Ruby master - Bug #3406

attr_accessor creates a private writer method

06/08/2010 01:18 AM - vjoel (Joel VanderWerf)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>matz (Yukihiro Matsumoto)</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.0.0</td>
</tr>
<tr>
<td>ruby -v:</td>
<td>ruby 1.9.2dev (2010-05-31) [x86_64-linux]</td>
</tr>
</tbody>
</table>

Description
=begin
(Also, there is a warning that may be incorrect.)

$ cat t.rb
module M
  private
  def attr_accessor(*args); super; end
end
class C
  extend M
  attr_accessor :x
end
C.new.x = 1

$ ruby19 -v t.rb
ruby 1.9.2dev (2010-05-31) [x86_64-linux]
t.rb:3: warning: private attribute?
t.rb:11:in <main>': private method x= called for # (NoMethodError)
=end

History
#1 - 06/08/2010 04:52 AM - nobu (Nobuyoshi Nakada)

=begin
Hi,

At Tue, 8 Jun 2010 01:18:50 +0900,
Joel VanderWerf wrote in [ruby-core:30638]:

Bug #3406: attr_accessor creates a private writer method
$ ruby19 -v t.rb
ruby 1.9.2dev (2010-05-31) [x86_64-linux]
t.rb:3: warning: private attribute?
t.rb:11:in <main>': private method x= called for # (NoMethodError)

The cause is that executing cref is used, whereas class
definition cref should be used.

Doesn't seem to need the separate module.
Just a test case for the present.

diff --git a/test/ruby/test_module.rb b/test/ruby/test_module.rb
index c2084a2..8963b37 100644
--- a/test/ruby/test_module.rb
+++ b/test/ruby/test_module.rb
@@ -921,4 +921,17 @@ class TestModule < Test::Unit::TestCase
end
end
+
  def test_attr_inherited_visibility
  
  03/18/2020
c = Class.new do
  class << self
    private
    def attr_accessor(*); super; end
  end
  attr_accessor :x
end

bug3406 = '[ruby-core:30638]'
assert_nothing_raised(bug3406) {c.x = 1}
assert_equal(1, c.x, bug3406)

--
Nobu Nakada
=end

#2 - 06/28/2010 03:59 AM - mame (Yusuke Endoh)
- Target version changed from 1.9.2 to 2.0.0

=begin
I really doubt whether or not this is a bug. Very clumsy semantics is required to "fix" this. At least, any change for this issue should not be included in 1.9.2.

What we really need is a general way for users to define attr_accessor-like method, I guess. IOW, we need a way to get a visibility of current caller context and to define a method with the visibility.

--
Yusuke Endoh mame@tsg.ne.jp
=end

#3 - 10/19/2010 02:56 PM - mrkn (Kenta Murata)
- Assignee set to matz (Yukihiro Matsumoto)

=begin
=end

#4 - 10/24/2010 02:17 PM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed
- % Done changed from 0 to 100

=begin
This issue was solved with changeset r29578.
Joel, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.
= end