

Ruby - Bug #15135

Refining #initialize method not working

09/18/2018 09:37 AM - cbruckmayer (Christian Bruckmayer)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 2.5.0p0 (2017-12-25 revision 61468)	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

See the following example:

```
class C
  def initialize(arg)
    puts arg
  end
end

module M
  refine C do
    def initialize(arg1, arg2)
      puts arg1, arg2
    end
  end
end

using M
C.new("foo") # works
C.new("foo", "bar") # wrong number of arguments (given 2, expected 1) (ArgumentError)
```

This code will throw an refinement.rb:2:in `initialize': wrong number of arguments (given 2, expected 1) (ArgumentError) exception. I would expect that it is possible to refine the initialize method to have similar behavior as monkey patching. If this is intended behavior I would at least expect this to be documented here: https://docs.ruby-lang.org/en/2.5.0/syntax/refinements_rdoc.html

History

#1 - 09/18/2018 12:49 PM - matz (Yukihiro Matsumoto)

- Status changed from Open to Closed

From the reference (https://docs.ruby-lang.org/en/2.5.0/syntax/refinements_rdoc.html):

Refinements are lexical in scope.
When control is transferred outside the scope, the refinement is deactivated. This means that if you require or load a file or call a method that is defined outside the current scope the refinement will be deactivated

So initialize which is called outside of the scope will not be overridden.

Matz.