

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ДЛЯ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЙ

Сапего Ю.С.

В статье проводится сравнительный анализ наиболее распространенных языков объектно-ориентированного программирования. Рассмотрены их достоинства и недостатки, сферы использования, синтаксис, семантика и некоторые возможности языков.

Ruby и PHP

Ruby является полностью динамическим и интерпретируемым высокоуровневым языком программирования, который создавался в первую очередь для быстрого и удобного объектно-ориентированного программирования. Язык обладает независимой от операционной системы реализацией многопоточности, строгой динамической типизацией, сборщиком мусора и многими другими возможностями. Ruby близок по особенностям синтаксиса к языкам Perl и Eiffel, по объектно-ориентированному подходу — к Smalltalk. Также некоторые черты языка взяты из Python, PHP, Lisp.

Язык PHP ещё в 1997г. разрабатывался как основа для World Wide Web (WWW), т.е. как «идеальный язык» для малых динамических web-страниц. С тех пор он развился на столько, что стал одним из самых популярных языков программирования для web-приложений. PHP почти повсеместно поддерживается всеми web-узлами (web-хостингами), и часто рассматривается как язык программирования «по умолчанию» для WWW. Ruby же является совершенно новым языком для этой сферы, и широко стал поддерживаться только с появлением Ruby-on-Rails в 2004г. Из-за этого не все web-хостинговые компании поддерживают Ruby на своих серверах.

Хотя сейчас PHP обладает объектно-ориентированными (ОО) возможностями, так же и Ruby (правда ООП в PHP в полной мере стало поддерживаться только в 5-ой версии), но ОО возможности не всегда могут применяться, в отличие от Ruby, в котором ООП является обязательным.

Тем не менее, Ruby и PHP обладают многими похожими возможностями:

1. Ruby и PHP являются динамически типизированными языками программирования, но в PHP имеется ряд требований к объявлению переменных: во-первых, имя любой переменной должно начинаться с префикса «\$», во-вторых, если переменная является массивом, то это явно необходимо обозначить.

Например, `$var = 1; $arr = array();`

2. В Ruby и в PHP имеются классы, доступ к которым обеспечивается за счёт явного объявления (`public`, `protected` и `private`)

3. В этих языках возможно использовать строку интерполяции, но только с некоторым отличием в синтаксисе:

в PHP: "\$foo is a \$bar",

в Ruby: "#{foo} is a #{bar}" (в Ruby это не работает, если строку поместить в одинарные кавычки)

4. В Ruby, как и в PHP, есть обработка исключений

5. Ruby имеет в своём арсенале довольно большую стандартную библиотеку, как и PHP и т.д.

Среди различий между Ruby и PHP можно выделить следующие:

1. Ruby является строго типизированным, т.е. чтобы преобразовать тип переменной из целого числа в строку и наоборот, необходимо применять методы «to_s» и «to_i» соответственно, в отличие от PHP, где за преобразование отвечает сам язык, например,

PHP

```
$str = 'abc';
```

```
$var = 123;
```

```
echo $str." ".$var;
```

Ruby

```
str = "abc"
```

```
var = 123
```

```
print (str+" "+var.to_s)
```

2. В Ruby абсолютно всё является объектами, даже просто положительное целое число, как например 1, представляет собой экземпляр класса Fixnum, дочернего класса для класса Integer, который является дочерним классом класса Numeric, в следствие чего к числу можно применяют методы.

Например, 10.times { puts "Ruby" }

3. Имеются существенные отличия в синтаксисе, например, в конце строки в Ruby не надо ставить «;», как в PHP, или в Ruby при вызове метода не обязательно писать круглые скобки (например, можно написать в Ruby -1.abs или -1.abs())

4. В Ruby «ложное» значение принимает только «false» и «nil», всё остальное является «истинным», в том числе 0, array() и ""

5. В Ruby хэши и массивы не являются взаимозаменяемыми. В PHP хэш и массив объявляются одинаково, а иногда можно встретить утверждение, что php-массивы являются упорядоченным хэшем, т.е.,

```
$hash = array(1 => 'a', 2 => 'b', 3 => 'c');
```

В Ruby в объявлениях хэша и массива есть существенная разница, в большой мере она кроется в типе используемых скобок:

Массив в Ruby: arr = ['a', 'b', 'c']

Хэш в Ruby: arr = { 1 => 'a', 2 => 'b', 3 => 'c' }

Ruby и Perl

Perl это универсальный язык программирования, первоначально разработанный для работы с текстом, и в настоящее время используются для широкого спектра задач, включая системное администрирование, web-разработки, сетевое программирование, GUI разработки и многое другое.

Perl является одним из языков, который в большей мере повлиял на формирование языка Ruby. Например, название языка языка навеяно именно Perl, т.е. perl ~ pearl -«жемчужина» и ruby – «рубин». Также прослеживается заимствование особенностей синтаксиса и семантики, некоторых принципов, например, в основе Perl лежит принцип «TIMTOWTDI», который в полной мере прослеживается и в Ruby.

Также в Ruby имеется своя система управления пакетами, как CPAN в Perl, только в Ruby он называется RubyGems.

Можно выделить следующие основные отличия:

1. Оба языка являются динамически типизированными, но как в PHP, так и в Perl, имена переменных обязаны начинаться с определенных префиксов, например, при объявлении скаляра переменная должна начинаться с «\$» (\$ от scalar), при объявлении массива с – «@» (@ от array), например, \$var = 'b'; @arr = ('a','b','c')

В Ruby однако тоже используются префиксы в именах переменных, но они в отличии от PHP и Perl определяют не тип переменной, а её область видимости, т.е.:

var или _var— локальная переменная

@var — переменная экземпляра (член или поле объекта класса)

@@var — переменная класса

\$var — глобальная переменная.

В Ruby имеет только один тип переменной: ссылка на объект (объект может действовать как массив, хэш и т.д.)

2. Как и PHP, так и Perl конец строки необходимо обозначать «;»

3. В Ruby нет понятия «undef», есть только «nil». При это nil это объект (как и всё остальное в этом языке). И это не тоже самое, что неопределённое значение переменной в Perl, в булевом выражение возвращается значение «false»

4. Полноценная встроенная поддержка Юникода в Ruby появилась только с выходом версии 1.9

5. В Perl, как и в PHP, осуществляется автоматическое преобразование строки в число и обратно

Ruby и Python

Ruby и Python очень похожи, как по объектно-ориентированным функциям, так и по их использованию.

Хотя считается, что ООП в Ruby более «чистое», различия часто замечаются в семантике, чем в практических возможностях.

Но реальные их отличия лежат, в том, как они выполняются и в их синтаксисе. Ruby является интерпретируемым языком, а Python – компилируемым.

В последнем различии Python по времени выполнения программы лидирует, т.к. при запуске сначала компилируется файл с расширением .pyc, в котором текстовый код программы преобразуется в байт-код, а затем этот файл запускается на виртуальной машине Python (PVM) (рисунки 1)

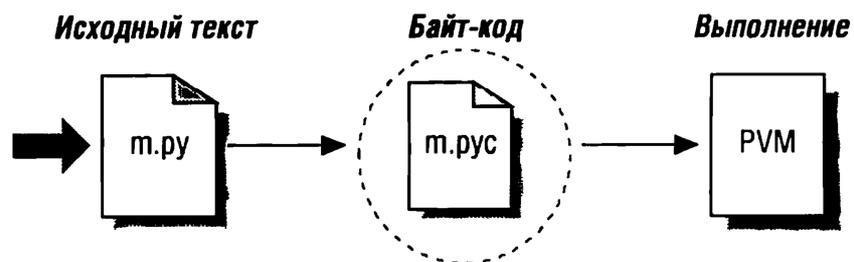


Рисунок 1 - Традиционная модель выполнения программ на Python

За счёт создания файл с байт-кодом, при повторном запуске программы, время выполнения уменьшается.

Как Ruby, так и Python, написаны полностью на языке C. Также существуют альтернативные реализации этих языков, написанных на Java, это Jython и JRuby, главной целью которых является взаимодействие с этим языком.

При быстром взгляде на программы, написанные на Ruby и Python, кажется, что их синтаксис почти одинаков: оба языка с динамической типизацией, не требуют в именах переменных префиксов (за исключением указания области видимости), не требует в конце каждого строки кода «;», многие операторы и методы по написанию почти идентичны, например, цикл for:

Ruby

```
for n in 1..10 do
  ...
end
```

Python

```
for n in range(1,10):
  ...
```

Прекрасно видно, что написание цикла for в корне отличается от того, как он реализуется в PHP или в Perl, но прослеживаются общие черты в Ruby и Python. Одним из существенных различий в синтаксисе это обозначение блоков: в Ruby используются операторы begin ...end или {...}, в Python для обозначения блока достаточно в коде сделать отступ.

Другим отличием между этими двумя языками заключается в том, что Python стремится к ясности, чёткости и по возможности сокращению лишних строк, так например, Python избегает использование всякого рода префиксах перед переменными, такие как «\$» или «@», или ограничивает одним заявлением в каждой строке.

Это жёсткие ограничения делают весь код Python одинаковым, в следствие чего две программы, написанные разными программистами, будут выглядеть почти идентично, т.к. как правило существует один способ решения задачи.

Синтаксис в Ruby тем не менее гораздо больше похожа на Perl, чем на Python. Синтаксис очень свободным, что является наследованием принципа TIMTOWTDI от Perl: возможно в одной строке писать несколько операторов и делать отступ как хочется программисту.

Другими отличительными особенностями в Ruby и Python является:

1. В Python строки являются неизменяемыми объектами
2. Ruby не поддерживает множественное наследование, но вместо него может использоваться концепция «примесей», основанная в данном языке на механизме модулей
3. В этих языках можно добавлять и изменять методы встроенных классов, в любой момент можно открывать и изменять классы, но Python позволяет модифицировать встроенные модули, а Ruby нет

Заключение

Как любой язык программирования Ruby имеет свои недостатки. Во-первых, это не высокопроизводительный язык программирования, в этой связи Python на виртуальной машине обладает огромным преимуществом.

Кроме того, если никакую, даже самую тривиальную задачу, невозможно реализовать на Ruby, не использовав объектно-ориентированные возможности языка.

На данный момент в сфере WWW, PHP является лидером, как говорилось выше, он изначально и писался, чтобы на нём создавались web-страницы. Также в этой области набирает обороты, такой язык как Python. Perl в основном применяется в сфере администрирования.

Ruby только-только начинает широко применяться, особый интерес к нему возник с выходом в 2007 г. версии 1.9, которая показала более хорошие результаты по производительности, чем предыдущие версии. С выходом новых версий, разработчики будут устранять всё больше недостатков, делая его более популярным.

Список информационных источников

- [1] Фитцджеральд М. - Изучаем Ruby: Пер. с англ. – СПб.: БХВ – Петербург, 2008 – 336с.
- [2] Лутц М. – Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011 – 1280с.
- [3] What is Ruby? (<http://ruby.about.com/od/beginningruby/a/WhatIsRuby.htm>)
- [4] How Does Ruby Compare to Python? (<http://ruby.about.com/od/beginningruby/a/vspython.htm>)
- [5] How Does Ruby Compare to PHP? (<http://ruby.about.com/od/beginningruby/a/compare.htm>)
- [6] <http://ru.wikipedia.org/wiki/Ruby>
- [7] <http://habrahabr.ru/post/70314/>
- [8] Ruby from other language (<http://www.ruby-lang.org/en/documentation/ruby-from-other-languages/>)
- [9] <http://habrahabr.ru/post/86011/>