider their role. The first high-level programming language was FORTRAN [10]. The name FORTRAN is derived from FORMulaTRANslation. It is a language created by a team of developers led by John Backus at IBM. It is intended for scientific research based on numerical calculations. After the first version of FORTRAN, several new ones appeared soon. The existence of various versions influenced that the problem of transferring programs from one computer to another appears. Therefore, in 1962, the American Standards Association established a committee with the task of proposing a FORTRAN standard. Starting with FORTRAN 66 and ending with FORTRAN 2023, several versions were proposed and with each new proposal the language was enriched according to new scientific achievements in this area. This has made FORTRAN one of the most vital and widely used programming languages. It is estimated that in the 1960s year of the 20th century, about 70% of computers in the world had a FORTRAN compiler. Thanks to the support of powerful corporations, many software packages from various fields have been created. A large number of them are still in use today. There is no mention of digitalization support by FORTRAN. FORTRAN was primarily used for numerical processing, while other types of processing (important for digitalization) were less common. It is possible that newer versions of FORTRAN were used for some applications, when digitalization began, but new languages that better supported digitalization were already in use.

In this period, it was considered that there should be special languages for mathematical and technical applications (such as FORTRAN) and special ones for business applications such as COBOL. The name COBOL is an acronym for COmmon Business Oriented Language and was implemented under the leadership of Grace Murray Hopper. COBOL has long been the leading language for business processing. COBOL contains a number of bad constructs and therefore it is not comfortable to use. According to recommendations of well-known experts in computing, this language is no longer used (it is not taught at colleges, it is not supported by modern literature, ...), although some applications developed in COBOL are still in use. There is no targeted support for digitalization in COBOL, but it can be said that the first steps towards digitalization are done in COBOL. Applications for: credit card transaction, traffic signals, phone calls etc. developed using COBOL are significant for the digitalization process.

An important programming language from this period is LISP, which was created under the direction of John McCarthy [11]. The name LISP is derived from the LIStProcessing. As the name suggests, LISP is used for list processing. LISP promoted a functional style of programming - operations in it are based on the mathematical notion of function. Immediately after its creation, it was recognized as the language of Artificial Intelligence (AI for short). After initial success in solving some problems, the creators and users of LISP pretended to use LISP to solve all the most important problems of AI. Later theoretical results on NP-complete problems showed that this was not possible, which contributed to the decline in popularity of LISP. After that, LISP was operated only in some research centers, without wide acceptance. LISP became relevant again with the advent of

PROLOG. Namely, it turned out that everything that can be done in PROLOG can also be done in LISP. As LISP was an American product, many American programmers began to return to LISP. (Thus, LISP was used in the creation of well-known software packages, such as: Mathematica, AutoCad, etc.) LISP is the language that probably has the most descendants [12]. Some AI problems solved in LISP are essential for digitalization. Therefore, it can be said that LISP is more significant for digitization compared to other languages of this period. LISP (some versions of it) is still used today, often in combination with scripting languages.

Another important language from this period is ALGOL 60 (Naur et al. 1960). This language is produced by a group of experts, half nominated by ACM and half by various European institutions. ALGOL is an abbreviation of ALGORithmic Language, which means that it is a language for describing algorithms. Due to the large number of “bad” (arithmetical if statement, go to statement, ...) constructions in FORTRAN, ALGOL 60 was created with the idea of replacing FORTRAN. However, ALGOL 60 never reached the popularity of FORTRAN - barely 20% of computers in the world had a compiler for ALGOL 60. So, how did ALGOL 60 become among the most popular programming languages in this period? As ALGOL 60 was created by top experts, many solutions were proposed in it, which were later implemented in ALGOL-60's descendant languages, such as: Pascal, C, Smalltalk, ... In this way, ALGOL 60 left a deep trace of development almost of all modern programming languages. ALGOL 60 did not directly influence digitalization, but it did through its successors

### **Programming languages of the Middle Ages and digitalization**

At the beginning of the 60s years of the 20th century, significant experience was gained in creating programming languages. The idea of creating a universal language appeared. The International Federation for Information Processing (IFIP, established in 1960) had founded a Working Group on ALGOL (WG 2.1) with the task that, using ALGOL 60, form a proposal for a new language. The members of the group disagreed about the conception of the new language. Some thought that it should be a powerful language that would contain the good features of most of the existing programming languages. Based on this concept, ALGOL 68 was created. This language was cumbersome, difficult to learn and compile, and it is almost forgotten today. In addition to ALGOL 68, several programming languages were created with the same idea - to produce a universal programming language that would cover different fields. Such languages were PL/1 and PLANNER. They suffered a similar fate as ALGOL 68, although PL/1 was used longer thanks to the support of IBM.

The members of the second group (led by Niklaus Wirth) believed that the new language should be simple, easy to learn and translate. That is how the Pascal programming language was born [13], [14]. The features that adorn Pascal are: simplicity, expressiveness, efficiency, conciseness and comprehensibility. With Pascal, the structured style of programming was successfully promoted. There are a number of programming languages that are direct descendants of Pascal. One of the most famous is the Ada programming language, which enables parallel programming. Pascal was mostly used for educational purposes - for learning programming. Pascal did not directly significantly influence digitalization, but indirectly (through his descendants) it did.

Unlike Pascal, the popularity of the C programming language is not declining. Various studies show that C [16], [17], [18] is among the 5 most popular programming

languages even today. C was developed in 1972 by Dennis Ritchie at Bell Labs for implementing the UNIX operating system[23]. From its inception until today, it has been used to create: operating systems, embedded systems, web environments by using the Common Gateway Interface, etc. It is enough to mention only these areas and it is immediately clear that C had a significant role in digitalization. C is a universal programming language and due to its good features it will certainly be used in the future.

In this period, the programming language PROLOG was created [20], [21]. PROLOG is based on mathematical logic, and its creators are Colmerauer and Roussel. PROLOG is a non-procedural programming language and it is suitable for solving AI problems. The Japanese have announced the creation of a 5th Generation computers in which PROLOG would play the role of a machine language. Unfortunately, such computers were not built, and interest in PROLOG declined. PROLOG did not significantly affect the digitalization process.

A team led by A. Kay worked on the development of the Smalltalk language at the Xerox company. When the Smalltalk-80 version [22] was created, it was clear that it was a valuable product that should be presented to the professional community. Through the language Smalltalk 80, a new Object-Oriented paradigm was proposed. The basic concept is an object. Programming is realized through a set of objects that interact with each other. Windows, thumbnails and icons are used intensively in the language, and a mouse is often used instead of a keyboard. The emphasis is on the graphical user interface. Programs written in Smalltalk are several times slower than those written in C. This is the main reason that Smalltalk was used on a small number of computers. The speed of computers is their most important property, so even the news brought by Smalltalk could not immediately influence the choice of language. However, these innovations are important for digitalization, so we can say that Smalltalk has significantly contributed to digitalization even though it was not directly used.

### **Programming languages created at the end of the 20th century and digitalization**

Programming languages from this period are still the most commonly used today. All of them (or packages created using them) enable working with various documents, images, sounds, ... It can be concluded that these languages fully support digitalization.

The C++ programming language was created as an object-oriented superstructure of the C language. Its creator is Stroustrup [24]. C++ supports procedural and object-oriented programming style. It is used in MS Office, Adobe Photoshop, and various types of high-performance software. Despite that, C++ is a complex language, it is accepted by professional programmers and according to various researches, it is one of the 5 most used programming languages all the time.

The emergence of the Java language, which was created at Sun Microsystems under the leadership of James Gosling [25], contributed a lot to the decline in popularity of the C++ language. Java is a universal, object-oriented language with C syntax, but simpler than C++. From its inception, it was oriented towards the Internet. By using Bytecode, Java programs are largely independent of the platform they run on. This is very important in the Internet era. Java is used for: web development, artificial intelligence, Android development, and enterprise software (from smartcards to space vehicles). It is enough to list these areas and understand how important they are to digitalization. Java is a compiler-

interpreter language. Programs written in Java are slower than corresponding programs in C/C++, but that does not prevent it from being among the most popular programming languages for years.

C# is a Microsoft product modeled on Java (although Microsoft claims that it is derived from C++ and Visual Basic). The language is used in almost all Microsoft products and is seen primarily in developing desktop applications. The features of C# are defined by using a standard library adapted to the .NET platform.

The fourth, chosen language from this period is Python [26], [27] whose creator is Guido Van Rossum. Python is a universal script language suitable for: web and Internet development, scientific computation, database access, education, software and game development, etc. Python's popularity is contributed by its support for different programming styles. Python is a language on the rise and is becoming one of the most popular programming languages in the world. It is used by software companies Google, Facebook, Spotify,... It can be said that Python supports the cutting-edge technology in the software industry. Due to the mentioned features, Python has a significant role in the digitalization process.

### **Future programming languages and digitization**

The question arises as to which programming languages are used the most nowadays. Languages created at the end of the 20th century are still the most popular today. They are joined by several script languages, among which JavaScript stands out. It is certain that in the near future the most important programming languages will be: C++, Java, C#, Python and JavaScript. These languages would not survive in competition with newly created ones if they were not constantly improving and adapting according to new trends. For example, the C++ language was modified in 2003, 2011, 2014, 2017 and 2020, and appropriate versions are obtained: C++03, C++11, C++14, C++17 and C++20. Java and C# have been improved in a similar way. Which programming language will be the most used is difficult to predict. Support for visualization, the Internet and digitalization will be crucial for the choice of programming language. Quantum computing will also influence the choice of programming language, as well as AI support.

Since its appearance, the popularity of the JavaScript language has grown. JavaScript [31] was created by Brendan Eich in 1995. It is universal, multi-paradigm programming language, supporting object-oriented, imperative and functional paradigm. JavaScript is used to create interactive web applications and games. There is no a modern browser without JavaScript support. JavaScript is primarily used on the client side in dynamic web applications. It can also be used on the server side, however, the PHP script language is used much more often on the server side. The JavaScript-PHP combination is often used in web applications.

We also included Swift as a promising programming language, although its popularity is not at the level of the previously mentioned ones. Swift appeared in 2014 and inherited the good features of the following languages: Objective-C, Rust, Haskell, Ruby, Python and C#. It is intended only for Apple computers and that is one of its shortcomings. It can be used on cellular devices, cloud applications, desktop computing, etc.

One of the promising languages is the language Go (or Golang) - a product of the company Google in 2007. It was created with the intention of replacing C, but to be less complex than C++. It is often used to create smartphone applications, design databases,



cloud computing, etc. It is a language that is easy to learn, fast and efficient. Go is used by companies: Google, Uber, Twitch and others.

Among the programming languages that will be used in the future, most experts also include script languages: PHP, Ruby, Perl, Scala and Groovy. We will not consider them here.

### Related work

ERP (Enterprise Resource Planning) systems play a very important role in digitalization. It is about complex software created by using various programming languages. In the paper [28] various ERP systems and the programming languages used in them are considered. Paper [29] contains a description of the impact of digitalization on the choice of programming language. Paper [4] describes the impact of digitalization on language in general. The programming languages essential for digital transformation in Japan are listed in the paper [30]. Here, the emphasis is on the banking sector. On the Internet, one can find a large number of sites about the popularity of programming languages and their future. We will not list them here, but we point out that the popularity of a programming language depends significantly on its support to digitalization.

### Conclusion

With the development of programming languages, their role in the digitalization process grew. On the genealogical tree of higher level programming languages, 4 periods are distinguished, and in each period several of the most famous languages (in the author's opinion) are selected. The contribution to digitalization for each of them is analyzed and some of their most important properties are listed. Some of these languages are directly involved in the digitalization process, while others are indirectly involved. At the time of the appearance of the first high level programming languages the concept of digitalization was not known (as we understand it today). Therefore, the languages of this period could not be directly included in the digitalization. Already the languages of the middle period (for example C) could be directly included. Almost all modern programming languages support digitalization.

### References

- [1] <https://en.wikipedia.org/wiki/Digitalization>
- [2] <https://www.walkme.com/glossary/digitalization/>
- [3] <https://www.eurofound.europa.eu/en/topic/digitalisation>
- [4] Samsudin, Samsudin & Mujab, Syaiful & Safar, Muh & Munandar, Haris. (2023). The effect of digitalization on language and culture in management practice modern education. *Lingua*. 19. 135-150. 10.34005/lingua.v19i2.3131.
- [5] Lidiia Oliinyk, Nataliia Romaniuk, Halyna Kuznetsova, Inna Horbenko, Nadiia Senchylo-Tatlilioglu, The impact of digital and internet technologies on language development, *Eduweb*, julio-septiembre, 16:3(2022), 41–54.
- [6] Stojković V. and Tošić D.: *Zbirka zadataka iz programiranja - programski jezik BASIC*, Privredno-finansijski vodič, Beograd, 1982.
- [7] <https://github.com/stereobooster/programming-languages-genealogical-tree>
- [8] <http://rigaux.org/language-study/diagram.html>

- [9] Tošić D.: "XML-tehnologije i digitalizacija", Pregled nacionalnog centra za digitalizaciju, 3, (2003), pp. 1-12.
- [10] Tošić D.: FORTRAN 77 - Zbirka rešenih zadataka, Tehnička knjiga, Beograd, 1989.
- [11] Budimac Z., Ivanović M., Putnik Z. i Tošić D.: LISP kroz primere, Institut za matematiku PMF-a, Novi Sad, 1991,
- [12] Budimac Z., Ivanović M., Bađonski M. i Tošić D.: Programski jezik SCHEME, Univerzitet u Novom Sadu – PMF, 1998.
- [13] Stojković V., Tošić D., Stojmenović I.: Programski jezik Pascal, Naučna knjiga, Beograd, 1990.
- [14] Tošić D.: Pascal-osnovi programiranja, Studentski trg, Beograd, 1977.
- [15] Tošić D. i Stojković V.: Programski jezik Pascal – zbirka rešenih zadataka, Tehnička knjiga, Beograd, 1991.
- [16] Peslak A.: Computer programming languages in 2020: what we use, who uses them, and how do they impact job satisfaction, Issues in Information Systems, Volume 21, Issue 2, pp. 259-269, 2020.
- [17] <https://www.tiobe.com/tiobe-index/>
- [18] The Top Programming Languages 2023 - IEEE Spectrum.
- [19] Steele G.L. and Gabriel R. P.: (1993), The Evolution of Lisp, in 'The Second ACM SIGPLAN Conference on History of Programming Languages (HOPL-II)', Cambridge, MA, USA, ACM, New York, pp. 231–270.
- [20] Alain Colmerauer and Philippe Roussel: The birth of Prolog, ACM SIGPLAN Notices, 28:3, 37–52, <https://doi.org/10.1145/155360.155362>
- [21] Tošić D. i Protić R.: PROLOG kroz primere, Tehnička knjiga, Beograd, 1991.
- [22] Kay, A. C. (1993), The Early History of Smalltalk, in The Second ACM SIGPLAN Conference on History of Programming Languages (HOPL-II), Cambridge, MA, USA, ACM, New York, pp. 69–95
- [23] Ritchie D. M.: (1993), The development of the C Language, in 'The Second ACM SIGPLAN Conference on History of Programming Languages (HOPL-II)', Cambridge, MA, USA, ACM, New York, pp.201-208.
- [24] Stroustrup B.: (1993): A History of C++: 1979-1991 in 'The Second ACM SIGPLAN Conference on History of Programming Languages (HOPL-II)', Cambridge, MA, USA, ACM, New York, pp. 271-298.
- [25] Kartelj A., Filipović V. i Tošić D.: Objektno orijentisano programiranje – programski jezik Java – 1, Matematički fakultet, Beograd, 2023.
- [26] Miloš A. Kovačević: Osnove programiranja u Pajtonu, Građevinski fakultet u Beogradu, 20021.
- [27] Eric Matthes: Python intenzivni kurs, Čačak, Kompjuter biblioteka, 2023.
- [28] Zečević A., Đ. Stakić i D. Đurđić: The role and importance of programming languages in the application of ERP solutions, Ekonomika preduzeća 71:7-8(2023), 448–459.
- [29] Starchikov C.: Digitalisation is the battlefield or how to choose your programming languages? 2020, <https://www.linkedin.com>.
- [30] Digital Transformation in Japan: Revealing Top Programming Languages, Tntra, 2023, <https://medium.com>
- [31] Marijn Haverbeke: JavaScript elokventno, Stručna knjiga, 2019.