

Сравнение современных языков программирования

Семченко Регина Викторовна

*Приамурский государственный университет имени Шолом-Алейхема
студент*

Еровлев Павел Андреевич

*Приамурский государственный университет имени Шолом-Алейхема
студент*

Лучанинов Дмитрий Васильевич

*Приамурский государственный университет имени Шолом-Алейхема
Старший преподаватель кафедры информационных систем, математики и
правовой информатики*

Аннотация

В данной статье описан анализ некоторых современных языки программирования, проведено их сравнение согласно существующим критериям в использовании.

Ключевые слова: Язык программирования, программирование, алгоритм, цикл.

Comparison of modern programming languages

Semchenko Regina Viktorovna

*Sholom-Aleichem Priamursky State University
student*

Erovlev Pavel Andreevich

*Sholom-Aleichem Priamursky State University
student*

Luchaninov Dmitry Vasilyevich

*Sholom-Aleichem Priamursky State University
Senior lecturer*

Abstract

In the article the analysis of some modern programming languages is described, comparison according to existing criteria in use is presented.

Keywords: Programming language, programming, algorithm, cycle.

В последнее время появляется все больше языков программирования, но существует проблема их распространенности при наличии очевидных

преимуществ. Целью данного исследования является описание современных языков программирования и их сравнения согласно критериям.

Go – это язык программирования с открытым исходным кодом, который упрощает создание простого, надежного и эффективного программного обеспечения (рис.1). Go является выразительным, лаконичным, чистым и эффективным. Механизмы параллельности позволяют создавать программы, которые получают максимальную отдачу от многоядерных и сетевых машин, в то время как новая система типов позволяет создавать гибкую и модульную программу. Go быстро компилирует машинный код, имеет удобство сбора мусора. Это быстрый, статически типизированный, скомпилированный язык, который выглядит как динамически типизированный, интерпретируемый язык [1].



```
vn@vn-laptop: ~/go-code
File Edit View Terminal Help
vn@vn-laptop:~/go-code$ ls
helloworld.go
vn@vn-laptop:~/go-code$ cat helloworld.go
package main

import "fmt"

func main() {
    fmt.Printf("Hello world!\n")
}
vn@vn-laptop:~/go-code$ 8g helloworld.go
vn@vn-laptop:~/go-code$ 8l helloworld.8
vn@vn-laptop:~/go-code$ ls
8.out helloworld.8 helloworld.go
vn@vn-laptop:~/go-code$ ./8.out
Hello world!
vn@vn-laptop:~/go-code$ █
```

Рис. 1. Язык программирования Go

Julia – высокоуровневый высокопроизводительный язык динамического программирования для численных вычислений. Он обеспечивает сложный компилятор, распределенное параллельное выполнение, численную точность и обширную библиотеку математических функций (рис.2). Базовая библиотека Julia также объединяет зрелые, лучшие в своем классе с открытым исходным кодом C и Fortran библиотеки для линейной алгебры, генерации случайных чисел, обработки сигналов и обработки строк. Кроме того, сообщество разработчиков Julia вносит быстрый вклад в ряд внешних пакетов через встроенный менеджер пакетов Julia. Julia, сотрудничество между сообществами Jupyter и Julia, обеспечивает мощный браузерный графический интерфейс ноутбука для Julia.

Программы Julia организованы вокруг нескольких рассылок, что позволяет перегружать встроенные и пользовательские функции для различных комбинаций типов аргументов [2].

```

function mandel(z)
  c = z
  maxiter = 80
  for n = 1:maxiter
    if abs2(z) > 4
      return n-1
    end
    z = z^2 + c
  end
  return maxiter
end

function randmatstat(t)
  n = 5
  v = zeros(t)
  w = zeros(t)
  for l = 1:t
    a = randn(n,n)
    b = randn(n,n)
    c = randn(n,n)
    d = randn(n,n)
    P = [a b c d]
    Q = [a b; c d]
    v[l] = trace((P.*P)^4)
    w[l] = trace((Q.*Q)^4)
  end
  std(v)/mean(v), std(w)/mean(w)
end

```

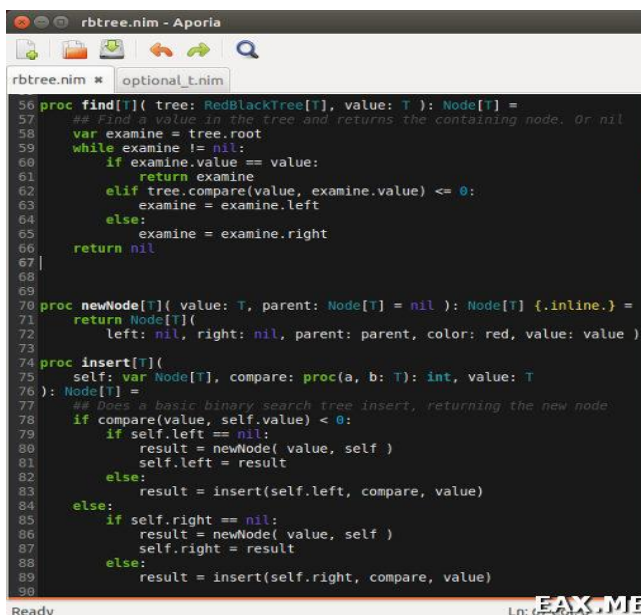
Рис. 2. Язык программирования Julia

LB Stanza – это новый, необязательно типизированный язык программирования общего назначения. Stanza был разработан, чтобы помочь программистам решить сложность архитектуры больших программ и значительно повысить производительность программистов приложений на протяжении всего жизненного цикла разработки программного обеспечения [3].

Nim – это язык программирования систем и приложений. Статически типизированный и скомпилированный, он обеспечивает беспрецедентную производительность в элегантном пакете (рис.3).

Его основные особенности [4]:

1. Высокопроизводительный сбор данных с мусором.
2. Компиляция на C, C++ или JavaScript.
3. Производит исполняемые файлы без зависимостей.
4. Работает в Windows, MacOS, Linux и т.д.



```

rbtree.nim * optional_t.nim
56 proc find[T]( tree: RedBlackTree[T], value: T ): Node[T] =
57   ## Find a value in the tree and returns the containing node. Or nil
58   var examine = tree.root
59   while examine != nil:
60     if examine.value == value:
61       return examine
62     elif tree.compare(value, examine.value) <= 0:
63       examine = examine.left
64     else:
65       examine = examine.right
66   return nil
67
68
69
70 proc newNode[T]( value: T, parent: Node[T] = nil ): Node[T] {.inline} =
71   return Node[T](
72     left: nil, right: nil, parent: parent, color: red, value: value )
73
74 proc insert[T](
75   self: var Node[T], compare: proc(a, b: T): int, value: T
76 ): Node[T] =
77   ## Does a basic binary search tree insert, returning the new node
78   if compare(value, self.value) < 0:
79     if self.left == nil:
80       result = newNode( value, self )
81       self.left = result
82     else:
83       result = insert(self.left, compare, value)
84   else:
85     if self.right == nil:
86       result = newNode( value, self )
87       self.right = result
88     else:
89       result = insert(self.right, compare, value)
90
Ready
Ln: 60/90

```

Рис. 3. Язык программирования Nim

Объявленный в 2014 году, язык программирования Swift быстро стал одним из самых быстрорастущих языков в истории. Swift позволяет легко писать программное обеспечение, которое невероятно быстро и безопасно по дизайну (рис.4). Цель Swift: упрощение программирования простых алгоритмов.

Так как Swift обладает открытым кодом, навыки в языке Swift могут быть применены к широкому диапазону платформ: от мобильных устройств до облачных технологий. [5].

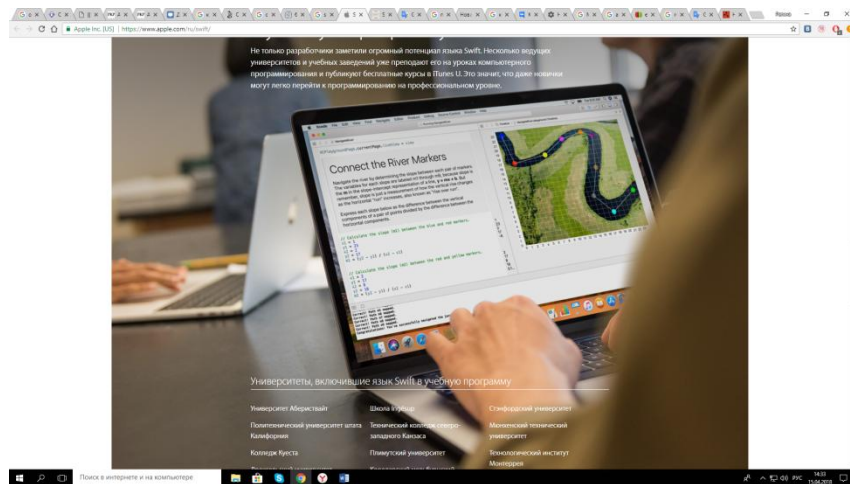


Рис. 4. Язык программирования Swift

Rust – невероятно быстрый язык для системного программирования без segfault'ов и с гарантиями потокобезопасности (рис.5) [6].

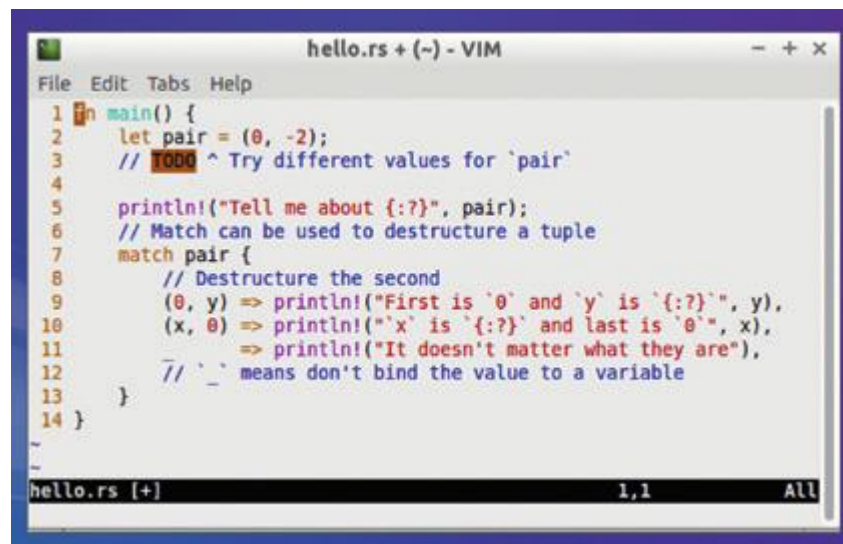


Рис. 5. Язык программирования Rust

Nack – это язык программирования для NHVM . Nack примиряет быстрый цикл разработки динамически типизированного языка с дисциплиной, предоставляемой статическим набором текста, добавляя много функций, обычно встречающихся в других современных языках программирования (рис.6).

Наск обеспечивает мгновенную проверку типов путем постепенной проверки файлов при их изменении. Обычно он работает менее чем за 200 миллисекунд, что упрощает интеграцию в рабочий процесс разработки без заметной задержки. Аннотации типа позволяют явно указывать код на параметры, переменные-члены класса и возвращаемые значения. Generics позволяют параметризовать классы и методы (т. е. тип, ассоциированный при создании экземпляра класса или вызов метода) в том же духе, что и языки статического типа, такие как C # и Java). Лямбда кратко описывает определение первоклассных функций. Другие существенные особенности Наск включают ХНР , Shapes , Type Aliasing , Async , обратную совместимость с PHP-кодом и многое другое [7].

```

1 <?php
2
3 function dbfetch() {
4     $r = mysql_query('SELECT ...');
5     if ($r === false || mysql_num_rows($r) == 0) {
6         return null;
7     }
8
9     return new DBData(mysql_fetch_array($r));
10 }
11
12 function test() {
13     $data = dbfetch();
14     $data->doFunStuff();
15 }

```

```

1 <?hh
2
3 function dbfetch(): ?DBData {
4     $r = mysql_query('SELECT ...');
5     if ($r === false || mysql_num_rows($r) == 0) {
6         return null;
7     }
8
9     return new DBData(mysql_fetch_array($r));
10 }
11
12 function test(): void {
13     $data = dbfetch();
14     $data->doFunStuff();
15 }

```

Рис. 6. Язык программирования Наск

Scala сочетает объектно-ориентированное и функциональное программирование на одном высокоуровневом языке (рис.7).

Статические типы Scala помогают избежать ошибок в сложных приложениях, а его время работы JVM и JavaScript позволяет создавать высокопроизводительные системы с легким доступом к библиотекам. [8].

Listing 1: Example kotlin code.

```

1 /* Block comment */
2 package hello
3 import kotlin.collections.* // line comment
4
5 /**
6  * Doc comment here for 'SomeClass'
7  * @see Iterator#next()
8  */
9 @Deprecated("Deprecated class")
10 private class MyClass<out T : Iterable<T>>(var prop1 : Int) {
11     fun foo(nullable : String?, r : Runnable, f : () -> Int,
12         fl : FunctionLike, dyn: dynamic) {
13         println("length\nis ${nullable?.length} \n")
14         val ints = java.util.ArrayList<Int?>(2)
15         ints[0] = 102 + f() + fl()
16         val myFun = { -> "" };
17         var ref = ints.size
18         ints.lastIndex + globalCounter
19         ints.forEach lit@ {
20             if (it == null) return@lit
21             println(it + ref)
22         }
23     }
24 }

```

Рис. 7. Язык программирования Scala

Рассмотрим возможности программ в таблице, где стоят баллы от 1 до 5.

Таблица 1. Результаты анализа языков программирования

Программа	Популярность	Простота использования	Безопасность	Универсальность	Ср. балл
GO	3	5	5	4	4.25
JULIA	2	4	4	3	3.25
STANZA	2	3	5	3	3.25
NIM	4	4	5	4	4.25
SWIFT	4	4	5	3	4
RUST	3	2	5	2	3
HACK	3	4	5	2	3.5
SCALA	4	3	5	4	4

Делая выводы можно сказать, что наиболее комфортными в использовании будут такие языки как: GO и NIM. Это языки, на которых пользователь сможет написать программу без особо больших усилий.

Библиографический список

1. GO – официальный сайт [Электронный ресурс]. URL: <http://lesnoy.name/golang/> (дата обращения 15.04.18).
2. JULIA – официальный сайт [Электронный ресурс]. URL: <https://julialang.org/> (дата обращения 15.04.18).
3. STANZA – официальный сайт [Электронный ресурс]. URL: <https://www.apple.com/ru/swift/> (дата обращения 15.04.18).
4. NIM – официальный сайт [Электронный ресурс]. URL: <https://nim-lang.org/> (дата обращения 15.04.18).
5. SWIFT – официальный сайт [Электронный ресурс]. URL: <https://swift.org/> (дата обращения 15.04.18).
6. RUST – официальный сайт [Электронный ресурс]. URL: <https://www.rust-lang.org/> (дата обращения 15.04.18).
7. HACK – официальный сайт [Электронный ресурс]. URL: <http://hacklang.org/> (дата обращения 15.04.18).
8. SCALA – официальный сайт [Электронный ресурс]. URL: <http://www.scala-lang.org/> (дата обращения 15.04.18).