УДК 044. 43+811.93

DISTRIBUTION OF PROGRAMMING LANGUAGES: HISTORY AND TRENDS

Stud. M.S. Marenych, gr. BIT1-17
Scientific supervisor S.O. Krasniuk
Kyiv National University of Technology and Design

Introduction. Thoughts on which language is best taught is different: from the fact that programming is not necessary, but should simply raise computer literacy and develop office programs (as in the West), to the fact that it's necessary to study operating systems and several programming languages of different levels of abstraction and with different paradigms. These are extreme cases, but the golden mean is difficult to find. First and foremost, we need to define the goal. Learn to think logically and algorithmically? Getting to know computers at home level so we can use the internet, email and text editors? To lay the basic knowledge necessary for future engineers, mathematicians, physicists and information technology professionals? Or maybe we need to introduce everyone to programming as a phenomenon, that represents the potential of computer systems? Will many of us become successful programmers? A bit. But Kirchoff's sinuses and equations in life, too, are not used by everyone.

The purpose of this study is to analyze the methods of distributing programming languages, to select and argue the appropriate method of studying programming languages.

Setting objectives:

1) to analyze the existing methods of distributing programming languages;

2) to choose expedient methods;

3) to argue your choice;

4) to conduct a survey among a social group of people, studying the language; to conduct a survey among the social group of people who teach programming languages;

5) to compare and make conclusions from the conducted questionnaires;

6) to summarize and highlight the main arguments for the need to apply the chosen method of learning programming languages.

Object and subject of research are programming languages as a subject of study and effective methods of studying.

Methods and means of research are gathering general information on this topic, analysis of existing methods of studying programming languages, conducting a survey among people who are learning programming languages [1].

Scientific novelty and practical significance of the results obtained is to identify and generalize errors that are admitted in the initial stage of the study. For the first time, common effective methods of consistent study of the material were found: logical thinking, algorithmization and the structure of the programming language.

Results of research In the science of programming is a fundamental component, but it is not easy to define it. Some believe that it's not so important which programming language is chosen for learning: first of all, you need to learn not programming languages, but programming methods and the system approach to problem solving. It is necessary to develop algorithmic thinking and on examples to get acquainted with the principles of constructing modern computer systems. Is it really not so important what environment and which specific programming language will be used for practical classes? It turns out that each teacher has his own list of requirements for the programming language [2]. For example: a simple, intuitive syntax, the presence of high-level tools for detecting and preventing errors and for debugging programs, the availability of quality documentation with examples, the existence of a friendly

development environment, interplatformism. For some teachers, this list is very short, for example: "Only Pascal" or "Anyone other than BASIC!"

The more a language tutorial is searched, the more popular the language is assumed to be. It is a leading indicator. The raw data comes from Google Trends.

If you believe in collective wisdom, the PYPL Popularity of Programming Language index can help you decide which language to study, or which one to use in a new software project.

Rank	Language	Share	Trend
1	Java	22.62 %	-0.8 %
2	Python	22.05 %	+5.2 %
3	Javascript	8.56 %	+0.2 %
4	PHP	8.22 %	-1.8 %
5	C#	7.95 %	-0.7 %
6	C	6.38 %	-1.1 %
7	R	4.26 %	+0.4 %
8	Objective-C	3.7 %	-1.0 %
9	Swift	2.92 %	-0.6 %
10	Matlab	2.31 %	-0.4 %
11	Ruby	1.7 %	-0.4 %
12	TypeScript	1.58 %	+0.5 %
13	VBA	1.39 %	+0.0 %
14	Visual Basic	1.27 %	-0.3 %
15	Scala	1.2 %	-0.0 %
16	Kotlin	0.92 %	+0.8 %
17	Go	0.84 %	+0.3 %
18	Perl	0.78 %	-0.1 %
19	lua	0.37 %	-0.2 %
20	Rust	0.36 %	+0.0 %
21	Haskell	0.31 %	-0.0 %
22	Delphi	0.29 %	-0.1 %

Conclusions. When choosing a programming language, the main criterion is the result. And the result - it is as soon as possible to master the knowledge and practical skills in this language and in general direction. So the main criteria for this choice are:

- the need for a given language and direction in general on the market;
- low entrance threshold, that is, the simplicity of this language for mastering;
- how much the audience likes this programming language.

REFERENCES

1. Robert Sebesta. Concepts of Programming Languages, 10th Edition, Pearson, 2012, p. 296.
2. https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/Closures.html#//apple_ref/doc/uid/TP40014097-CH11-ID94
3. Koetsier, Teun (2001). On the prehistory of programmable machines; musical automata, looms, calculators. PERGAMON, Mechanisma and Machine Theory 36. pp. 589–603.
4. Dijkstra, Edsger W. On the foolishness of "natural language programming." Archived 20 January 2008 at the Wayback Machine. EWD667.
5. Richard L. Wexelblat: History of Programming Languages, Academic Press, 1981, chapter XIV.