NGÔN NGỮ LẬP TRÌNH RUBY

GVHD: Thầy Huỳnh Lê Tấn Tài

Nhóm: FRI – 06th

TÀI LIỆU THAM KHẢO

- Sách:
  Agile Web Development with Rails.
  Ruby and its world.
  Simply Rails.
  Ruby Cook Book.
  Visual Quick Start Guide.
  The Pragmatic Programmer's Guide Programming Ruby.
- Web:
  http://api.rubyonrails.org
  http://www.ruby-lang.org/en/about/

I. GIỚI THIỆU RUBY

1) Lịch sử hình thành.
2) Định nghĩa.
3) Ruby on Rails.
4) Đặc trưng.
5) Phong cách viết code.

I.1. Lịch sử hình thành

- Do Yukihiro Matz Matsumoto tạo ra từ 24/02/1993.
- 21/12/1995, bản chính thức Ruby 0.95 được công bố.
- 09/2005, phiên bản mới nhất 1.8.3 Ruby 1.9 ra đời.

I.2. Ruby là gì?

- Ngôn ngữ scripting.
- Ngôn ngữ hướng đối tượng.
- Ngôn ngữ linh hoạt.
- Ngôn ngữ cấp cao.
- Ngôn ngữ hướng về con người.
- Đề án nguồn mở.

I.2. Ruby là gì?

- Lấy ý tưởng từ:
  - Perl (1 vài cú pháp, biểu thức thông thường).
  - Smalltalk (hướng đối tượng, tính linh hoạt).
  - CLU (vòng lặp).
  - LISP (tính linh hoạt).
  - C (toán tử, prints/spintf, …).
  - …

I.3. Ruby on Rails

- Framework cho phép phát triển ứng dụng Web.
- Gồm 2 thành phần cơ bản:
  - Ngôn ngữ tích hợp trong Ruby.
  - Rails bao gồm nhiều thư viện liên kết.

I.4. Đặc trưng

- Nhìn mọi thứ là đối tượng.
- Mã ngắn gọn.
- Không cần chương trình dịch và dễ thay đổi.
- Tính thừa kế có mục đích.

I.5. Phong cách viết code

- Dùng 2 khoảng trắng thụt đầu dòng.
- Tên lớp viết hoa ký tự đầu.
- Trình soạn thảo có tối đa 80 cột/1 dòng.
- Dùng dòng trắng ngăn cách các khối mã.

II. Cú pháp

1. Quy ước.
2. Toán tử.
3. Biến.
4. Cấu trúc điều khiển.
5. Vòng lặp.

II.1. Quy ước

- Quy ước chung:
  - Dùng tiếng Anh để đặt tên.
  - Không viết tắt.
  - Biến vòng lặp đặt theo thứ tự i,j,k
  - Biến dùng trong phạm vi hẹp có thể viết tắt từ tên lớp.
    VD: eo=ExampleObject

II.1. Quy ước

- Tên lớp, tên module:
  - Viết hoa chữ đầu mỗi từ, không dùng dấu gạch dưới.
  - Từ viết tắt thì ghi toàn bộ là chữ hoa.
- Tên phương thức:
  - Trả về giá trị đúng/sai phải kết thúc bằng "?".
  - Không dùng "is_"
  - Tên làm thay đổi nội bộ kết thúc bằng "!"

II.1. Quy ước (tt)

- Tên hằng:
  - Viết hoa toàn bộ, các từ nối nhau bằng dấu gạch dưới.
- Chú thích:

  1 dòng: #

  1 đoạn:

  =begin

  [chú thích]

  =end

II.1. Quy ước

- Phương thức:
  - Phương thức của lớp: dùng self.
  - Gọi phương thức: phải có ngoặc nếu phương thức có tham số.
  - Dùng do…end nếu nhiều dòng.
  - Dùng {…} nếu có 1 dòng.
  - Không dùng return nếu không cần thiết, nếu dùng thì không dùng (…) bao quanh giá trị trả về.

II.2. Toán tử

Table 21.4. Ruby operators (high to low precedence)

Method	Operator	Description
✓	[] []=	Element reference, element set
✓	**	Exponentiation
✓	! ~ + −	Not, complement, unary plus and minus (method names for the last two are +@ and -@)
✓	* / %	Multiply, divide, and modulo
✓	+ −	Plus and minus
✓	>> <<	Right and left shift
✓	&	"And" (bitwise for integers)
✓	^ |	Exclusive "or" and regular "or" (bitwise for integers)
✓	<= < > >=	Comparison operators
✓	<=> == === != =~ !~	Equality and pattern match operators (!= and !~ may not be defined as methods)
	&&	Logical "and"
	||	Logical "or"
	.. ...	Range (inclusive and exclusive)
	? :	Ternary if-then-else
	= %= /= −= += |= &= >>= <<= *= &&= ||= **=	Assignment
	defined?	Check if symbol defined
	not	Logical negation
	or and	Logical composition
	if unless while until	Expression modifiers
	begin/end	Block expression

II.2. Toán tử

- Dùng !, && thay cho not, and, or.
- Chuỗi ký tự: dùng ‘’ khi không có nội suy, “” khi có nội suy.
- Không dùng “here document” mà dùng &, %q, %Q, thường dùng {} hay ().

II.3. Biến

- Định nghĩa hằng là định nghĩa chuỗi.
- Nhập biến: bắt đầu bằng STDOUT.fush và dùng gets.chome.
- Xuất: puts.
- Tạo tập hợp: <tên>=(giá trị)
  VD: h=(1..10)

II.4. Cấu trúc điều kiện

```
if [not] <điều kiện>
    <hàm thực thi>
else
    if [not] <điều kiện
        <hàm thực thi>
    end
end
```

```
if [not] <điều kiện>
    <hàm thực thi>
elsif [not] <điều kiện>
    <hàm thực thi>
end
```

- Dùng unless x thay cho !x.
- Nếu có else, không dùng unless.

II.5. Vòng lặp

- While:
  while <điều kiện>
  <hàm thực thi>
  end
- Until:
  until <điều kiện>
  <hàm thực thi>
  end
- for:
  for <tên biến> in <tập hợp>
  <hàm thực thi>
  end

II.5. Vòng lặp

- Lượt bỏ do khi dùng while/until, thay while !x bằng until x.
- Dùng loop cho vòng lặp vô hạn.

III. SO SÁNH

1. Ruby vs C/C++
2. Ruby on Rails vs Java

III.1. Ruby vs C/C++

- Code của Ruby không cần khai báo biến, biến trong Ruby sẽ được tự động nhận dạng về kiểu.
- Code của Ruby ngắn gọn và dễ hiểu.
- Dễ debug lỗi.

III.2. Ruby on Rails vs Java

RUBY ON RAILS

- Có sẵn WebBrick server dùng cho phát triển
- Tổng hợp những ưu điểm của STRUTS, Spring, Hibernate, JAspect, JUnit… và được xây dựng từ đầu để hỗ trợ tất cả trong 1.

JAVA

- Phải cài Tomcat/Jboss
- Có tất cả những cái đó nhưng bạn sẽ phải làm rất nhiều việc để STRUTS, Spring, Hibernate, JAspect, JUnit… làm việc hài hòa với nhau.

III.2. Ruby on Rails vs Java

RUBY ON RAILS

- Rails đã tiến rất xa trong kiến trúc MVC: model, controller được tách rời.
- Có tốc độ nhanh trung bình gấp 5 lần Java

JAVA

- Cần phải tích hợp một framework khác.
- Struts chậm hơn Rails