

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/297909273>

Trends And Developments In The Programming Languages Usage

Conference Paper · January 2012

CITATIONS

0

READS

1,445

1 author:



Vladimir Sulov

University of Economics Varna

27 PUBLICATIONS 67 CITATIONS

SEE PROFILE

Sulov V.S. ©

Assoc. prof., Ph.D., University of Economics, Bulgaria, Varna

TRENDS AND DEVELOPMENTS IN THE PROGRAMMING LANGUAGES USAGE

Programming languages are an important factor in the development of information technology today as more and more computer and mobile devices and software are used in all spheres of life. The aim of this paper is to gather, summarize and present statistical data for the usage of different programming languages and to analyse their trends and developments.

Some sources [8] list more than 8000 programming languages that have been created so far. Others [2; 11] list «only» about 2500 languages considering the rest as dialects, variants or versions. Some of these languages are long forgotten and not used any more but a great deal are still active. It would be quite a difficult job to try to describe them and analyse which of them are important and should be included in our study. That is why we will choose a different filtering approach: a reality check. We will assume that the most important programming languages are the ones that are actually and predominantly used in the software industry as it definitely drives the market and even computer science in general.

In order to determine which are the most widely used programming languages, we will present and combine data from several studies conducted by different organizations. As things change quite rapidly, most of the data is presented on the Internet as up-to-date statistics. It would be inappropriate to gather information about thousands of languages as it turns out (see below) that the top 15 most used languages represent more than 80% of all usage, and languages outside the top 15 list have each less than 1% popularity. That is why we have chosen this limit for inclusion in the paper.

We will present and combine data from 3 studies (which in turn have slightly different methodologies and a multitude of sources).

Table 1 shows data from the Transparent Language Popularity Index [12]. The measuring mechanism is automated using results from general web sites, blogs, book sales, wikis, open-source projects, jobs and videos using 7 different sources.

Table 1

Popularity of Programming Languages according to the Transparent Language Popularity Index

Rank	Programming Language	Popularity
1	Java	19.086%
2	C	16.878%
3	Objective-C	9.545%
4	Basic	7.534%
5	C++	6.062%
6	PHP	5.518%
7	C#	4.113%
8	Python	3.571%
9	Perl	3.524%
10	JavaScript	2.180%
11	Delphi	1.850%
12	Ruby	1.545%
13	D	1.342%
14	R	1.288%
15	Pascal	1.116%

Table 2 presents data from the TIOBE Programming Community Index [13]. The index is determined by a weigh-based system using 8 sources.

Table 2

Popularity of Programming Languages according to the TIOBE Programming Community Index

Rank	Programming Language	Popularity
1	Java	17.05%
2	C	16.52%
3	C#	8.65%
4	C++	7.85%
5	Objective-C	7.06%
6	PHP	5.64%
7	(Visual) Basic	4.32%
8	Python	3.15%
9	Perl	2.93%
10	JavaScript	2.47%

The end of the table 2

Rank	Programming Language	Popularity
11	Delphi/Object Pascal	1.96%
12	Ruby	1.56%
13	Lisp	0.91%
14	Transact-SQL	0.85%
15	Pascal	0.81%

Table 3 presents data from the Programming Language Popularity site [13]. The raw data (taken from 12 sources) is in absolute numbers but for our purposes it is interpreted in percentage.

Table 3

Popularity of Programming Languages according to the Programming Language Popularity Site

Rank	Programming Language	Popularity
1	C	14.19%
2	Java	13.60%
3	C++	11.63%
4	PHP	9.27%
5	JavaScript	7.49%
6	Python	5.72%
7	C#	4.53%
8	Perl	4.34%
9	Transact-SQL	3.94%
10	Ruby	3.35%
11	(Visual) Basic	2.37%
12	Objective-C	0.79%
13	Lisp	0.79%
14	Delphi/Object Pascal	0.59%
15	Pascal	0.39%

If we consider the three ranking systems we can extract the programming languages that appear in all of them, namely: Java, C, C++, Objective-C, C#, (Visual) BASIC, PHP, Python, Perl, JavaScript, Delphi (Object Pascal), Pascal, and Ruby. The languages that do not appear in all three charts will be excluded: R, D, Lisp, Transact-SQL.

Applying a simple not-weighted average produces our final results as shown on Table 4.

The resulting 13 most widely used programming languages can be classified into 3 groups according to their target use:

- for local, desktop development (including locally executable software for portable devices such as tablets, cell phones);

- for web development;
- suitable for both targets.

Table 4

Combined Average Popularity of Programming Languages

Rank	Programming Language	Popularity
1	Java	16.58%
2	C	15.86%
3	C++	8.52%
4	PHP	6.81%
5	Objective-C	5.80%
6	C#	5.77%
7	(Visual) Basic	4.74%
8	Python	4.15%
9	JavaScript	4.05%
10	Perl	3.60%
11	Ruby	2.15%
12	Delphi/Object Pascal	1.47%
13	Pascal	0.77%

According to this classification the distribution of programming languages is as follows:

- Languages suitable only for local, desktop development: C, C++, Objective-C, Delphi, Pascal. The total absolute share of the group is approx. 32% and 40% relatively to our target selection (the 13 languages).
- Languages suitable only for web development: PHP, Python, JavaScript, Perl, Ruby. The total absolute share of the group is approx. 21% and 26% relatively to our target selection.
- Languages suitable both for local, desktop and web development: Java, C#, Visual BASIC. The total share of the group is approx. 27% and 34% relatively to our target selection.

We will briefly analyze the main trends in application development.

Past data shows that most programming languages suitable only for local, desktop development have declining usage patterns with only pure C relatively stable in the long-term, although fluctuating in the short-term (see Figure 1). This can be explained by conversion to generally- and memory managed code [1], rapid application development [3] and languages producing such for simpler applications. Large applications are still predominantly written using C and C++ [4, 10].

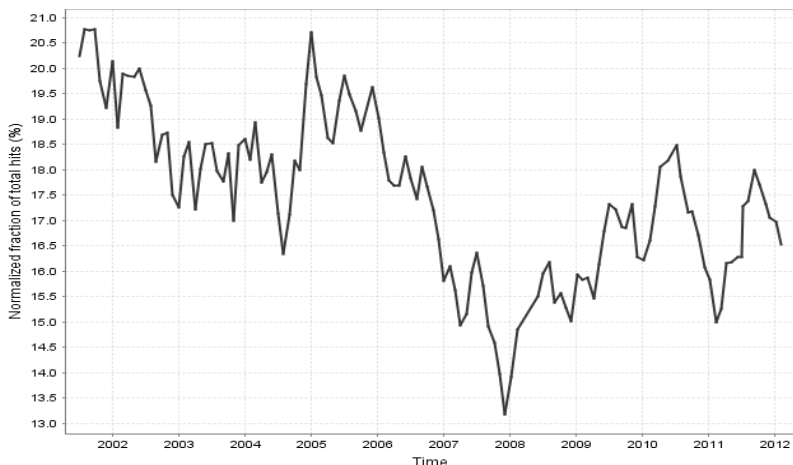


Figure 1. Popularity Trend of the C Programming Language [13]

A notable growth trend can be observed in this group as Objective-C is steadily on the rise (see Figure 2). This can be attributed to the success of Apple as its driven force and mainly its portable devices like iPhone and iPad (after the introduction of iPhone in 2007 and iPad in 2010) which use applications written in Objective-C.

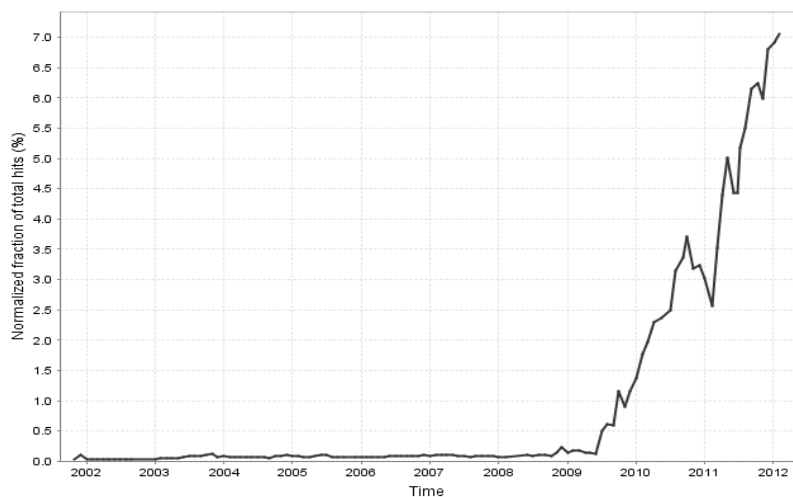


Figure 2. Popularity Trend of the Objective-C Programming Language [13]

The languages suitable only for web development are relatively the smallest group and their share continues to decline. In no way this means that web development is deteriorating. On the contrary: more and more applications are developed for the web but in the same time developers need to target both desktop and web, often utilizing a common code base. These circumstances call for languages that can be used both for local, desktop and web development. We think that this is one of the main reasons that web-only languages are losing share and languages that are suitable both for the desktop and web are gaining share. Another reason is that often web-only languages lack real power.

The group of languages suitable for both desktop and web development represents 34% of the top 13 list and is still gaining popularity. A typical example in this group is Java (which is the all-group leader) while the most notable long-term trends (characteristic of the above-mentioned developments) are the decline of Perl (see Figure 3) and the rise of C# (see Figure 4).

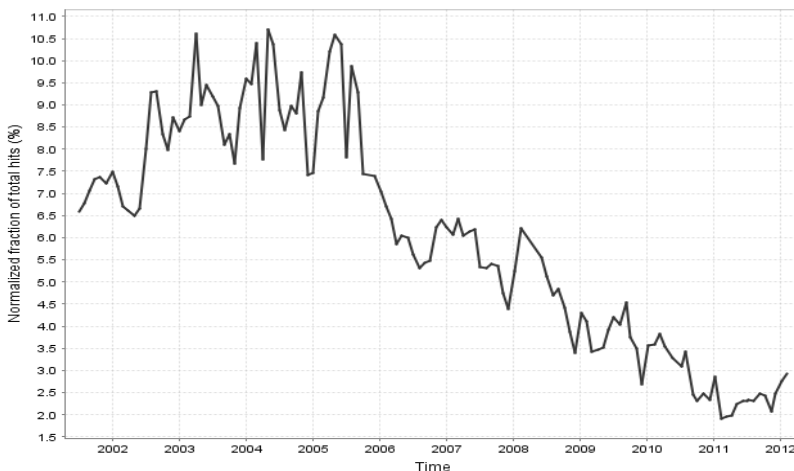


Figure 3. Popularity Trend of the Perl Programming Language [13]

As a conclusion we can point out that the usage of programming languages is constantly fluctuating but no new languages have entered the market in the last 10 years. Software development is driven by business needs and economic efficiency more than it is driven by improvements and advances in the languages themselves. Therefore we can assume that both academia and industry should focus on languages that can be used and adapted for a variety of tasks and applications.

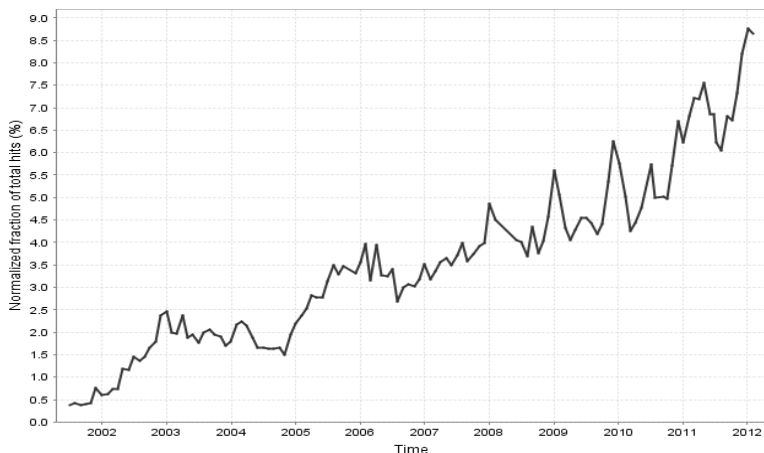


Figure 4. Popularity Trend of the C# Programming Language [13]

References

- 1 Aho A.V., Lam M.S., Sethi R., Ullman J.D. Compilers: Principles, Techniques, and Tools. Addison Wesley, 2006.
- 2 Bergin T.J., Gibson R.G.. History of Programming Languages, Volume 2. Addison-Wesley Professional, 1996.
- 3 Martin J. Rapid Application Development. Macmillan, 1991.
- 4 Park R. Software Size Measurement: A Framework for Counting Source Statements. Mode of access: <http://www.sei.cmu.edu/pub/documents/92.reports/pdf/tr20.92.pdf> (11.03.2012).
- 5 Pratt T.W., Zelkowitz M.V. Programming Languages: Design and Implementation. Prentice Hall, 2001.
- 6 Sebesta R. W. Concepts of Programming Languages. Addison Wesley, 2012.
- 7 Watt D.A. Programming Language Processors: Compilers and Interpreters. Prentice Hall, 1993.
- 8 HOPL: an interactive Roster of Programming Languages. Mode of access: <http://hopl.murdoch.edu.au>. 10.03.2012.
- 9 Programming Language Popularity. <http://langpop.com/>. 7.03.2012.
- 10 Source Lines of Code. Wikipedia. Mode of access: http://en.wikipedia.org/wiki/Source_lines_of_code (10.03.2012).
- 11 The History of Programming Languages. Mode of access: http://www.oreilly.com/news/graphics/prog_lang_poster.pdf (9.03.2012).
- 12 The Transparent Language Popularity Index. <http://lang-index.sourceforge.net>. 7.03.2012.
- 13 TIOBE Software: Tiobe Index. Mode of access: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> (7.03.2012).