**PRAKTIČNA PRIMENA JEZIKA MALOG VOKABULARA: M**

Matija Čupić

*Ignjata Joba 3*

# Uvod

Jezici velikog vokabulara najšire su rasprostranjeni bez obzira na paradigmu (Objektno Orijentisani, Proceduralni, … ). Nameće se pitanje praktične primene jezika na drugom kraju spectra tj. onih sa malim vokabularom. Na koji način vokabular jezika utiče na magnitudu i opširnost projekata napisanih u njemu, kao i platforme pogodne za razvoj takvim jezikom biće tema ovog istraživanja.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Active** | **Push** | **Year** |
| JavaScript | 323,938 | 3,461,415 | 1995 |
| Java | 222,852 | 2,323,315 | 1995 |
| Python | 164,852 | 1,654,226 | 1991 |
| PHP | 138,771 | 1,391,467 | 1995 |
| C# | 56,062 | 558,332 | 2000 |
| C++ | 86,505 | 1,013,761 | 1983 |
| Swift | 11,138 | 64,575 | 2014 |
| Go | 22,264 | 196,130 | 2009 |
| Clojure | 6,840 | 50,208 | 2007 |

Tabela 1 – Rasprostranjenost najpopulanrijih jezika na GitHub-u

Kao rezultat istraživanja predviđena je implementacija namenskog programskog jezika sa malim vokabularom kao i primeri njegove praktičnosti i poređenje sa jezicima opšte primene.

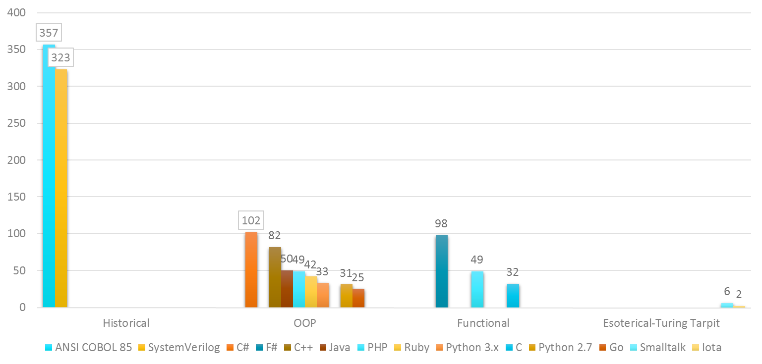
# Metoda Rada

Napravljen je pregled ključnih reči najpopularnijih programskih jezika prema njihovim specifikacijama, kao i uzorci magnitude projekata iz javnih GitHub repozitorijuma i izvršena analiza odnosa broja ključnih reči, veličine projekta kao i popularnost datog jezika.

Na osnovu dobijenih informacija izrađena je specifikacija jezika opisanog u uvodu. Izvršena je implementacija istog, kao i izrada razvojnog okruženja koje dodatno olakšava izradu projekata služeći se i potpomažući iste principe na kom je jezik zasnovan.

# Rezultati Istraživanja

Potvrđena je početna hipoteza o nedostatku primenjivosti opštih programskih jezika za POC projekte i dizajniran i implementiran M programski jezik kao i razvojno okruženje za mobilne platforme, namenjeno razvoju takvih projekata.



Slika 1 – Distribucija ključnih reči u programskim jezicima

# Zaključak

Iz analize rezultata istraživanja zaključujemo da trend opštosti programskih jezika svakako dominantan i superioran, ali u određenim segmentima ima i svoje nedostatke. Ti nedostaci se ne mogu nadomestiti u okviru dotičnih jezika jer je koren problema u ideologiji jezika, a ne u njihovoj implementaciji.

Takav nadostatak se, ipak, veoma lako nadomešćuje implementacijom zasebnog namenskog programskog jezika, namenjenog razvoju manjih projekata, koji su vrlo često POC (proof of concept) projekti, sa mogućnošću razvoja na mobilnim platformama.

# Literatura

1. F. Turbak, D. Gifford, M. A. Sheldon; Design Concepts in Programming Languages; The MIT Press 2008
2. A. Aho, M. Lam, R. Sethi, J. Ullman; Compilers: Principles, Techniques, and Tools; Pearson Education 2006
3. B. Pierce; Types and Programming Languages; The MIT Press 2002
4. R. Harper; Practical Foundations for Programming Languages; Cambridge University Press 2012
5. G. Winskel; Formal Semantics of Programming Languages; The MIT Press 1993
6. A. Appel; Compiling with Continuations; Cambridge University Press 2007
7. H. Abelson, G. Sussman, J. Sussman; Structure and Interpretation of Computer Programs; The MIT Press 1996

**PRACTICAL APPLICATION OF A SMALL VOCABULARY LANGUAGE: M**

Matija Čupić

*Ignjata Joba 3*

# Introduction

Languages with large vocabularies are the most widespread, without regard for their paradigm (Object Oriented, Procedural, …). A question imposes itself: Are languages on the other end of the spectrum (small vocabulary ones) practical? In what way the vocabulary of a language influences the magnitude and scope of projects written in it, as well as what platforms are suitable for development in such languages are the subject of this research.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Active** | **Push** | **Year** |
| JavaScript | 323,938 | 3,461,415 | 1995 |
| Java | 222,852 | 2,323,315 | 1995 |
| Python | 164,852 | 1,654,226 | 1991 |
| PHP | 138,771 | 1,391,467 | 1995 |
| C# | 56,062 | 558,332 | 2000 |
| C++ | 86,505 | 1,013,761 | 1983 |
| Swift | 11,138 | 64,575 | 2014 |
| Go | 22,264 | 196,130 | 2009 |
| Clojure | 6,840 | 50,208 | 2007 |

Table 1 – Distribution of the most popular languages on GitHub

An implementation of a specific programming language with a small language with examples of its practicality and a comparison with other all-purpose languages is the goal of this project.

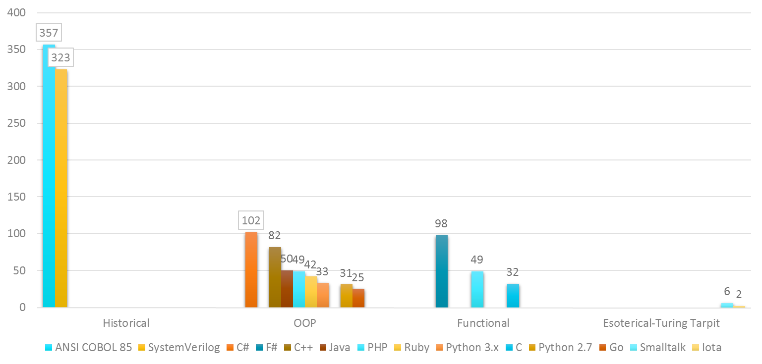
# Method of operation

A summary of keyword density of the most popular programming languages according to their specifications was made. Samples were taken from public GitHub repositories and an analysis of the relationship between the number of keywords, size of the project and popularity of the language was made.

According to the acquired information, a specification of a language described in the Introduction was made. The language was implemented, and an IDE which eases the development of projects using the same principles on which the language was designed upon was made.

# Results

The original hypothesis about the lack of practicality of general programming languages for POC projects was confirmed. A programming language was designed and implemented, as well as an IDE for mobile platforms intended for development in that language.



Picture 1 – Keyword distribution in programming languages

# Conclusion

From the analysis of the research results we conclude that the trend of generality of programming languages is certainly dominant and superior, but in specific segments it has flaws. That flaws can’t be overcome in the frame of the language itself, because the flaw is in the ideology of the language and not in its implementation.

That kind of flaw is, however, very easily overcome by implementing of a separate specific programming language designed for development of small projects, which are very often POC projects with the possibility of development mobile platforms.

# Literature

1. F. Turbak, D. Gifford, M. A. Sheldon; Design Concepts in Programming Languages; The MIT Press 2008
2. A. Aho, M. Lam, R. Sethi, J. Ullman; Compilers: Principles, Techniques, and Tools; Pearson Education 2006
3. B. Pierce; Types and Programming Languages; The MIT Press 2002
4. R. Harper; Practical Foundations for Programming Languages; Cambridge University Press 2012
5. G. Winskel; Formal Semantics of Programming Languages; The MIT Press 1993
6. A. Appel; Compiling with Continuations; Cambridge University Press 2007
7. H. Abelson, G. Sussman, J. Sussman; Structure and Interpretation of Computer Programs; The MIT Press 1996