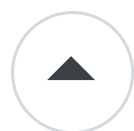


Is a Ruby module equivalent to a Java Interface?

Asked 15 years, 10 months ago Modified 12 years, 7 months ago

Viewed 10k times



21



As I understand it, an interface in Java is intended to enforce a design by laying out methods for classes implementing the interface to fill in. Is this the idea with a Ruby module also? I see that just like with Interfaces in Java, you can't instantiate a module in Ruby.



java

ruby

interface

module



Share

Improve this question

Follow

asked Feb 22, 2009 at 22:07



Bryan Locke

2,357 ● 6 ● 25 ● 30

5 Answers

Sorted by:

Highest score (default)



17



The short answer is no.

Here's the reasoning, a Java/C# interface defines the method signatures an implementing class will provide at minimum.

Additionally:



- With ruby modules there is no such contract because of the duck-typing.
- Modules are just a way to extract out common functionality for easy re-use. The closest relation is C# extension methods, but those aren't an exact match since they exist in a static context.
- Modules can add state to an existing class.
- Modules can have static methods
- Modules can act as namespaces

Example:

```
module SimpleConversation
  class NamespacedExample
    def poke
      puts "ouch"
    end
  end
end

attr_accessor :partner_name
def converse
  partner_name ||= "Slowpoke"
  speak + "\n#{partner_name}: Yes they are"
end

def self.yay
  puts "yay"
end
end

class Foo
  include SimpleConversation
  attr_accessor :name

  def speak
    name ||= "Speedy"
    "#{name}: tacos are yummy"
  end
end
```

```
end
end

x = Foo.new
x.name = "Joe"
x.partner_name = "Max"
puts x.speak
puts x.converse

y = SimpleConversation::NamespacedExample.new
y.poke

SimpleConversation.yay
```

Share Improve this answer

answered Feb 26, 2009 at 5:40

Follow



[segy](#)

644 ● 4 ● 12



7



I think I'd equate a module to something more akin to an extension method in C#. You're adding functionality to an existing class that is actually defined elsewhere. There isn't an exact analog in either C# or Java, but I definitely wouldn't think of it as an interface because the implementation is derived as well as the interface.



Share Improve this answer

answered Feb 22, 2009 at 22:13

Follow



[tvanfoss](#)

532k ● 102 ● 699 ● 798



For my own benefit here, am I wrong in my assessment?
That's how I have been using them up to now, maybe I am missing out on something. – [Ed Swangren](#) Feb 22, 2009 at 22:22

You can include a module in a Ruby class and inherit its behavior, instance methods at least. Take a look at ruby-doc.org/core/classes/Module.html – [tvanfosson](#) Feb 22, 2009 at 22:41

That's true, I did know about that, I just rarely use it in rpg maker. Thanks, I see the point now. – [Ed Swangren](#) Feb 23, 2009 at 0:10



4



A Module in ruby is a bit of scope/namespace that can be added to other things. They are used for two distinct but related purposes: bundling up a bunch of related things (constants, classes, etc.) that belong together and then adding them into some other scope (like multiple inheritance).

For example, there are modules called things like Comparable and Enumerable and so forth that encapsulate the functionality you'd expect something to have if these adjectives applied. By providing a basic definition (a method that compares two instances for Comparable and an iterator for Enumerable) you can import the module and find yourself with the full interface.

On the other hand there are modules like Net that you would seldom include in a class but which provide a bundle of functionality such as Net::HTTP, Net::FTP, Net::SMTP, and so on.

In the middle there are things like Math and Process which might be used one way or the other.

-- MarkusQ

Share Improve this answer

answered Feb 26, 2009 at 5:22

Follow



MarkusQ

21.9k ● 4 ● 57 ● 68



3



No. A module in ruby is more akin to a static class. I am not a Java developer, but I am *guessing* that Java interfaces are similar to C# interfaces, i.e., they define a contract, but not an implementation.

I should not that, while I have experience in ruby, it is in video game creation (RGSS/2). I am sure that I am ignorant of many things a veteran ruby programmer would know.

Share Improve this answer

edited Feb 22, 2009 at 22:15

Follow

answered Feb 22, 2009 at 22:09



Ed Swangren

125k ● 24 ● 188 ● 268

so if a `ruby module` is like a `Java static class`,
then is a `ruby module` like a `C# namespace` ?

– Don Cheadle Feb 3, 2015 at 23:07



From formal point of view modules in Ruby implement design pattern called "mixin".

2

<http://en.wikipedia.org/wiki/Mixin>



It can be compared to PHP traits.

<http://php.net/manual/en/language.oop5.traits.php>



Such architecture is useful in languages that don't allow multiple inheritance, e.g. Ruby, PHP.

Share Improve this answer

answered May 6, 2012 at 14:13

Follow



[shark555](#)

2,572 ● 1 ● 20 ● 10
