

MODERN PROGRAMMING LANGUAGES (LINGUISTIC ASPECTS)

Mukhiddinov Anvar Gafurovich

*Doctor of Philology, Professor
Fergana State University*

Isroilova Nozima Kozimjon kizi

*1st year master's student,
direction: Linguistics (Russian language)
Fergana State University*

Annotation. *This article is devoted to the study of modern programming languages from a linguistic point of view. The paper examines the main programming languages currently in use and their features in the context of the language structure. The research methodology includes analysis of the syntax, vocabulary and semantics of programming languages. The results of the work show that programming languages have much in common with natural languages, but also have their own specific features. It concludes that studying the linguistic aspects of programming languages can help developers create more efficient and usable languages.*

Key words and expressions: *programming languages, linguistics, syntax, vocabulary, semantics.*

Introduction.

Programming languages are special languages that are used to write computer programs. There are many different programming languages, each of which has its own characteristics and is used in different areas. This article examines programming languages from a linguistic point of view, that is, their features in the context of the language structure.

II. Literature review.

In the 1990s, in connection with the active development of the Internet, languages that allow you to create scripts for web pages became widespread - mainly Perl, which evolved from a scripting tool for Unix systems, and Java. New virtualization technologies have been developed. These innovations were improvements to already existing paradigms and languages (in the latter case, mainly the programming language of the C family). At the new stage of IT development, Java has become the most popular and successful programming





language. This programming language was created by Dr. J. A. Gosling, known as the “Father of Java.” Since January 2010, Oracle has been developing new versions of this language. Currently, Java is one of the most popular languages, which is based on the WORA principle (Write Once Run Anywhere). Java has become one of the fundamental reasons for the success of this language in the enterprise environment due to its platform independence. Nowadays, programming languages are widely used in various fields, ranging from software development to data analysis and artificial intelligence. Studying the linguistic aspects of programming languages allows you to better understand their structure and features, which can be useful for both professional programmers and beginners.

III. Main part (methodology, results).

As a starting point for solving the problem of constructing a syntax tree, the system of generators of the Lex and Yacc compilers is used. The generators are modified so that instead of the source code of the analyzing finite state machines, which must first be compiled, ready-to-use analyzer structures are immediately created. The specification file format used is the original Lex and Yacc vocabulary and syntax specification file formats, modified in terms of describing the actions performed during convolution according to syntactic rules. Lexical rules are described as a set of regular expressions, which are associated with control constructs that regulate the return of detected lexemes to the parser. Syntactic rules are descriptions of language structures that are associated with control structures that regulate the construction of an abstract syntax tree, identifier tables, namespace trees and other elements [1].

The object being studied is modern programming languages. Among them are such main languages as Java, Python, C++, C#, JavaScript and others. Each of these languages has its own characteristics in syntax, vocabulary and semantics.

The methodology used is the analysis of the syntax, vocabulary and semantics of programming languages. During the research, code examples in various programming languages were analyzed in order to identify common and distinctive features.

The results of the work showed that programming languages have much in common with natural languages. For example, most programming languages use parsing to determine the structure of a sentence. Also, programming languages have their own vocabulary, consisting of keywords and identifiers. [4,5].

However, programming languages also have their own specific features. For example, the C++ programming language uses pointers that allow you to work with computer memory. The Python programming language has a concept called





dynamic typing, which means that the type of a variable can change during program execution. One of the most common programming languages is Java. It is an object-oriented language and has strong typing. Java uses many constructs such as classes, methods, interfaces, and inheritance, making it useful for creating large and complex applications. Additionally, Java has many libraries and frameworks that make software development easier.

Python is an interpreted programming language that is used for a variety of purposes, including scientific computing, web development, and machine learning. Python has a simple and straightforward syntax, making it easy to learn and use. It also supports dynamic typing and has many libraries and frameworks that make software development easier.

C++ is a programming language that is used to create high-performance code. It has a more complex syntax than Java or Python, but allows more precise control of computer resources. C++ is also used to develop games and operating systems.

To exchange information in the global media space, complex programming languages are used, including HyperText Markup Language (HTML), a standardized language for marking up web pages on the World Wide Web. HTML code is interpreted by browsers; the resulting page is displayed on the screen of a computer monitor or mobile device.

Each programming language has its own characteristics and applications, and learning their linguistic aspects can help programmers use their capabilities more effectively. For example, learning the syntax and structure of a language can help programmers write more readable and understandable code, making it easier to maintain and develop [6].

In addition, studying the linguistic aspects of programming languages can be useful for developing new programming languages or improving existing ones. For example, studying semantics and syntax can help create a more intuitive and user-friendly programming language for novice programmers.

In general, studying the linguistic aspects of programming languages can be useful for various fields related to information technology. This can help improve the quality of software, create new programming languages, and improve the efficiency of software development in general.

The study of the linguistic aspects of programming languages has shown that they have much in common with natural languages, but also have their own specific features. Learning these features can help developers create more efficient and easier-to-use programming languages. [5].





In today's world, where information technology plays an increasingly important role, knowledge of programming languages is becoming increasingly valuable. Learning the linguistic aspects of programming languages can help programmers be more efficient and productive in their work. It can also help improve software quality and create new innovative information technology solutions. Overall, learning the linguistic aspects of programming languages is important and beneficial for anyone working or interested in programming and information technology.

The study revealed common and distinctive features of programming languages. However, it should be noted that each programming language has its own unique features that can be useful in specific tasks. For example, the R programming language is used for statistical data analysis, and the Swift programming language is used for developing mobile applications for the iOS operating system.

Learning the linguistic aspects of programming languages can lead to improvements in the software development process. Knowledge of programming languages and their features allows programmers to write more readable and understandable code, which in turn can speed up the debugging process and improve the quality of the final product.


Additionally, learning the linguistic aspects of programming languages can help programmers communicate more effectively with other developers and clients. Understanding the terminology and syntax of programming languages can simplify communication and help avoid misunderstandings.

Finally, studying the linguistic aspects of programming languages can help programmers create new and innovative information technology solutions. Understanding the characteristics of programming languages can help programmers create new algorithms and data structures, which can lead to the creation of new products and services.

Overall, learning the linguistic aspects of programming languages is important and beneficial for anyone working or interested in programming and information technology. This can help improve software quality, speed up the development process, and create new innovative solutions.

Learning the linguistic aspects of programming languages can help developers create more efficient and easier-to-use languages. Additionally, understanding the features of programming languages can help students and aspiring programmers learn new programming languages faster and more efficiently. Learning the linguistic aspects of programming languages has many benefits and can help





programmers become more efficient in their work. This can help improve code quality, simplify communication with other developers and clients, and create new innovative information technology solutions. Therefore, anyone who wants to become a successful programmer should take the time to learn the linguistic aspects of programming languages.

Natural language as a quasi-semiotic system, cognitive units of the “language - thinking” dichotomy are a prerequisite for the creation of artificial languages, in particular, programming languages and artificial intelligence. Natural language and thinking will continue to dominate in the era of technological singularity, when computers will be exponentially smarter than humans.


From the above, the following conclusions can be drawn:

Thus, the study of modern programming languages from a linguistic point of view allows us to better understand their features and apply them more effectively in various tasks. The development of programming languages will continue, and the study of their linguistic aspects will remain a relevant area of research. In general, studying the linguistic aspects of programming languages is an important element in developing the professional skills of programmers. This allows them to better understand and use different programming languages and create more efficient and readable code. In addition, knowledge of the linguistic features of programming languages can help programmers better understand customer requirements and improve communication within the development team. Thus, learning the linguistic aspects of programming languages is an important element in the professional development of programmers and can help them create better software.

BIBLIOGRAPHY

1. Aho A. V., Lam M. S., Sethi R., Ullman D. D. Compilers. Principles, technologies and tools. M.: Williams, 2008. 1184 p.
2. Aho A.W., Lam M.S., Sethi R. and Ullman J.D. (2006). Compiled by: principles, methods and tools. Pearson Education. M.: Williams, 2008. 1184 p.
3. Dyakonov V. P., Borisov A. V. Fundamentals of artificial intelligence. – Smolensk, 2007. –193 p.
4. Alfred W. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Compilers: Principles, Technologies and Tools = Compilers: Principles, Techniques, and Tools. — 2nd ed. - M.: Williams, 2008.
5. Sebesta R.U. Basic concepts of programming languages = Concepts of Programming Languages / Transl. from English; 5th ed. M., 2001.





6. Sinzhapova, R. R. Theory of conceptual integration and linguistic means of its expression / R. R. Sinzhapova. — Text: immediate // Young scientist. — 2019. — No. 2 (240). — P. 393-395.

