

Loseu M., Lebed A., Vanik I.

## **Classification of Programming Languages**

Belarusian National Technical University  
Minsk, Belarus

A programming language is the medium of communication between a user and a computer system. There are basically three types of programming languages, they are Machine level language (MLL), Assembly level language (ALL) and High-level language (HLL) [1].

Machine level language is the lowest level and is named as the first generation of programming languages. Machine level language consists only of two conditions i.e. either true (1) or false (0); this type of language is known as a binary language. A computer system can understand only a binary language.

The advantages of machine level language include the following: machine level languages are directly interacting with a computer system; there is no requirement of software of conversion like a compiler or interpreter; it takes very less time to execute a program, because there is no conversion. However, a machine language has a number of disadvantages: it's a machine dependent language i.e. individual program is required for each machine; it's rather time-consuming to develop a new program on a machine language; the debugging process is very hard.

Assembly level language is a middle level language and is named as the second-generation programming language. It contains the same instruction as machine level language, but the instructions and the variables have a specific name. They are called commands instead of being just binary numbers.

As for the advantages of Assembly language, they are: it is easily understood by the humans because this language uses statements instead of binary digits; it takes less time to develop a program; debugging and troubleshooting is not hard due to the ease of finding errors. There are several disadvantages of Assembly language: it's a machine dependent language due to that program design for one machine is not suitable for another machine; sometimes it's hard to understand the statement or a command use [1].

High level language is the upper level language and is also known as the third-generation programming language. It is considered as high level because any language that comes under this category is close to human languages. There are many examples of high-level languages such as, Fortran, Pascal, C, C++, Java, Ada, Cobol, LISP, Prolog etc.

The strong points of a high-level language comprise the following: the instructions and commands are much easier to be remembered by a programmer; the logic and structure a high-level language are much easier to understand; debugging is easier compared to other languages; it's less time-consuming to write new programs. The weak points of high-level languages include: HLL programming language takes more space compared to MLL (machine level language) and/or ALL (Assembly level language); the execution of this type of programming language is slow.

Next, programming languages are classified as static / dynamic typed. Static typed programming languages are those in which variables don't need to be defined before they're used. This implies that static typing has to do with the explicit declaration (or initialization) of variables before they're employed. Java is an example of a static typed language; C and C++ are also static typed languages. Note that in C (and C++ also), variables can be cast into other types, but they don't get

converted; you just read them assuming they are of another type.

Dynamic typed programming languages are those languages in which variables must be defined before they are used. This implies that dynamic typed languages do not require the explicit declaration of the variables before they're used. Python is an example of a dynamic typed programming language, and so is PHP [2].

Further, programming languages are categorized as compiled or interpreted. With a compiled language, the code you enter is reduced to a set of machine-specific instructions before being saved as an executable file. For example, C, C++ are compiled languages. With interpreted languages, the code is saved in the same format that you entered. For instance, Python is an interpreted language. Compiled programs generally run faster than interpreted ones because interpreted programs must be reduced to machine instructions at runtime. However, with an interpreted language you can do the things that cannot be done in a compiled language.

#### References:

1. Classification of computer programming languages [Electronic resource]. – Mode of access: <https://www.er.yuvayana.org/definition-classification-of-computer-programming-languages/> – Date of access: 28.04.2019.
2. Introduction to static and dynamic typing [Electronic resource]. – Mode of access: <https://www.sitepoint.com/typing-versus-dynamic-typing/> – Date of access: 28.04.2019.