

Re: Question about Typesafe Enums

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2005-12/msg02218.html>

- *From:* "Rhino" <no.offline.contact.please@xxxxxxxxxx>
 - *Date:* Wed, 14 Dec 2005 12:45:05 -0500
-

"Hendrik Maryns" <hendrik_maryns@xxxxxxxxxxxxxx> wrote in message
[news:dnph8d\\$1@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:dnph8d$1@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)

Rhino schreef:

> I'm trying to incorporate some of the new features of Java 1.5 into some
> of my existing classes but I'm somewhat confused about how to do something
> with specific with typesafe enums.

>

> The main article in the SDK documentation is

> <http://java.sun.com/j2se/1.5.0/docs/guide/language/enums.html>. Based on

> that article, I'd like to go a beyond something they show in the Planets

> example. I need the guidance of those of you who understand this better to

> tell me if I can do what I want to do. I don't have the terminology

> straight in my mind yet so forgive me if I don't articulate this

> correctly.

>

> In the Planet enum that is shown in the article, they have a list of

> planets and each planet has two specific values, a mass and a radius. The

> enum provides methods mass() and radius() to get those values when they

> are needed. Therefore, if I want to get the mass of the Earth, my class

> can get it via Planet.EARTH.mass(), e.g.

>

> System.out.println("The mass of the Earth is " + Planet.EARTH.mass());

>

> So far so good.

Indeed.

Now, would it be possible to set up other enums to

> correspond to the values associated with the planets and use them to

> reference the values? For example, could I create an enum called Mass that

> was associated with the mass of each planet, then refer to the mass of a

> specific planet via the new enum, something like this:

>

> System.out.println("The mass of the Earth is " + Planet.EARTH.Mass);

>

> I'm not wedded to the syntax in that last example; in fact, it looks a bit

> nasty to me. I'm just trying to indicate that it would use the new Mass

> enum instead of the mass() method to refer to the first value associated

> with the given planet.

Re: Question about Typesafe Enums

I don't think you can do this sort of hard-linking two enums together, but you could define `mass()` to return something of type `Mass`, and in the initialisation of `Earth` tell it to return `Mass.EARTH_MASS`.

- > It would be neat to be able to avoid writing methods like `mass()` and just
- > use additional enums like `Mass` to refer to the specific facts that I want.
- > Basically, I'd like to be able to use meaningful words like `Mass`, rather
- > than numbers, as subscripts to the implied array that is in the `Planet`
- > enum.

Why would you want to have anything to do with those numbers? Although they are used by the JVM internally, there is absolutely no to access them. And how do you see this connected to `Mass`?

- > How could I do what I want to do?

What DO you want to do?

And, maybe more importantly, would it be a
> good idea or would it cause problems?

Doesn't seem like a good idea, but you are really unclear.

H.

=====
I was going to intersperse my followups within your post, just like you did to mine, but my newsreader isn't co-operating by putting >> in front of my original words and > in front of your reply....

I'm sorry to be so unclear; like I said, I'm still struggling with the terminology. Let me try explaining it a bit differently.

Okay, let's say we have an array called `Planet` and it looks like this (I'll just use the inner planets to minimize typing):

```
Object[][] Planet = {
{"Mercury", 3.303e+23, 2.4397e6, 0},
{"Venus", 4.869e+24, 6.0518e6, 0},
{"Earth", 5.976e+24, 6.37814e6, 1},
{"Mars", 6.421e+23, 3.3972e6, 1}
};
```

In this array, the first column contains a planet name, the second column contains the mass of the planet, the third column contains the radius of the planet, and the fourth column contains the number of natural satellites that the planet has.

If I want to know the number of satellites for the planet Mars, I can get it via this: `Planet[3][3]`. But the meanings of the '3's in the subscripts are not terribly obvious to someone looking at the code; no one would have any

Re: Question about Typesafe Enums

way of knowing the `_meaning_` of the fourth row or the fourth column of the array.

I'd much rather be able to reference the number of satellites for a planet with syntax that demonstrates a knowledge of the nature of the data, i.e. the first column contains the planet's name, the fourth column is the number of satellites, etc. To my way of thinking, this array has (implied) column names and those column names comprise an enum consisting of the values `NAME`, `MASS`, `RADIUS`, and `SATELLITES`. Also, the individual rows of the array can be uniquely identified by the planet name which is therefore an implied enum too, consisting of the values `Mercury`, `Venus`, `Earth` and `Mars`.

I'm not sure how the syntax should look but I'm picturing something that has words from enums in place of numeric subscripts, something like this: `Planet["Mars"][SATELLITES]`, which means 'the number-of-satellites value from the row of the array `Planet` whose key (first column value) is "Mars"'.

Again, this syntax is probably not entirely appropriate for maintaining consistency with other syntax in Java but it gives the flavour I am looking for.

I would also like to be able to use variables so that it would be possible to loop through the array and list the number of satellites for each planet without using a numeric subscript. Something like:

```
for (String name, int satellites : Planet) {  
    System.out.println("Planet " + name + " has " + Planet[NAME][SATELLITES]  
    + " satellites.");  
}
```

Sorry, I know that won't compile as it is. I'm not trying to make a firm proposal on what the syntax should be, I'm just trying to convey the flavour of what I'd like to do.

It seems to me that Enums are the vital ingredient in making this work but I may be perverting them beyond their intended use. Or maybe there is a much better way to do this without using Enums at all.

I'm just looking for some feedback on whether the concept I'm exploring even makes sense and, if it does, how could I accomplish it with typesafe enums?

Rhino

-
- *Follow-Ups:*
 - ◆ *Re: Question about Typesafe Enums*
 - ◇ *From: ricky.clarkson@xxxxxxxxxx*

Re: Question about Typesafe Enums

- **References:**

- ◆ **Question about Typesafe Enums**
 - ◇ *From:* Rhino
- ◆ **Re: Question about Typesafe Enums**
 - ◇ *From:* Hendrik Maryns

- Prev by Date: **Re: Program to list all dirs, subdirs and files older than 1 year**
- Next by Date: **Executing a compiled class' main from another class**
- Previous by thread: **Re: Question about Typesafe Enums**
- Next by thread: **Re: Question about Typesafe Enums**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**