

262. HISTORY OF PROGRAMMING LANGUAGES

Lebed S. P., Lazovsky T. S.

*Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Ryabykh V. A. - Lecturer, Master of Arts

This paper examines the historical development of programming languages, tracing their evolution from early conceptual algorithms to modern high-level languages. The result is a concise chronological overview of key languages and their influence on subsequent generations.

Programming languages enable human-machine interaction. Their history spans nearly two centuries, beginning before electronic computers. Origins and first languages (1843–1950s). In 1843, Ada Lovelace created the first machine algorithm, enabling general-purpose computing. In 1936, Alan Turing established the theoretical foundations of computer science. Konrad Zuse developed Plankalkül in the early 1940s, the first programming language for computers. Assembly language (1949) simplified machine code. In 1952, Autocode became the first compiled language. FORTRAN (1957) was the first high-level language to gain widespread adoption [1].

From the beginning of 1958 until 1972, there was an era of the spread of languages. ALGOL (1958) became the progenitor of influential languages including Pascal, C, and Java. LISP (1958) was designed for artificial intelligence research. COBOL (1959) was designed to operate across all computer types for business applications. BASIC (1964) was developed for students without technical backgrounds. Pascal (1970) was designed for teaching structured programming. C (1972) became the foundation for C++, C#, Java, JavaScript, Perl, PHP, and Python. SQL (1972) was developed for database management. Smalltalk (1972) introduced true object-oriented programming.

Somewhere in the 1980s, object-oriented and scripting languages were being created. Ada, named after Ada Lovelace and developed by Jean Ichbiah's team, was commissioned by the U.S. Department of Defense in the 1980s for mission-critical systems [2]. C++ emerged in 1983 when Bjarne Stroustrup extended C with classes and virtual functions at Bell Labs. Objective-C, developed by Brad Cox and Tom Love in 1983, became Apple's primary language. Perl, created by Larry Wall in 1987, was designed as a scripting language for text processing.

Somewhere starting from 1990 and up to 1995, internet revolutions were happening. Python, developed by Guido van Rossum in 1991, emphasized code readability. Visual Basic, released by Microsoft in 1991, introduced drag-and-drop GUI programming. Ruby, created by Yukihiro Matsumoto in 1993, prioritized programmer happiness and web application development. Java, developed by James Gosling at Sun Microsystems in 1995, enabled cross-platform development through the Java Virtual Machine. JavaScript, created by Brendan Eich in 1995, transformed web development with dynamic page interactions [1]. PHP, developed by Rasmus Lerdorf in 1995, became a server-side web development language powering Facebook and Wikipedia.

Well, at the moment, starting from the year 2000 and up to our time, the era of modern languages is happening. C#, developed by Microsoft in 2000, combined elements of C++ and Visual Basic for the .NET framework. Scala, created by Martin Odersky in 2003, merged functional and object-oriented programming on the Java Virtual Machine. Go, developed by Google in 2009, addressed challenges in large-scale software systems with built-in concurrency support. Swift, introduced by Apple in 2014, was designed to replace C, C++, and Objective-C for Apple ecosystem development [3].

Key Evolutionary Patterns. The evolution of programming languages reveals key patterns. Fortran is a foundational ancestor; C and Java became the most influential in industry. Successful languages solve real problems, evolve while maintaining compatibility, and often combine multiple paradigms.

From Ada Lovelace's 1843 algorithm to modern languages like Go and Swift, programming languages have continuously evolved to meet changing computational needs. The foundational work of early pioneers remains relevant, with concepts from the 1950s embedded in today's most modern languages.

References:

1. A Timeline of Programming Languages. IEEE Computer Society. – URL: <https://www.computer.org/publications/tech-news/insider-membership-news/timeline-of-programming-languages> (date of access: 12.03.2026).
2. Computer History: A Timeline of Computer Programming Languages. – URL: <https://www.hp.com/gb-en/shop/tech-takes/computer-history-programming-languages> (date of access: 12.03.2026).
3. Alekseeva, V. P. Development of Programming Languages / V. P. Alekseeva, E. R. Kovalieva. — Москва : МИПЭА — Российский технологический университет, 2023. — 85 с.