Ruby master - Bug #7261
Symbol#to_proc to retrieve a method from current refinement
11/02/2012 05:52 AM - matz (Yukihiro Matsumoto)

Status: Closed
Priority: Normal
Assignee: shugo (Shugo Maeda)
Target version: 2.0.0
ruby -v: ruby 2.0.0dev (2012-11-01 trunk 37415) [i486-linux]

Description
Currently Symbol#to_proc retrieves the global definition of the method, since refinement is a lexical scope. But if possible, it is better that Symbol#to_proc can retrieve a method from current refinement.

module Silly
refine Integer do
def bar; p :bar; end
end
end

module Stupid
using Silly
1.bar #=> :bar
[1,2].each(&.bar) #=> undefined method `bar'
end

What do you think?
It should be done before 2.0.0 if we merge, to avoid future compatibility issues.
Matz.

Associated revisions
Revision bb504213 - 11/02/2012 05:48 AM - shugo (Shugo Maeda)

- string.c (sym_to_proc, sym_call): A Proc created by Symbol#to_proc should close over the current refinements. [ruby-dev:46345] [Bug #7261]
- vm_eval.c (rb_call0, rb_search_method_entry, rb_funcall_passing_block_with_refinements): add a new argument 'refinements' for the above changes.
- test/ruby/test_refinement.rb: related test.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@37418 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 37418 - 11/02/2012 05:48 AM - shugo (Shugo Maeda)

- string.c (sym_to_proc, sym_call): A Proc created by Symbol#to_proc should close over the current refinements. [ruby-dev:46345] [Bug #7261]
- vm_eval.c (rb_call0, rb_search_method_entry, rb_funcall_passing_block_with_refinements): add a new argument 'refinements' for the above changes.
- test/ruby/test_refinement.rb: related test.
string.c (sym_to_proc, sym_call): A Proc created by Symbol#to_proc should close over the current refinements. [ruby-dev:46345] [Bug #7261]

vm_eval.c (rb_call0, rb_search_method_entry, rb_funcall_passing_block_with_refinements): add a new argument 'refinements' for the above changes.

test/ruby/test_refinement.rb: related test.

string.c (sym_to_proc, sym_call): A Proc created by Symbol#to_proc should close over the current refinements. [ruby-dev:46345] [Bug #7261]

vm_eval.c (rb_call0, rb_search_method_entry, rb_funcall_passing_block_with_refinements): add a new argument 'refinements' for the above changes.

test/ruby/test_refinement.rb: related test.

string.c (sym_to_proc, sym_call): A Proc created by Symbol#to_proc should close over the current refinements. [ruby-dev:46345] [Bug #7261]

vm_eval.c (rb_call0, rb_search_method_entry, rb_funcall_passing_block_with_refinements): add a new argument 'refinements' for the above changes.

test/ruby/test_refinement.rb: related test.

string.c (sym_to_proc, sym_call): A Proc created by Symbol#to_proc should close over the current refinements. [ruby-dev:46345] [Bug #7261]

vm_eval.c (rb_call0, rb_search_method_entry, rb_funcall_passing_block_with_refinements): add a new argument 'refinements' for the above changes.

test/ruby/test_refinement.rb: related test.
string.c (sym_to_proc, sym_call): A Proc created by Symbol#to_proc should close over the current refinements.
[ruby-dev:46345] [Bug #7261]

vm_eval.c (rb_call0, rb_search_method_entry, 
rb_funcall_passing_block_with_refinements): add a new argument 'refinements' for the above changes.

test/ruby/test_refinement.rb: related test.

---

History

#1 - 11/02/2012 02:48 PM - shugo (Shugo Maeda)
- Status changed from Open to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r37418. 
Yukihiro, thank you for reporting this issue. 
Your contribution to Ruby is greatly appreciated. 
May Ruby be with you.

---

string.c (sym_to_proc, sym_call): A Proc created by Symbol#to_proc should close over the current refinements.
[ruby-dev:46345] [Bug #7261]

vm_eval.c (rb_call0, rb_search_method_entry, 
rb_funcall_passing_block_with_refinements): add a new argument 'refinements' for the above changes.

test/ruby/test_refinement.rb: related test.