5分で理解するRubyのリフレクション

大林一平

KMC/Dept. Math., Kyoto U.

2010/11/6

自己紹介

- KMC
- 京都大学
  - 数学の研究者 (力学系)
- Ruby/SDL, RRSE
- 最近はるりまの文章書き
- 趣味:ボードゲーム,rouge like,STG, プログラミング言語評論, 数学とコンピュータ

リフレクション

- 実行時にプログラム自体の情報を問い合わせる
- クラス，メソッド，変数
- メタ機能

5 分で

- Ruby のリフレクションの全貌を理解

5分で

- Rubyのリフレクションの全貌を理解
- するのは不可能なので1トピックだけ

5分で

- Rubyのリフレクションの全貌を理解
- するのは不可能なので1トピックだけ
- Module#method_added
  - 厳密にはリフレクションとは言わないかも (問い合わせ以上のことをしている)
  - メタ機能ではある
- とりあえず 1.9.2-p0 準拠

Module#method_added

- クラス/モジュールにメソッドが追加されたときに呼びだされる
  - def . . . end
  - Module#define_method

例題

```
module K
  def method_added(name)
    puts "#{name} is added"
    super(name)
  end
end

class A
  extend K
  def f; end  # => f is added
  def g; end  # => g is added
end
```

実用

```
require 'obsolete'

class A
  extend Obsolete
  def f; p "f"; end

  obsolete
  def g; p "g"; end
  def h; p "h"; end
end
# g と h は obsolete になる
# example.rb を実行
```

準備

```
module Obsolete
  def self.extended(obj)
    obj.__obsolete_initialize
  end

  def __obsolete_initialize
    @__obsolete_on = false
    @__obsolete_in_method_added = false
  end

  def obsolete
    @__obsolete_on = true
  end
```

メイン

```
def method_added(method_name)
  if @__obsolete_on
    unbound_method =
      instance_method(method_name)
    msg = "#{self.class}\##{method_name}" +
          "is obsolete"
    define_method(method_name) do
      |*arg, &blk|
      warn msg
      unbound_method.bind(self).
        call(*arg,&blk)
    end
  end
```

ん

- def . . . end
- method_added が呼びだされる
- define_method が呼びだされる

ん

- def . . . end
- method_added が呼びだされる
- define_method が呼びだされる
- method_added が呼びだされる
- . . .

メイン

```
  def method_added(method_name)
    if @__obsolete_on &&
       !@__obsolete_in_method_added
    @__obsolete_in_method_added = true
    # define new method
    @__obsolete_in_method_added = false
    super(method_name)
  end
end
```

まとめ

- define_method と method_added を組み合わせると楽しい

まとめ

- define_method と method_added を組み合わせると楽しい
- いろいろ相互作用するのであんまり過激なことをすべきではないんだけど

まとめ

- define_method と method_added を組み合わせると楽しい
- いろいろ相互作用するのであんまり過激なことをすべきではないんだけど
- ご静聴ありがとうございました