Inline rescue is inconsistent when rescuing NoMethodError

12/04/2016 05:20 PM - josh.cheek (Josh Cheek)

Status: Closed
Priority: Normal
Assignee: matz (Yukihiro Matsumoto)
Target version:
ruby -v:
ruby 2.4.0dev (2016-11-16 trunk 56815)
[x86_64-darwin15]
Backport:
2.1: UNKNOWN, 2.2: UNKNOWN, 2.3:

Description

class << o
  private
  attr_accessor :a
end

# GETTER can be rescued in a begin-block or inline
begin o.a
  rescue; !$!.class # => NoMethodError
end
o.a rescue !$!.class # => NoMethodError

# SETTER can be rescued in a begin-block but NOT inline!
begin o.a = 1
  rescue; !$!.class # => NoMethodError
end
o.a = 1 rescue !$!.class

# ~> NoMethodError
# ~> private method `a=' called for "some object":String
# ~> Did you mean? a
# ~> 
# ~> program.rb:18:in `<main>'

Related issues:
Related to Ruby master - Bug #12402: Inline rescue behavior inconsistent for ...
Closed

History

#1 - 12/04/2016 06:27 PM - matz (Yukihiro Matsumoto)
- Status changed from Open to Closed

o.a = 1 rescue !$!.class

is parsed as

o.a = (1 rescue !$!.class)

Try

(o.a = 1) rescue $1.class

Matz.

#2 - 12/06/2016 07:28 AM - duerst (Martin Dürst)
- Assignee set to matz (Yukihiro Matsumoto)
- Status changed from Closed to Open

matz (Yukihiro Matsumoto): Is there a good reason that o.a = 1 rescue $!.class is interpreted as o.a = (1 rescue $1.class), while it looks to me as if o.a = 1 if $!.class is (o.a = 1) if $!.class?
(Backwards compatibility might be a reasonable good reason, but I can't imagine it being useful.)

#3 - 12/07/2016 12:09 AM - shyouhei (Shyouhei Urabe)
Martin Dürst wrote:

matz (Yukihiro Matsumoto): Is there a good reason that o.a = 1 rescue $!.class is interpreted as o.a = (1 rescue $1.class), while it looks to me as if o.a = 1 if $!.class is (o.a = 1) if $!.class?

(Backwards compatibility might be a reasonable good reason, but I can't imagine it being useful.)

I guess issue #12402 is the reason behind this behaviour.

#4 - 12/07/2016 12:10 AM - shyouhei (Shyouhei Urabe)
- Related to Bug #12402: Inline rescue behavior inconsistent for method calls with arguments and assignment added

#5 - 07/25/2019 08:58 PM - jeremyevans0 (Jeremy Evans)
- Status changed from Open to Closed