

PROGRAMMING LANGUAGES

A programming language is a formal language intended for recording computer programs. A programming language defines a set of lexical, syntactic and semantic rules that determine the appearance of a program and the actions that the performer (usually the computer) will perform under its control. A programming language is designed to write computer programs that are a set of rules that allow a computer to perform a particular computational process, organize the management of various objects, etc. A programming language differs from natural languages in that it is designed to control a computer, while as natural languages are used primarily for communication among people.

Low-level programming language is a programming language that is close to programming directly in the machine codes of a real or virtual (for example, byte-code, Microsoft, NET) processor used. For the designation of machine commands usually mnemonic designation is used. This allows you to memorize commands not as a sequence of binary zeros and ones, but as meaningful abbreviations of the words of the human language (usually English).

Sometimes one mnemonic designation corresponds to a whole group of machine instructions that perform the same action on different memory cells of the processor. In addition to machine commands, low-level programming languages may provide additional features, such as macros. With the help of directives, it is possible to control the process of translating machine codes, providing the ability to write constants and literal strings, reserve memory for variables, and place executable code at specific addresses. Often, these languages allow you to work instead of specific memory cells with variables.

High-level programming language is a programming language designed for speed and ease of use by the programmer. The main feature of high-level languages is an abstraction, that is, the introduction of semantic structures, briefly describing such data structures and operations on them, which descriptions on machine code (or other low-level programming language) are very long and difficult to understand.

High-level programming languages have been developed for the platform independence of the essence of the algorithms. Dependence on the platform is shifted to instrumental programs — translators compiling text written in a high-level language into elementary machine instructions (instructions). Therefore, a platform-unique translator is developed for each platform for each high-level

language, for example, translating text written in Delphi in elementary x86 microprocessor commands.

Thus, high-level languages seek not only to facilitate the solution of complex software problems, but also to simplify the porting of software. The use of various translators and interpreters provides communication of programs written in high-level languages with various operating systems, programmable devices and equipment, and, ideally, does not require modification of the source code (text written in a high-level language) for any platform.

Why are so many programming languages? The answer is obvious - many of them were created for different machines, for different purposes and applications. This is for serious languages. Some of them are just invented for the sake of fun, but they are few and they are not widely spread, of course. And there are about eight thousand five hundred programming languages!