

5分でわかる

JRuby

高井 直人

2007年8月4日 (土)

Lightweight Language Spirit

於 日本教育会館一ツ橋ホール

別名

JRubyの説明を**VM魂**とい
うパネルディスカッション
の**テーマ**に則して解説しま
すから当然**マニアック**なイ
ントロダクションになって
しまうのです

あ、私は

高井 直人です

takai@recompile.net

<http://recompile.net>

さて

JRubyって何だ
ろう？

それは

Javaで書かれた

Rubyです

Rubyバージョン

1.8.5と互換性が

あります

<http://www.jruby.org>

現在のメイン開発者は

id:enebo

id:headius

id:ola

id:nicksieger



ThoughtWorks®

Ruby会議2007

でバージョン1.0

が発表されました

**メンテナンスリリースの
バージョン1.0.1と次期
バージョン1.1が予定さ
れています**

おことわり

今回の話の内容は、トランクの最新版（2007年8月3日）に基づいています。

というわけで

今回は、この後の
ディスカッション
を念頭に少々説明
をさせていただきます

JRubyは、**Java**
との統合がされて
いるのが最大の特
徴です

require "java"

**Javaを呼び出す
こともできます**

例えば、

Javaのクラスを
インスタンス化し
たり

そのインスタンス

のメソッドを

Rubyから呼び出

してみたり

Javaのクラスを
Rubyで継承して
みたり

Javaのインター
フェイスを実装し
てみたり

Javaのオブジェ
クトに特異メソッ
ドを追加してみた
り

色々できます

それとは逆に、
JavaにRubyを
埋め込んだりして
も楽しいです

さて、そんな
JRubyですが

VMということ
いわせていただけ
ればJRubyこそ

最強

なぜならば！

バーチャルマシン
の数がちがうから
なのです



JVM

org.jruby.ast.executable.
YARVMachine

YARV

org.jruby.ast.executable.
RubiniusMachine

Rubinius

3つのバーチャル
マシンを備えるの
はJRubyだけ！

今なら **インタープ**
リタ もつけちゃう

org.jruby.evaluator
EvaluationState

評価器

スクリプトの断片
の評価や、コンパ
イルできないとき
に利用します

基本的にはJava

バイトコードにコ

ンパイルしようと

します

それができないと
きは、インタープ
リタ実行です

さて、ここで**簡単**
に仕組みをみま
しょう

```
module WF
  class Transition
    attr_reader :node
    attr_reader :action

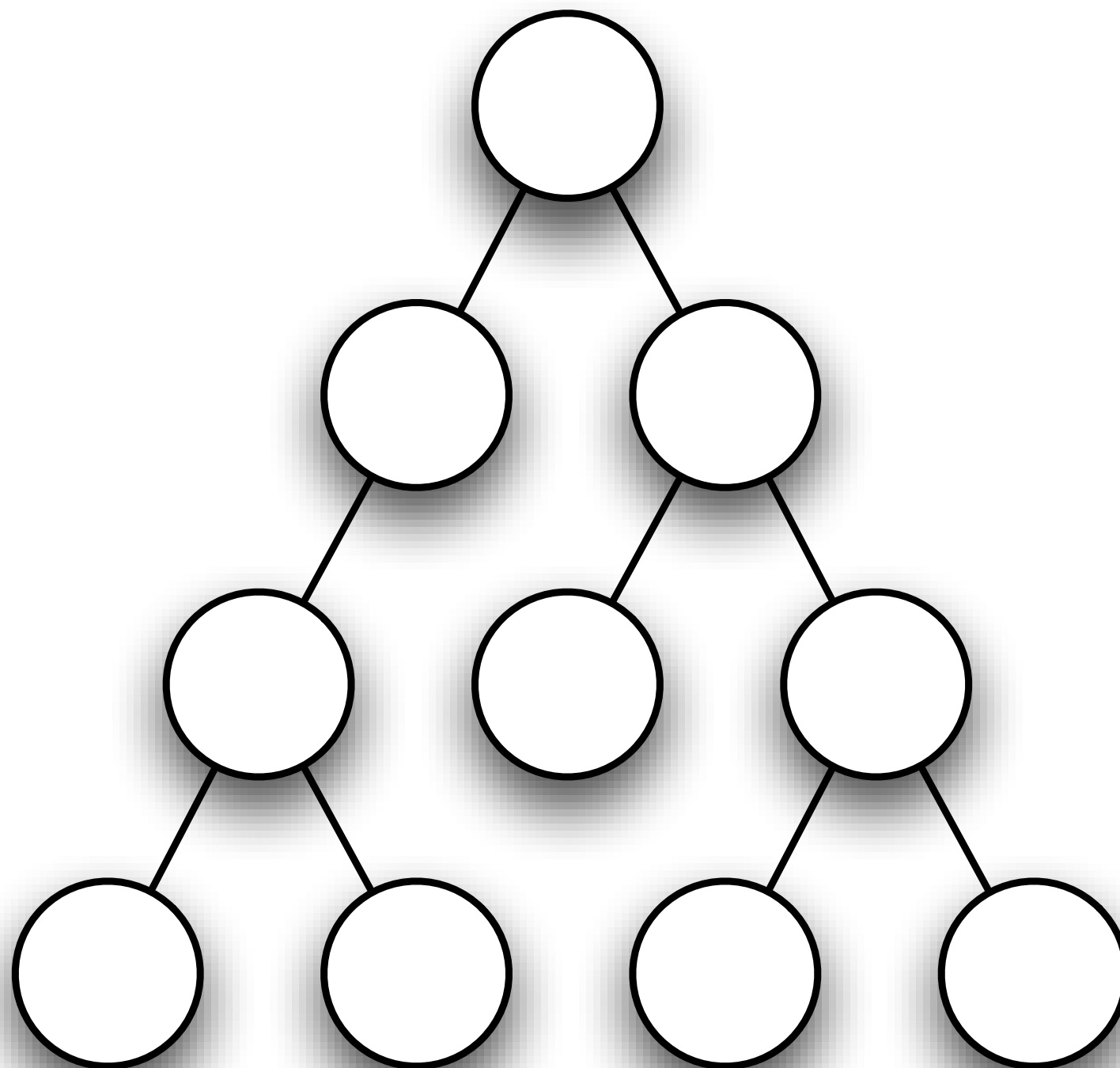
    def initialize(node, action = nil)
      @node = node
      @action = action
    end
  end
end
```

ソースコード

org.jruby.lexer.yacc.
RubyYaccLexer

org.jruby.parser
DefaultRubyParser

字句解析・構文解析



抽象構文本

ここまでは共通

ここから、Java
バイトコードへコ
ンパイルしたり

org.jruby.compiler.impl
StandardASMCompiler

コード生成

00000000	ca	fe	ba	be	00
00000020	07	00	01	01	00
00000040	4f	62	6a	65	63
00000060	6a	72	75	62	79

Javaバイトコード



JVM

YARV中間コード

へとコンパイルし

てみたり

org.jruby.compiler.yarv
StandardYARVCompiler

コード生成

0000000	52	42	43	4d	04
0000020	73	74	72	75	63
0000040	61	74	61	46	6f
0000060	22	0b	3c	6d	61

YARV中間コード

org.jruby.ast.executable.
YARVMachine

YARV

Rubiniusのコン

パイラはまだです

がバーチャルマシ

ンがあります

00000000	52	42	49	58	00
00000020	00	00	00	01	69
00000040	49	62	00	00	00
00000060	00	4e	00	00	00

Rubinius中間コード

org.jruby.ast.executable.
RubiniusMachine

Rubinius

Javaバイトコード
へのコンパイル
には**ASM**を利用
しています

汎用技術を流用し
ているので、最適
化はまだまだ改善
の余地あり

今は中途半端だけ
れども**将来**とかは
こんな感じ

```
package WF;

class Transition {
  private Node node;
  private Action action;

  public Transition(Node node,
                    Action action) {
    this.node = node;
    this.action = action;
  }
}
```

Javaソースコード

```
module WF
class Transition
  attr_reader :node
  attr_reader :action

  def initialize(node, action = nil)
    @node = node
    @action = action
  end
end
end
```

Rubyソースコード

Javac

JRuby

```
00000000  ca fe ba be 00
00000020  07 00 01 01 00
00000040  4f 62 6a 65 63
00000060  6a 72 75 62 79
```

Javaバイトコード



JVM

**JVMでの一級市民
の言語として頑張
る心構えです**

角谷宣言 「もう
Javaやのうてえ
えがな」 から4年

ついにここまで来
ました

Rubyに勝てるのは
Rubyだけ！

今後とも**JRuby**
をよろしくおねが
いします。

ありがとうございました

8月24日発売

WEB+DB PRESSにて

JRuby入門記事が掲載！

