Subject: What about real polymorphism ??
Posted by Antonio Santiago on Thu, 09 Dec 2004 14:12:12 GMT
View Forum Message <> Reply to Message

Hi group,

my problem is with IDL's polymorphism, i think it's a half-polymorphims instead real polymorphism.

Supose we have a class Person and two subclasses Man and Woman. Class Person has a class called GetInfo() that is overriden in both Man and Woman. The question is:

How can i create an array of Person's (that is Man or Woman objects) and call the method GetInfo() in the way that depends on the subtype of every object it invoques the Man GetInfo() or the Woman GetInfo()??

One possible solution is using OBJ_ISA funtion for every object in the array but i want to know if it is possible only with polymorfism in IDL like in Java or other OO languages.

Thanks.

Subject: Re: What about real polymorphism ??
Posted by Michael Wallace on Tue, 14 Dec 2004 08:29:57 GMT
View Forum Message <> Reply to Message

> A singleton is very possible with IDL:
> Just use a global variable (also as your access point) and
> check its existance in the constructor.

You know you drive all of us OO guys crazy with comments like "just use a global variable"?

There is a big difference between a global variable and a singleton. You're only thinking of the access point, not the rest of the pattern. I'd be VERY surprised if it's possible to implement a true singleton pattern in IDL. I'd love to be proved wrong, but I don't think it's possible with IDL's so-called "objects."

Mike

Subject: Re: What about real polymorphism ??
Posted by JD Smith on Tue, 14 Dec 2004 17:06:48 GMT
View Forum Message <> Reply to Message

On Tue, 14 Dec 2004 02:29:57 -0600, Michael Wallace wrote:

>> A singleton is very possible with IDL:
>> Just use a global variable (also as your access point) and
>> check its existance in the constructor.
>
> You know you drive all of us OO guys crazy with comments like "just use
> a global variable"?
>
> There is a big difference between a global variable and a singleton.
> You're only thinking of the access point, not the rest of the pattern.
> I'd be VERY surprised if it's possible to implement a true singleton
> pattern in IDL. I'd love to be proved wrong, but I don't think it's
> possible with IDL's so-called "objects."
>
> Mike

Good Wiki article with description and plusses and minuses on singletons: http://c2.com/cgi/wiki?SingletonPattern

My favorite quote:

> Singleton is a social disease. Because Singleton is so easy to understand, it is the single pattern that almost anybody who merely thumbed through GoF at the bookstore can remember and explain. That leads to unwarranted exuberance for the pattern among those who can least afford such exuberence.

You *can* implement a singleton in IDL (yes, using a common block or system variable). Try a search:

http://groups.google.com/groups?as_q=singleton&as_ugroup =*idl-pvwave*

The chief drawback is, since there are no class methods in IDL, either you can't override or extend the code which enforces "only one instance" (without modifying it directly), or you have to use a regular method as if it were a class method, discarding a "dummy" instance. A bit ugly, but it gets the job done. Note that IDL>6 won't let you delete the self variable in Init, so my old Singleton class would need changing. I prefer the singleton generator function technique.

JD

Subject: Re: What about real polymorphism ??
Posted by Michael Wallace on Tue, 14 Dec 2004 18:56:47 GMT

> My favorite quote:
>
> Singleton is a social disease. Because Singleton is so easy to
> understand, it is the single pattern that almost anybody who merely
> thumbed through GoF at the bookstore can remember and explain. That
> leads to unwarranted exuberance for the pattern among those who can
> least afford such exuberence.

That's a pretty good quote. Really, you could say that about any of the patterns. One thing that novice programmers will do is discover the Singleton, Decorator, Visitor, Factory or some other pattern and then try to apply the pattern to every possible task usually with great exuberance. What you wind up with are programs that are 10 times as big as what they need to be (if not more) and layer upon layer of needless complexity. I've suffered that same exuberance before, so I know exactly what it feels like and what it feels like to look at the code a year later. :-p

> You *can* implement a singleton in IDL (yes, using a common block or
> system variable). Try a search:
>
> http://groups.google.com/groups?as_q=singleton&as_ugroup =*idl-pvwave*

That's interesting. I had no clue that you could create your own system variables. While not totally ideal, the read only system variable allows you to do what I thought that IDL couldn't do -- be able to enforce the rule of having one and only one class in existence. The previous example just said to have a global variable, which wouldn't be a singleton at all because you could instantiate as many of those variables as you want. I sure didn't know of any way to limit IDL so that you could only create a single instance. I must admit that I am VERY surprised that IDL has this feature and happy to see it. Will wonders never cease?

Mike

Subject: Re: What about real polymorphism ??
Posted by David Fanning on Tue, 14 Dec 2004 21:58:56 GMT

Michael Wallace writes:

> Will wonders never cease?

No, never. I suppose it's even possible I'll find a reason

for an iTool. :-)

Cheers,

David

P.S. I notice that two of the four iTool examples on
the RSI code site I couldn't get to work (following
the author's notes). I'm not sure what this says about
iTools, but I'm afraid it doesn't make me hopeful. The
box plot just doesn't even want to try, giving a cryptic
error about "Internal Error" and no traceback. I wouldn't
know where to begin to look, but maybe that is the point. :-(

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http:/www.dfanning.com/
Phone: 970-221-0438, IDL Book Orders: 1-888-461-0155
