Ruby master - Feature #16769

Struct.new(..., immutable: true)

04/08/2020 09:00 AM - k0kubun (Takashi Kokubun)

Status:	Rejected
Priority:	Normal
Assignee:	
Target version:	

Description

Background

We've discussed interface to pass Struct attributes (like immutable: true, which is actually not added yet) at once. But I believe just adding immutable: true alone is really helpful in various cases. Thus I've spun out this ticket only for immutable: true from [Feature #16122].

Proposal

```
Post = Struct.new(:id, :name, immutable: true)

post = Post.new(1, "hello world")
post.id = 2 # NoMethodError (undefined method `id=' for #<struct Post id=1, name="hello world">)
```

Given immutable: true, an instance returned by .new is frozen, and writer methods are not defined.

Use case

- Allow using Struct's nice features when we need an immutable model, instead of defining a normal class with attr_readers and methods to support the Struct's features.
  - If it were a Struct, to_s, inspect, ==, and a bunch of other methods are nicely defined by default. Deconstructing a Struct on Pattern Matching is also available.
    - This level of support from the entire ecosystem may not be available if it's just a third-party library.
  - We could achieve a similar thing if we call Post.new(...).freeze or override #initialize to call freeze inside it, but it is not fun and feels like a workaround.
    - Today I suggested to use Struct for a model class to take advantage of the above benefits in a code review, but the implementation stuck with a bare class with attr_readers because the author didn't want writer methods to be defined (of course we don't want to manually undef them from a Struct class either) and calling freeze to workaround it seems tricky. I strongly desired Ruby's Struct is useful enough to cover this use case.

Related issues:

Related to Ruby master - Feature #16122: Struct::Value: simple immutable valu...	**Feedback**

History

#1 - 04/08/2020 09:01 AM - k0kubun (Takashi Kokubun)

- Backport deleted (2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN)

- Tracker changed from Bug to Feature

#2 - 04/08/2020 09:03 AM - k0kubun (Takashi Kokubun)

- Related to Feature #16122: Struct::Value: simple immutable value object added

#3 - 04/08/2020 10:45 AM - shevegen (Robert A. Heiler)

Makes sense.

#4 - 04/08/2020 06:35 PM - Eregon (Benoit Daloze)

Agreed, and ioquatix (Samuel Williams) and headius (Charles Nutter) seemed positive too in some recent discussion.

#5 - 04/09/2020 01:05 AM - mame (Yusuke Endoh)

It would be good to reuse an existing "freeze" mechanism.

```
Post = Struct.new(:id, :name, freeze: true)

post = Post.new(1, "hello world")
p post.frozen? #=> true
post.id = 2 #=> FrozenError
```

I'd like to avoid the word immutable because it is a new terminology and a negative form.

#6 - 04/09/2020 02:05 AM - nobu (Nobuyoshi Nakada)

How about:

```
Freezing = ->*{def initialize(...) super; freeze; end}
Post = Struct.new(:id, :name, &Freezing)
```

#7 - 04/09/2020 01:36 PM - Eregon (Benoit Daloze)

nobu (Nobuyoshi Nakada) setter methods shouldn't be defined, so just .freeze is not enough.

#8 - 04/10/2020 08:38 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

I don't like the keyword argument that changes the fundamental behavior. I prefer #16122 to this proposal.
Let's discuss there.

Matz.

#9 - 04/10/2020 12:15 PM - Eregon (Benoit Daloze)

Sad to see this rejected as there was a lot of agreement here, and I think #16122 might take a lot longer before anything gets implemented.