

УДК 004.438

ББК 3973.2

С.О. ИВАНОВ, Д.В. ИЛЬИН, И.Ю. БОЛЬШАКОВ

СРАВНИТЕЛЬНОЕ ТЕСТИРОВАНИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Ключевые слова: языки программирования, тестирование, фрактал Мандельброта, C, C#, Java, JavaScript, RUST, Go, Swift.

Данная статья посвящена исследованию влияния языка программирования на эффективность и надежность программного обеспечения. Известно, что выбор языка программирования влияет на характеристики создаваемых программ. В данной работе производится сравнение ведущих языков программирования (C, C#, Java, JavaScript, RUST, Go и Swift) с помощью тестирования написанных на них программ. Для получения объективных и сравнимых результатов на каждом языке программирования написана программа, которая решает одну задачу – вычисление фрактала Мандельброта одним и тем же способом. Были выбраны следующие характеристики: скорость выполнения, объем используемой памяти и лаконичность исходного кода. В результате было выделено несколько языков-лидеров в своей области, ни один язык программирования не стал лучшим по всем критериям.

В настоящее время существует огромное количество языков программирования. Их более 2500, но наиболее значимых – около 700, и это без учета их диалектов. Споры от том, какой среди них лучший, ведутся давно. Существует объективно наблюдаемая разница в работе программ, написанных на разных языках. Так, например, функциональный язык SISAL в 6 раз медленнее C и в 2,5 раза больше использует оперативную память [6]. В каждом языке имеет место определенная зависимость между выразительностью языка и его эффективностью, а также существует множество других критериев, по которым можно оценить язык программирования. Таким образом, задача выбора языка программирования является сложной, многокритериальной задачей с большим (теоретически бесконечным) количеством оцениваемых объектов.

Выбор «живого» перспективного языка важен и для обучения, и для написания оптимальной программы как по производительности, так и по скорости написания и удобству дальнейшего сопровождения. Так, например, у ведущих языков программирования отсутствуют возможности для уточнения, автоматической верификации, безопасности [9]. С необходимостью выбора языка программирования чаще всего сталкиваются программисты при изучении первого языка программирования. Здесь на выбор первого языка программирования оказывает влияние множество факторов: дидактические, педагогические, отраслевые и технологические [7]. В работе [1] отмечено, что выбор самими учащимися подходящего языка программирования для решения конкретной задачи повышает их интерес к изучаемой дисциплине. Правильный выбор средств и инструментов оказывает существенное влияние на результат любого инженерного проекта, особенно для таких критических систем, как космические спутники [2, 4].

Существуют различные подходы к решению проблемы выбора языка программирования. В данной статье производится сравнение «продвинутых»

языков программирования с традиционными «ведущими» языками для выявления «перспективных» языков.

Особенности сравнения языков программирования. Среди огромного количества существующих языков программирования широко используются около 50. В различных рейтингах, например в TIOBE Index [13], учитываются только 100 наиболее популярных. Поэтому выбор языка программирования можно сократить до 100 из постоянно обновляемого списка TIOBE Index.

Языки программирования используются для различных задач. Хотя большинство языков являются универсальными, некоторые являются более подходящими для конкретных целей. Можно выделить следующие независимые группы целей: прототипирование (скорость) – проверка концепций и экспериментирование; переносимость (легкость) – многократно используемый, свободно распространяемый код; производство (надежность) – создание ПО для заказчика, быстрый, надежный, системный код. Таким образом в каждой категории будет свой язык-«лидер».

Так как язык программирования – это абстрактное описание вычислительного процесса, то объективно измерить можно только сам процесс и его результаты. Суть подобных методов сравнения языков программирования заключается в том, что оценивается код программы, написанной на разных языках. При этом следует учитывать разницу между языком программирования и программой при использовании этого метода и интерпретации результатов.

Рассмотрим особенности этого метода сравнения языков программирования.

1. *Зависимость от задачи.* Некоторые языки предназначены для решения определенных задач, поэтому выбор задачи для тестирования оказывает существенное влияние на результаты. Наиболее распространенный подход к выбору задач [5, 6, 8] – это использовать вычислительно трудоемкие задачи, такие, как функция Аккермана, тринолиномы, серии экспоненциальных функций и др. В других работах [11, 12] задачи акцентированы на возможностях языка по распараллеливанию вычислений – проблемы Куочена (Cowichan), синхронизации и «бутылочного горлышка», эффективности коммуникации. Все предыдущие тесты основаны на реальных задачах из практики, что позволяет говорить о существовании оптимальных решений, на нахождение которых ориентированы языки программирования и программисты. Для проверки других случаев используются синтетические тесты [3, 10], проверяющие комбинации операций – whetstone, перебор комбинаций вариантов выполняемых действий.

2. *Зависимость от алгоритма.* Алгоритм решения выбранных задач также оказывает влияние на результаты сравнения, так как языки программирования предоставляют разные вычислительные примитивы (абстракции, конструкции). В работе [6] отмечают сильное влияние на производительность кода выбора рекурсии вместо итерации. Работы [11, 12] посвящены оценке влияния различных подходов к распараллеливанию на эффективность программы. Необходимо отметить также влияние на производительность программы эффективности реализации операций над структурами данных (массивы и списки) и средств ввода-вывода (необходимость дополнительного перекодирования данных).

Таким образом, для сопоставимости результатов программы для каждого языка должны быть либо самыми производительными, либо точно следовать

одному алгоритму. Для нахождения наилучших программ для решения определенной задачи на разных языках можно использовать системы, используемые в олимпиадах по программированию. Этот подход используется в существующем проекте [5].

3. *Зависимость от компилятора.* Компиляторы отличаются возможностями оптимизации и параметрами компиляции – результаты тестирования одной и той же программы, скомпилированной с разными параметрами, могут существенно отличаться. Для усложнения оптимизации в работе [3] используются трудно оптимизируемые для компиляторов выражения. Многие компиляторы для разных языков программирования используют также одинаковый «back-end», например *llvm* для *clang*, *swift*, *rust* и т.п., который выполняет окончательную оптимизацию и генерацию кода. Таким образом результаты тестирования в первую очередь будут показывать, насколько хорошо работает «back-end» и насколько удобен язык программирования для него.

4. *Зависимость от платформы.* Достаточно очевидным фактом является то, что производительность компьютерной платформы непосредственно влияет на результаты выполнения кода. Так, например, машина Тьюринга теоретически может выполнить любую известную вычислительную задачу, но на практике ее реализация совершенно неэффективна и по удобству программирования, и по размеру кода, и по эффективности вычисления. Поэтому для сопоставимости результатов необходимо либо использовать только одну платформу для всех тестов, либо ввести поправочный коэффициент. Для вычисления коэффициента можно использовать результаты выполнения одного и того же эталонного кода. Кроме производительности платформы обладают различным набором вычислительных средств и компонент среды, что делает невозможным проведение некоторых тестов, например в некоторых случаях необходимо наличие веб-браузеров и сетевых средств [10].

Критерии тестирования. Рассмотрим критерии, с помощью которых можно оценить языки программирования:

1) скорость выполнения – время, затрачиваемое на выполнение программы – показывает производительность языка;

2) используемая память – объем необходимой оперативной памяти – показывает экономичность языка;

3) размер исходного кода – количество символов в программе без учета разделителей – показывает лаконичность языка;

4) размер результирующей программы – размер исполняемой программы – показывает громоздкость получаемых программ;

5) поддерживаемые абстракции – набор конструкций и парадигм – показывает удобство языка;

6) поддерживаемые специальные средства – количество компонент для решения специальных задач – показывает развитость языка;

7) популярность среди программистов и работодателей – распространенность языка – показывает перспективность языка.

Скорость выполнения программы можно считать основным критерием оценки языка программирования. С учетом используемых объемов памяти сегодня объем используемой памяти и размер результирующей программы не

существенны. Оценить популярность языка можно по специальным индексам, например TIOBE Index [13].

Для сравнения языков программирования были выбраны следующие критерии: время работы программы от запуска до конца ее выполнения; количество символов в исходном коде программы данного языка.

Тестирование языков программирования. Для сравнения были выбраны известные языки программирования: C, C#, Java, JavaScript, RUST, Go и Swift. На каждом из приведенных языков была написана линейная программа, которая выводила символьный рисунок фрактала Мандельброта размером 4096 на 4096 символа. Все программы имели один алгоритм и имели идентичные типы используемых переменных.

Пример программы на языке C:

```
#include <stdio.h>
#include <stdlib.h>

double bias = 5.0;
int maxIter = 64;

char mandelbrot(double p, double q){
double x = 0;
double y = 0;
char res = 0;

while((x*x + y*y < bias) && (res < maxIter)){
double x1 = x, y1 = y;
x = x1*x1 - y1*y1 + p;
y = 2*x1*y1 + q;
res++;
}
return res;
}

int main(int argc, char **argv){
double fac;
char *buf;
int xi,yi;
int size = 1024;

if(argc > 1){
sscanf(argv[1], "%d", &size);
}

fac = 2.0 / size;
buf = (char *)calloc(size*size+1,sizeof(char));

for(yi=0; yi<size; yi++){
for(xi=0; xi<size; xi++){
double y = (yi*fac - 1.0);
double x = (xi*fac - 1.0);

buf[yi*size+xi] = mandelbrot(x,y)+ '0';
}
buf[yi*size + size - 1] = '\n';
}
buf[size*size] = 0;

printf("Mandelbrot: %d, %d\n",size,size);
puts(buf);
free(buf);
}
```

Все тесты были выполнены на выделенной удаленной машине, на которой была установлена ОС Ubuntu Server 16.04. Результаты приведены в таблице.

Результаты тестирования

Язык программирования	Среднее время выполнения, с			Количество символов	
	общее	вне ядра	в ядре	с пробелами	без пробелов
RUST	2:46	2:34	0:04	953	612
C	2:90	2:09	0:13	813	618
JavaScript	3:22	3:20	0:17	768	591
Swift	4:23	0:13	0:19	857	806
Java	4:21	2:90	0:24	1418	881
Go	7:09	6:20	0:97	750	575
C#	7:49	6:17	0:27	1161	750

В данном эксперименте по скорости выполнения программы новый системный язык программирования RUST оказался быстрее, чем классический системный язык Си. Язык Go по общему времени выполнения не оказался на последнем месте, но затратил существенно больше времени на вычисления, чем остальные языки. Схожие по синтаксису и возможностям языки программирования Java и C# показали одинаковую скорость вычисления, но на запуск программы на C# ушло больше времени.

Самым лаконичным языком оказался Go, а самым избыточным Java, что соответствует принципам, использовавшимся при их создании.

Выводы. Сравнение языков программирования способом тестирования программ, написанных на них, используется давно. В настоящее время основное внимание уделяется тестированию возможностей параллелизации и оптимизации. Но применение этого метода это сложная, многокритериальная задача, так как выразительность языка программирования зависит от его синтаксиса, а эффективность – от платформы, на которой выполняют программу. Поэтому однозначно выбрать лучший язык программирования невозможно, необходимо в зависимости от выбранных целей расставлять приоритеты между критериями. Например, по популярности среди работодателей уже несколько лет подряд первое место занимает язык программирования Java, который отстает по скорости и избыточен по синтаксису. При выборе языка программирования стоит помнить, что плохие, «неудачные» языки вымирают, поэтому выгоднее изучать перспективные языки.

Литература / References

1. Aparanji U., Kumar V. Improving Classroom Discussions of Programming Assignments via Language Choice. Proc. of 2016 IEEE Eighth Int. Conf. on Technology for Education (T4E), 2017. DOI: 10.1109/T4E.2016.017.
2. Blanco-Cuaresma S., Bolmont E. What can the programming language Rust do for astrophysics? Proc. of the Int. Astronomical Union, IAU Symposium, 2017, vol. 325, pp. 341–344.
3. Curnow H.J., Wichmann B.A. A synthetic benchmark. *Comput J* 1976; 19 (1): 43-49. doi: 10.1093/comjnl/19.1.43
4. Garrido J., Zamorano J., de la Puente J.A., Alonso A., Salazar E. ADA, the Programming Language of Choice for the UPMSat-2 Satellite. *Data Systems in Aerospace, Proceedings of the conference held 19-21 May, 2015 in Barcelona, Spain. Edited by L. Ouwehand. ESA-SP vol. 732, 2015, id. 41.*
5. Gouy I. The Computer Language Benchmarks Game. Available at: <http://benchmarks-game.alioth.debian.org>.
6. Hammes J., Wim Bohm A.P. On the Performance of Functional Programming Languages on Realistic Benchmarks. *PDPTA*, 1997, pp. 296–304.

7. Ivanović M., Budimac Z., Radovanović M., Savić M. Does the choice of the first programming language influence students' grades? *CompSysTech '15 Proc. of the 16th Int. Conf. on Computer Systems and Technologies*. Dublin, Ireland, 2015, pp. 305–312.
8. Knoebel A. Benchmarks of programming languages for special purposes in the space station. *NASA. Marshall Space Flight Center Research Reports: 1986 NASA (ASEE Summer Faculty Fellowship Program)*, 1986, 20 p.
9. Koenig J., Rustan K., Leino M. Programming Language Features for Refinement. *EPTCS* 209, 2016, pp. 87–106. DOI: 10.4204/EPTCS.209.7
10. Kuuttila M., Mantyla M., Raulamo-Jurvanen P. Benchmarking Web-testing – Selenium versus Watir and the Choice of Programming Language and Browser. *eprint arXiv:1611.00578*. 2016, 40 p.
11. Nanz S., West S., Soares da Silveira K. Benchmarking Usability and Performance of Multicore Languages. *Proc. of the 7th ACM-IEEE Int. Symp. Empirical Software Engineering and Measurement (ESEM'13)*, 2013, pp. 183–192. doi:10.1109/ESEM.2013.10
12. Ricci A., Ben-Gurion G.W., Riken A.Y. Proceedings of the 2013 workshop on Programming based on actors, agents, and decentralized control. *SPLASH '13 Conference on Systems, Programming, and Applications: Software for Humanity*. ACM New York, 2013, pp. 115–126.
13. TIOBE Index. Available at: <https://www.tiobe.com/tiobe-index> (Accessed 01.05.2017).

ИВАНОВ СЕРГЕЙ ОЛЕГОВИЧ – старший преподаватель кафедры математического и аппаратного обеспечения информационных систем, Чувашский государственный университет, Россия, Чебоксары (v101-11@mail.ru).

ИЛЬИН ДМИТРИЙ ВЛАДИМИРОВИЧ – кандидат физико-математических наук, заведующий кафедрой математического и аппаратного обеспечения информационных систем, Чувашский государственный университет, Россия, Чебоксары (destr@mail.ru).

БОЛЬШАКОВ ИВАН ЮРЬЕВИЧ – студент IV курса факультета информатики и вычислительной техники, Чувашский государственный университет, Россия, Чебоксары.

S. IVANOV, D. ILIN, I. BOLSHAKOV
BENCHMARK OF PROGRAMMING LANGUAGES

Key words: programming languages, benchmark, Mandelbrot Fractal, C, C#, Java, JavaScript, RUST, Go, Swift.

This article is devoted to studying the influence of a programming language on the efficiency and reliability of software. It is known that the choice of a programming language has a strong influence on the programs characteristics that are being created. In this paper we compare the leading programming languages: C, C #, Java, JavaScript, RUST, Go and Swift, using the testing of programs written on them. The features of this method of comparison and their influence on results are considered. To obtain the objective and comparable results in each programming language, the program is written that solves one task – the calculation of the Mandelbrot fractal, using the same algorithm. Among the available characteristics, the following were selected: the speed of execution, the amount of memory used, and the conciseness of the source code. As a result, several languages-leaders in their field were determined, but no programming language has become the best by all the criteria.

IVANOV SERGEY – Senior Lecturer of Math and Hardware Information Systems Department, Chuvash State University, Russia, Cheboksary (v101-11@mail.ru).

ILIN DMITRY – Candidate of Physical and Mathematical Sciences, Head of Math and Hardware Information Systems Department, Chuvash State University, Russia, Cheboksary.

BOLSHAKOV IVAN – Student of Informatics and Computing Faculty, Chuvash State University, Russia, Cheboksary.

Ссылка на статью: Иванов С.О., Ильин Д.В., Большаков И.Ю. Сравнительное тестирование языков программирования // Вестник Чувашского университета. – 2017. – № 3. – С. 222–227.