# Feature #12534

**Refinements: refine modules as well**

06/29/2016 12:19 PM - chucke (Tiago Cardoso)

<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>shugo (Shugo Maeda)</td>
</tr>
<tr>
<td>Target version</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

Refinements were added as a feature to scope monkey-patches on ruby core elements. This works for elements such as String, Array (all classes), but not for modules (Enumerable, Timeout...).

This might be related to refinements not working for singleton classes (for example, I can't define a class method inside a refine block).

This code is evaluated, but the method is undefined:

```ruby
module Extension
  refine Array do
    def self.foo
      puts "bar"
    end
  end
end
```

This code breaks:

```ruby
module Extension
  refine Kernel do
    def foo
      puts "bar"
    end
  end
end
```

**Associated revisions**

Revision a463ab1f - 09/23/2016 11:46 AM - shugo (Shugo Maeda)

- eval.c (rb_mod_refine): refine modules as well. [ruby-core:76199] [Feature #12534]

Revision 56213 - 09/23/2016 11:46 AM - shugo (Shugo Maeda)

- eval.c (rb_mod_refine): refine modules as well. [ruby-core:76199] [Feature #12534]

Revision 56213 - 09/23/2016 11:46 AM - shugo (Shugo Maeda)

- eval.c (rb_mod_refine): refine modules as well. [ruby-core:76199] [Feature #12534]

Revision 56213 - 09/23/2016 11:46 AM - shugo (Shugo Maeda)

- eval.c (rb_mod_refine): refine modules as well. [ruby-core:76199] [Feature #12534]

Revision 56213 - 09/23/2016 11:46 AM - nobu (Nobuyoshi Nakada)

vm_method.c: update assertion [ci skip]

- vm_method.c (prepare_callable_method_entry): update assertion as defined_class may be TMODULE not only I_CLASS since r56213. [Feature #12534]
Revision 56380 - 10/09/2016 02:57 AM - nobu (Nobuyoshi Nakada)

vm_method.c: update assertion [ci skip]

* vm_method.c (prepare_callable_method_entry): update assertion as defined_class may be T_MODULE not only I_ICLASS since r56213. [Feature #12534]

Revision 56380 - 10/09/2016 02:57 AM - nobu (Nobuyoshi Nakada)

vm_method.c: update assertion [ci skip]

* vm_method.c (prepare_callable_method_entry): update assertion as defined_class may be T_MODULE not only I_ICLASS since r56213. [Feature #12534]

Revision 56380 - 10/09/2016 02:57 AM - nobu (Nobuyoshi Nakada)

vm_method.c: update assertion [ci skip]

* vm_method.c (prepare_callable_method_entry): update assertion as defined_class may be T_MODULE not only I_ICLASS since r56213. [Feature #12534]

Revision 56381 - 10/09/2016 09:42 AM - nobu (Nobuyoshi Nakada)

vm_insnhelper.c: update assertion [ci skip]

* vm_insnhelper.c (callable_class_p): update assertion as callable class may be T_MODULE or I_ICLASS which refines a module since r56213. [Feature #12534]

Revision 56381 - 10/09/2016 09:42 AM - nobu (Nobuyoshi Nakada)

vm_insnhelper.c: update assertion [ci skip]

* vm_insnhelper.c (callable_class_p): update assertion as callable class may be T_MODULE or I_ICLASS which refines a module since r56213. [Feature #12534]

Revision 56381 - 10/09/2016 09:42 AM - nobu (Nobuyoshi Nakada)

vm_insnhelper.c: update assertion [ci skip]

* vm_insnhelper.c (callable_class_p): update assertion as callable class may be T_MODULE or I_ICLASS which refines a module since r56213. [Feature #12534]

Revision 56381 - 10/09/2016 09:42 AM - nobu (Nobuyoshi Nakada)

vm_insnhelper.c: update assertion [ci skip]

* vm_insnhelper.c (callable_class_p): update assertion as callable class may be T_MODULE or I_ICLASS which refines a module since r56213. [Feature #12534]

Revision 56381 - 10/09/2016 09:42 AM - nobu (Nobuyoshi Nakada)

vm_insnhelper.c: update assertion [ci skip]

* vm_insnhelper.c (callable_class_p): update assertion as callable class may be T_MODULE or I_ICLASS which refines a module since r56213. [Feature #12534]

History

#1 - 06/30/2016 03:51 AM - shevegen (Robert A. Heiler)

Interesting. I was not aware of this.

It is however mentioned in the documentation:

http://ruby-doc.org/core-2.3.1/doc/syntax/refinements_rdoc.html
"Refinements only modify classes, not modules so the argument must be a class."

As to why, I don't know either. Perhaps the documentation can add some words to briefly state whether this is on purpose or not.

#2 - 07/08/2016 07:18 AM - shugo (Shugo Maeda)
- Assignee set to shugo (Shugo Maeda)
- Status changed from Open to Feedback

There is an implementation difficulty when refining modules and calling super in that refinement.

One solution is prohibiting super in a refinement for a module.
Other proposals (and patches) are welcome.

#3 - 09/07/2016 08:03 AM - matz (Yukihiro Matsumoto)
I understand the implementation difficulty. If we prohibit super in refined methods in Modules, is it possible to refine modules? If it's reasonably possible, I'd like to accept this.

Matz.

#4 - 09/23/2016 11:46 AM - shugo (Shugo Maeda)
- Status changed from Feedback to Closed

Applied in changeset r56213.

- eval.c (rb_mod_refine): refine modules as well. [ruby-core:76199] [Feature #12534]