

## THE EVOLUTION OF PROGRAMMING LANGUAGES: A MULTINATIONAL ACADEMIC PERSPECTIVE

Mukhammadova Rahimakhon Asadjon qizi

Fergana State University

Philology and Language Teaching (English) 1st year student

Toshboltayev Faxriddin O'rinboyevich

Scientific advisor: p.f.b.f.d (PhD)

<https://doi.org/10.5281/zenodo.19755232>

**Abstract:** This article examines the evolution of programming languages based on Uzbek and international academic sources. It describes the main stages of development, from machine language to modern high-level programming systems. The paper also discusses key programming paradigms such as procedural, object-oriented, and functional programming, and their role in software development. The study is based on qualitative analysis of scientific literature. The results show that programming languages have evolved toward greater abstraction, usability, and efficiency, and are now closely connected with modern technologies such as artificial intelligence and cloud computing.

**Keywords:** programming languages, evolution, paradigms, software development, history

**Annotatsiya:** Ushbu maqolada dasturlash tillarining rivojlanish tarixi o'zbek va xalqaro ilmiy manbalar asosida tahlil qilinadi. Unda mashina tilidan boshlab zamonaviy yuqori darajadagi dasturlash tizimlarigacha bo'lgan asosiy bosqichlar yoritilgan. Shuningdek, protseduraviy, obyektga yo'naltirilgan va funksional dasturlash paradigmlarining ahamiyati ko'rib chiqilgan.

Tadqiqotda ilmiy adabiyotlarni tahlil qilish usuli qo'llanilgan. Natijalar shuni ko'rsatadiki, dasturlash tillari vaqt o'tishi bilan soddalashib, samaradorligi oshgan va zamonaviy texnologiyalar bilan chambarchas bog'langan.

**Kalit so'zlar:** dasturlash tillari, rivojlanish, paradigma, tarix, dasturiy ta'minot

**Аннотация:** В статье рассматривается эволюция языков программирования на основе узбекских и международных научных источников. Описываются основные этапы развития — от машинного языка до современных высокоуровневых систем. Также анализируются ключевые парадигмы программирования и их влияние на разработку программного обеспечения.

Исследование основано на анализе научной литературы. Результаты показывают, что языки программирования развиваются в направлении повышения уровня абстракции, удобства использования и эффективности, а также тесно связаны с современными технологиями.

**Ключевые слова:** языки программирования, эволюция, парадигмы, разработка ПО, история

**Introduction.** Programming languages are essential instruments that enable communication between humans and computers. Their evolution reflects the broader development of computer science and information technology. Early computing systems relied heavily on machine-level instructions, requiring programmers to interact directly with hardware components. This approach was inefficient and prone to errors, as emphasized in Uzbek educational sources on informatics ([Abduqodirov, 2019](#)).

International research literature also highlights that early programming required detailed knowledge of processor architecture and memory management, limiting accessibility to a small group of specialists (Sebesta, 2012). Similarly, South Asian academic materials stress that the complexity of early programming practices created a demand for more user-friendly and abstract programming tools (Balagurusamy, 2017).

The purpose of this study is to provide a comprehensive overview of the historical evolution of programming languages by integrating insights from different academic traditions. This approach allows for a more balanced and globally informed understanding of programming language development.

**Methods.** This study employs a qualitative research methodology based on literature review and comparative analysis. Sources include Uzbek textbooks on information technology, internationally recognized academic books, and widely used South Asian programming manuals.

A chronological framework is used to examine the development of programming languages across different generations. Additionally, a comparative method is applied to analyze how different programming paradigms have evolved and influenced software development practices in various educational contexts.

**Results.** Machine language and early computing. The earliest stage in programming language development is associated with machine language. This low-level language consists of binary instructions that are directly executed by the computer's processor. Uzbek sources emphasize that while machine language offers maximum efficiency, it is extremely difficult for humans to understand and use (Po'latov, 2020).

International studies support this view, noting that machine-level programming required precise control of hardware resources, making it time-consuming and error-prone (Tanenbaum, 2014). South Asian authors similarly describe this period as the “foundation stage” of programming, where efficiency was prioritized over usability (Balagurusamy, 2017).

Assembly Languages and Symbolic Representation. The development of assembly languages marked a transition from binary code to symbolic representation. Instead of using numerical instructions, programmers could write mnemonics such as “ADD” and “MOV.”

According to Uzbek academic literature, this innovation significantly improved readability while still maintaining close interaction with hardware (Karimov, 2018). However, international sources point out that assembly languages remained hardware-dependent, limiting their portability (Sebesta, 2012).

High-level languages and Abstraction. The introduction of high-level programming languages in the mid-20<sup>th</sup> century was a major breakthrough. Languages such as FORTRAN, COBOL, and C allowed programmers to write instructions using syntax closer to human language.

Uzbek textbooks emphasize that high-level languages reduced programming complexity and increased productivity (Rasulov, 2021). Similarly, global research highlights that abstraction enabled software to be developed more efficiently and made programming accessible to a wider audience (Louden, 2011).

South Asian sources also underline the importance of structured programming introduced in languages like C, which improved code organization and maintainability.

Emergence of Programming Paradigms. The evolution of programming languages led to the development of different programming paradigms. Procedural programming focuses on step-by-step instructions, while object-oriented programming (OOP) organizes code into objects.

Uzbek sources highlight that OOP significantly improved modularity and code reuse (Karimov, 2018). International literature further explains that languages such as Java and C++ popularized OOP due to their scalability and flexibility (Sebesta, 2012).

Functional programming, another paradigm, is based on mathematical functions and immutability. Modern research shows that this paradigm is particularly useful in parallel computing and data processing environments (Louden, 2011).

#### Modern Programming Languages and Trends

Modern programming languages such as Python, JavaScript, and Go are designed with simplicity and efficiency in mind. They are widely used in web development, artificial intelligence, and data science.

Uzbek academic sources note that these languages are increasingly integrated with cloud technologies and large-scale data systems (Abduqodirov, 2019). International studies confirm that modern programming focuses on scalability, cross-platform compatibility, and developer productivity (Tanenbaum, 2014).

South Asian educational materials also emphasize the growing importance of languages that support rapid application development and real-time processing (Balagurusamy, 2017).

**Discussion.** The findings indicate that the evolution of programming languages has been driven by the need to improve usability, efficiency, and flexibility. Each stage of development has introduced new features that address the limitations of previous languages.

A key trend observed across all academic traditions is the increasing level of abstraction. While early programming required detailed hardware knowledge, modern languages allow developers to focus on solving complex problems.

Another important observation is the diversity of programming paradigms. Different paradigms provide unique approaches to problem-solving, enabling developers to choose the most appropriate method depending on the context.

Furthermore, globalization of education has contributed to a shared understanding of programming concepts. Uzbek, international, and South Asian sources all emphasize similar trends, highlighting the universal nature of programming language evolution.

**Conclusion.** In conclusion, the evolution of programming languages reflect the continuous advancement of computer science and global collaboration in education and research. From machine-level programming to modern intelligent systems, programming languages have become more accessible, powerful, and versatile.

The integration of perspectives from Uzbek, international, and South Asian sources demonstrates that programming language development follows common global patterns. Future programming languages are expected to focus on artificial intelligence, automation, and improved human-computer interaction.

### Adabiyotlar, References, Литературы:

1. Abduqodirov, A.A. (2019). Axborot texnologiyalari. Toshkent: O'qituvchi.
2. Po'latov, B.B. (2020). Informatika va dasturlash asoslari. Toshkent.

3. Karimov, N.R. (2018). Dasturlash tillari. Toshkent: Fan.
4. Rasulov, T.X. (2021). Dasturiy ta'minot asoslari. Toshkent.
5. Sebesta, R.W. (2012). Concepts of Programming Languages. Pearson.
6. Tanenbaum, A.S. (2014). Structured Computer Organization. Pearson.
7. Louden, K.C. (2011). Programming Languages: Principles and Practice. Cengage Learning.
8. Balagurusamy, E. (2017). Programming in C. McGraw-Hill Education.