

АНАЛИЗ СВОЙСТВ И ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ СОВРЕМЕННОГО ЯЗЫКА RUBY

Черепков А.В., Перкин П.В., Зеленёва И.Я.
ДонНТУ, Кафедра КИ, cherepkov0905@mail.ru

Аннотация

Черепков А.В., Перкин П.В., Зеленёва И.Я. Анализ свойств и практическое применение современного языка RUBY. В этой работе проанализирован ряд важных вопросов, касающихся свойств, практического применения и перспектив современного языка программирования Ruby. В статье рассмотрены преимущества, области применения, сравнительный анализ и основные характеристики данного языка. Среди изученных вопросов одними из ключевых являются показатели производительности Ruby и затраты ресурсов, что может рассматриваться в качестве основных параметров при выборе языка программирования. Таким образом, авторы раскрыли основные вопросы, интересующие как начинающих программистов, так и профессионалов, сталкивающихся с проблемой выбора ресурсов.

В наше время существует очень много языков программирования. Каждый язык имеет свои преимущества и недостатки, но в то же время, каждый из них создается для того, чтобы облегчить труд программиста, увеличить скорость обучения людей данному языку и, конечно же, увеличить скорость решения тех или иных задач с использованием данного языка. Одним из таких, относительно молодых языков, является Ruby. Он появился в 1995 году. Для сравнения - PHP был создан в 1995 году, Perl – 1987, Python – 1990.

Преимущества и области применения языка Ruby

Достаточно молодой современный язык программирования Ruby – это динамический, рефлексивный, интерпретируемый высокоуровневый язык для быстрого и удобного программирования [1]. Или, если сказать простыми словами, этот язык дает нам возможность определять типы данных и осуществлять компиляцию непосредственно во время выполнения. Это помогает ускорить процесс разработки приложений. Также в этом языке мы имеем возможность дополнять и изменять код программы во время выполнения. Это свойство и называется свойством рефлексивности языка. Ruby также обладает, так называемым, «сборщиком мусора», что является очень полезным свойством динамического языка. «Сборка мусора» – это технология автоматического управления памятью, которая самостоятельно ищет в динамической памяти уже не нужные объекты и удаляет их. Ruby является кроссплатформенным языком, и его программы могут выполняться в любой операционной системе.

Что касается областей применения языка Ruby, то, во-первых, можно сказать, что этот язык может быть первым языком для начинающих программистов, так как разработчики старались сделать его максимально интуитивно понятным. Это видно на следующем примере, который иллюстрирует разницу между фрагментами программы языка Java и Ruby, выполняющими одно и то же действие (рис. 1).

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println("Здравствуй, мир!");  
    }  
}
```

Рисунок 1 – Пример кода программы на языке Java

Аналогичная программа на Ruby, показывающая краткость кода программы (рис.2).

```
puts "Здравствуй, мир!"
```

Рисунок 2 – Пример кода программы на языке Ruby

Для понимания такой программы особых профессиональных знаний в программировании не нужно, поэтому язык можно назвать дружелюбным пользователю.

Язык Ruby очень предсказуем и даже новичок может быстро разобраться в коде. В одном из Московских вузов (МГИУ) Ruby даже используется для ознакомления с программированием первокурсников [2].

Сравнительный анализ языка Ruby и других популярных языков программирования.

Если сравнивать Ruby с другими языками, то можно увидеть, что размер его кода для решения одних и тех же задач значительно меньше, но это не мешает данному языку сохранять высокую функциональность. Приведем таблицу сравнения различных возможностей для нескольких популярных языков программирования [4].

Перед этим введем следующие обозначения для создания сравнительной характеристики (табл. 1).

Таблица 1 – Условные обозначения, используемые для сравнительной характеристики

+	Возможность присутствует
–	Возможность отсутствует
+/-	Возможность поддерживается не полностью
-/+	Возможность поддерживается очень ограничено
?	Нет данных

Из следующей таблицы мы увидим, что язык Ruby подобен по своим возможностям PHP (табл. 2). Ruby, в отличие от языков C, C++, ADA, имеет возможность оперировать собственным кодом, как данными. Рассматриваемый язык является декларативным, так как описывает не столько шаги решения задачи, сколько саму постановку задачи, а решение уже должен определять интерпретатор. Противоположный тип языков - императивный. Но стоит отметить, что это разделение условное, так как в принципе на любом языке можно создать определенные процедуры решения, которые потом будут описывать задачи, обрабатываемые этими процедурами. Способность Ruby к распараллеливанию программы ограничена, но всё-таки она есть, что дает преимущество над PHP, Perl, Delphi. Кроме того, Ruby имеет возможность вернуть кортежи (кортеж – это неименованный тип данных, содержащий несколько безымянных полей произвольного типа) из метода (функции).

Таблица 2 – Сравнительная характеристика функциональных возможностей языков программирования

Возможность	Язык									
	Ada	C	C++	C#	Prolog	Java	Perl	PHP	Delphi	Ruby
Рефлексивный	-	-	-	-/+	+	-/+	+/-	+	-/+	+
Декларативный	-	-	-	-/+	+	-	-/+	+	-	+
Распределенный	+	+/-	+/-	-/+	+	+	-	-	-	-/+
Ручное управление памятью	+	+	+	+	?	-	-	-	+	-
Сборка мусора	-/+	-	-	+	?	+	+	+	-	+
Кортежи	-	-	+/-	+/-	?	-	+	-	-	+
Целые числа произвольной длины	-	-	-	+	?	+	+	?	-	+
Статическая (динамическая) типизация	+(-)	+(-)	+(-)	+(-/+)	+(-)	+(-)	+/-(+)	-(+)	+(-/+)	-(+)

Языку Ruby сложно конкурировать с компилируемыми языками в производительности (рис. 3). Это наглядно показано на следующей диаграмме (рис.3). Здесь изображены результаты выполнения четырех тестов. Первые два теста - на выполнение циклов. Тест № 3 — заполнение большого массива. Массив в 100 млн. переменных типа int PHP не удалось заполнить, поэтому ограничились 20 млн. Тест № 4 — рекурсивное вычисление факториала.

test	№ 1	№ 2	№ 3	№ 4	
C++	234	219	234	1203	
Java	78	51	484	1469	
PHP	15460	7287	12490	127329	
Perl	16726	7417	15526	126943	
Ruby 1.9	28125	10093	19297	81937	
Ruby 1.8	25922	33782	33922	222157	
test	№ 1	№ 2	№ 3	№ 4	
C++	1	1	1	1	1,00
Java	0,333333	0,232877	2,068376	1,221114	0,96
PHP	66,06838	33,27397	53,37607	105,8429	64,64
Perl	71,47863	33,86758	66,35043	105,522	69,30
Ruby 1.9	120,1923	46,08676	82,46581	68,11056	79,21
Ruby 1.8	110,7778	154,2557	144,9658	184,6692	148,67

Рисунок 3 – Время выполнения тестов

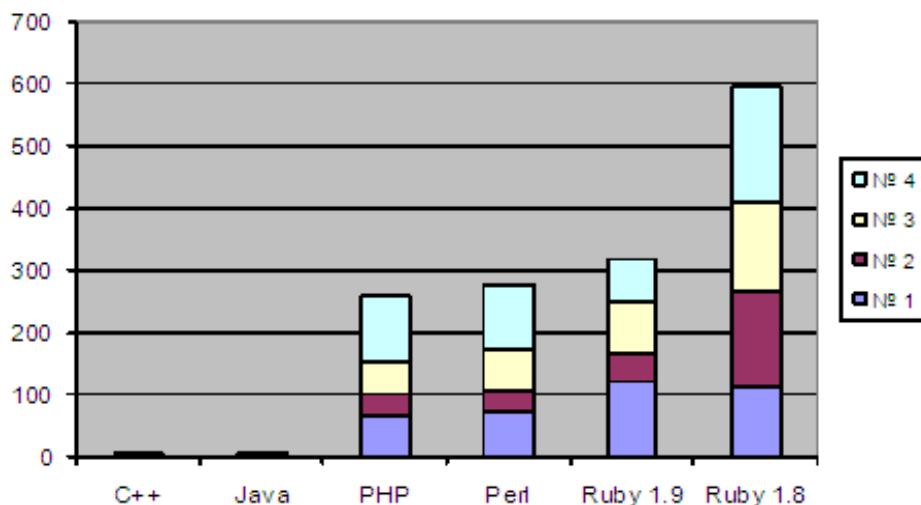


Рисунок 4 – Диаграмма результатов тестов

Конечно же, следует помнить, что мы сравнили компилируемые языки и интерпретируемые, и именно поэтому такая разница в результатах.

Так как Ruby имеет свой «сборщик мусора», эффективность работы с памятью становится на порядок выше, чем для других языков. Здесь необходимо сделать несколько пояснений к экспериментам. Большинство интерпретаторов сразу занимали определенный объем памяти, и в течение работы этот показатель уже не менялся. Однако запросы PHP во время работы линейно возрастали, а Ruby использовал и освобождал память волнообразно (табл. 3). В таблицу вошли максимальные значения проведенных тестов.

Таблица 3 – Загрузка памяти во время прохождения тестов

Язык	Память, мегабайт
Groovy 1.7.4	27
PHP 5.2.14	37
Python 2.6.5	3.8
Python 2.6.5 + Psyc0 2.0	4
Ruby 1.8.7	2
Ruby 1.9.2	1.8

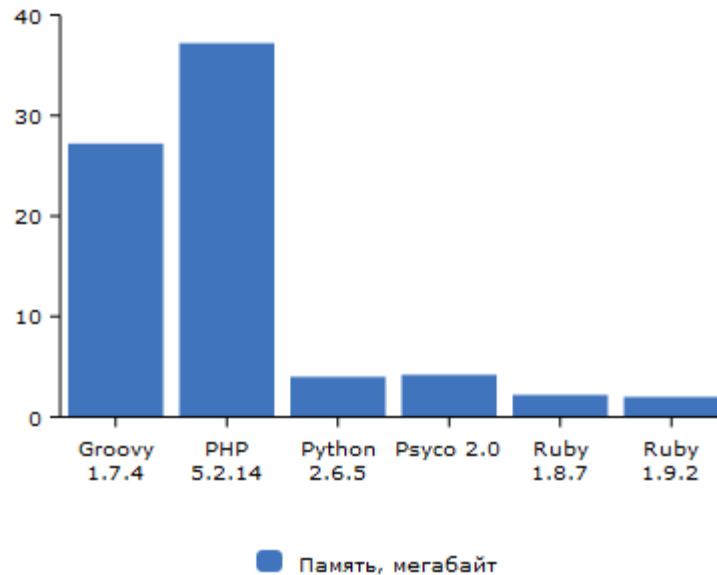


Рисунок 5 – Сравнительная диаграмма работы с памятью во время тестирования

Заключение

В заключение можно сказать, что данный язык программирования Ruby может уступать некоторым мощным языкам по производительности, и, возможно, особо мощные проекты, используя только один язык Ruby, создавать не целесообразно. Но Ruby имеет существенные плюсы, которые заключаются в том, что этот язык дает возможность быстро создавать довольно мощные приложения, в том числе и Web-приложения, с помощью инструментария Ruby on Rails, написанного также на Ruby, а также способствует быстрому обучению этому интуитивно понятному языку. Ruby имеет очень простой синтаксис. Таким образом, данный язык акцентирует внимание на понимании именно приемов программирования. На основе проведенного анализа можно сформулировать следующие рекомендации: язык Ruby целесообразно применять для обучения студентов младших курсов основам программирования в целом и объектно-ориентированному программированию в частности; кроме того, язык Ruby может эффективно и широко применяться для разработки разнообразных Web-приложений.

Список литературы

1. Программирование на языке Ruby. – М. : ДМК Пресс, 2007. – 688с.: ил.
2. Ruby лучший друг программиста // Режим доступа: <http://ruby.inuse.ru/article/yazyk-programmirovaniya-ruby>
3. Using JRuby: Bringing Ruby to Java, Pragmatic Bookshelf, Charles O Nutter. – 387с.
4. Программирование на Ruby. Учебное пособие по языку Ruby, Роганова Н. А., Тузов А. С. – 75с