Refinement#{extend_object,append_features,prepend_features} should be removed

10/26/2021 10:20 AM - shugo (Shugo Maeda)

Status: Closed
Priority: Normal
Assignee: shugo (Shugo Maeda)
Target version:

Description
Refinement#{extend_object,append_features,prepend_features} are not useful and should be removed.

How about to deprecate them in Ruby 3.1 and remove in Ruby 3.1?

Related issues:
Related to Ruby master - Bug #17429: Prohibit include/prepend in refinement m...

Closed

Revision 791343b5 - 01/05/2022 06:59 PM - jeremyevans (Jeremy Evans)
Remove Refinement#{extend_object,append_features,prepend_features}
Also make include, prepend, and extend raise a TypeError if one of the modules is a refinement.

Implements [Feature #18270]

History
#1 - 10/26/2021 10:20 AM - shugo (Shugo Maeda)
- Related to Bug #17429: Prohibit include/prepend in refinement modules added

#2 - 10/27/2021 04:31 PM - jeremyevans0 (Jeremy Evans)
- Backport deleted (2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN)
- Tracker changed from Bug to Feature

#3 - 11/17/2021 07:05 AM - shugo (Shugo Maeda)
These methods raise ArgumentError since #13236 was fixed, so it may not be necessary to remove them.

#4 - 11/17/2021 07:58 AM - shugo (Shugo Maeda)
append_features is undefined in the class Class, and Module#include raises TypeError before calling append_features if the given argument is a Class.

#5 - 11/18/2021 06:14 AM - shugo (Shugo Maeda)
- Assignee changed from matz (Yukihiro Matsumoto) to shugo (Shugo Maeda)

At the developers meeting on 2011-11-18, Matz accepted the following changes in Ruby 3.2:
- Undefine extend_object, append_features, prepend_features in Refinement.
- extend, include, and prepend raise a TypeError if the given argument is a Refinement as in the case the argument is a Class.

No warnings will be added in Ruby 3.1, because append_features etc. currently already raises ArgumentError.

excelsior.rb$ ruby -e 'class Foo; end; module Bar; refine Integer do append_features(Foo); end; end'
    -e:1:in `append_features': refinement module is not allowed (ArgumentError)
    from -e:1:in `block in <module:Bar>,'
    from -e:1:in `refine'
    from -e:1:in `<module:Bar>,'
    from -e:1:in `<main>,'

#6 - 11/18/2021 09:55 AM - Eregon (Benoit Daloze)
Sounds good.
Could you please add tests in ruby/spec to track those changes, under core/refinement?
The structure can be generated with `../mspec/bin/mkspec -b core -c Refinement`.

#7 - 11/19/2021 02:08 AM - shugo (Shugo Maeda)
Eregon (Benoit Daloze) wrote in #note-6:

Sounds good.

Thanks.

Could you please add tests in ruby/spec to track those changes, under core/refinement? The structure can be generated with `../mspec/bin/mkspec -b core -c Refinement`.

OK, I'll add tests after implementing these changes after the release of Ruby 3.1.

#8 - 12/27/2021 08:14 PM - jeremyevans0 (Jeremy Evans)
I've submitted a pull request to implement this: https://github.com/ruby/ruby/pull/5358

#9 - 01/05/2022 09:00 AM - shugo (Shugo Maeda)
jeremyevans0 (Jeremy Evans) wrote in #note-8:

I've submitted a pull request to implement this: https://github.com/ruby/ruby/pull/5358

Thank you. I've approved it, but it has conflicts with my commits.... Could you merge it resolving conflicts?

#10 - 01/05/2022 06:59 PM - jeremyevans (Jeremy Evans)
- Status changed from Assigned to Closed

Applied in changeset git|791343b5bb1b8bb11de4ba2800118c20e36b5a71.

Remove Refinement#{extend_object,append_features,prepend_features}
Also make include, prepend, and extend raise a TypeError if one of the modules is a refinement.
Implements [Feature #18270]