

Ruby 初級者向けレッスン in KOF 2005 (解答)

かずひこ@ネットワーク応用通信研究所

2005 年 10 月 29 日

解答

- MyEnumerable モジュールのコード (my_enumerable.rb)

```
module MyEnumerable
  def collect
    ret = []
    self.each do |elem|
      ret << yield(elem)
    end
    return ret
  end

  def select
    ret = []
    self.each do |elem|
      ret << elem if yield(elem)
    end
    return ret
  end

  def reject
    ret = []
    self.each do |elem|
      ret << elem unless yield(elem)
    end
    return ret
  end
end
```

```
def detect(ifnone = nil)
  self.each do |elem|
    return elem if yield(elem)
  end
  ifnone.call unless ifnone.nil?
end
```

- MyEnumerable モジュールのテストコード (test_my_enumerable.rb)

```
require 'my_enumerable'
require 'test/unit'

class TestMyEnumerable < Test::Unit::TestCase
  def setup
    @array = [1, 2, 3, 4, 5]
    @my_array = [1, 2, 3, 4, 5]
    @my_array.extend(MyEnumerable)
  end

  def test_collect
    array_collect = @array.collect do |i|
      i * 2
    end
    my_array_collect = @my_array.collect do |i|
      i * 2
    end
    assert_equal(array_collect, my_array_collect)
  end

  def test_select
    array_select = @array.select do |i|
      i % 2 == 0
    end
    my_array_select = @my_array.select do |i|
      i % 2 == 0
    end
    assert_equal(array_select, my_array_select)
  end
end
```

```

def test_reject
  array_reject = @array.reject do |i|
    i % 2 == 0
  end
  my_array_reject = @my_array.reject do |i|
    i % 2 == 0
  end
  assert_equal(array_reject, my_array_reject)
end

def test_detect
  array_detect = @array.detect do |i|
    i % 2 == 0
  end
  my_array_detect = @my_array.detect do |i|
    i % 2 == 0
  end
  assert_equal(array_detect, my_array_detect)

  array_detect = @array.detect do |i|
    i > 5
  end
  my_array_detect = @my_array.detect do |i|
    i > 5
  end
  assert_equal(array_detect, my_array_detect)

  proc = Proc.new do
    'not found'
  end
  array_detect = @array.detect(proc) do |i|
    i > 5
  end
  my_array_detect = @my_array.detect(proc) do |i|
    i > 5
  end
  assert_equal(array_detect, my_array_detect)
end
end

```

補足

Enumerable#detect の正確な仕様は以下のとおりで、この解答もその仕様にあわせて実装しています。

Enumerable#detect(ifnone = nil) {|item| ... }

要素に対してブロックを評価した値が真になった最初の要素を返します。真になる要素がひとつも見つからなかったときは、ifnone が指定されていれば ifnone を評価した結果を返し、ifnone が指定されていなければ nil を返します。

テストコードの setup メソッドでは、@my_array の各イテレータメソッドを MyEnumerable モジュールのメソッドで上書きするために、extend メソッドを使っています。

Object#extend(module ...)

引数で指定したモジュールで定義されているメソッドが self の特異メソッドとして追加されます。あるオブジェクトだけにモジュールの機能を追加したいときに使用します。

```

module Foo
  def a
    'ok'
  end
end

obj = Object.new
obj.extend Foo
p obj.a          #=> "ok"

```

extend の機能は、「特異クラスに対する include」と言い替えることもできます。