Subject: do I really need to use loops on objects?
Posted by Brad Gom on Tue, 30 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

It seems odd to me that I can't treat object arrays with the same elegance that is possible with all other array types. If I have a large list of objects of the same type, and I want to call the same method on each of them, do I really have to use a for loop? Wouldn't it be more consistent with the IDL philosophy to write: object_array->method() instead of: for i=0,10 do object_array[i]->method()

I'm just getting back into IDL after a short hiatus, and back to object programming in particular.. is there something I've missed?

Brad

Subject: Re: do I really need to use loops on objects?
Posted by J.D. Smith on Thu, 02 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

bjackel@phys.ucalgary.ca wrote:
>
> Brad Gom wrote:
>>
>> It seems odd to me that I can't treat object arrays with the same
>> elegance that is possible with all other array types. If I have a large
>> list of objects of the same type, and I want to call the same method on
>> each of them, do I really have to use a for loop? Wouldn't it be more
>> consistent with the IDL philosophy to write: object_array->method()
>> instead of: for i=0,10 do object_array[i]->method()
>
> I agree completely, but most people on this group don't appear to.
>
> The counter argument seems to be that an object array can contain
> different kinds of objects, so you can't be sure that any particular
> method will work for every element of the array. While true, this
> seems (to me) to be a reason why object arrays should contain only
> similar elements *JUST LIKE EVERY OTHER IDL ARRAY TYPE*.

Uhhh, how about pointer arrays? Think of Object arrays as object *pointer* arrays, as in C++, and it will become more intuitive. You want to call a method on an array of objects of the same class? Fine. Make them data members of another object which does the dirty work for you, including enforcing the single class requirement (see obj_isa())!

Since IDL would just be doing a loop anyway (you can't optimize polymorphism), this is just as convenient. I too at first felt the way you do, but then I

began to realize their point.

E.g.: With this superclass, you could then simply say

IDL> objarr->Method5

and have Method5 called on 50 objects at a time. You could make the superclass quite general, if you want (e.g. every class you ever write has a "Print" method...). When it comes to object oriented programming, to quote from Dune, , "you must bend like a reed in the wind."

JD

--

J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: do I really need to use loops on objects?
Posted by Mark Hadfield on Thu, 02 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote in message news:onr9h6o315.fsf@cow.physics.wisc.edu...
> So, object_array->method() *should* call "method" for each object in
> object_array, irregardless of the type of the object. If one of the
> objects doesn't define "method" then an exception should be raised.

I agree, but how should IDL handle data returned by the objects (either the result of function-type methods or the keyword data returned by a method like GetProperty)? There is no reason to expect that the return values of a group of objects should all be of a data type and dimensionality that would allow them to be bundled into an array.

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield/
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand

Subject: Re: do I really need to use loops on objects?
Posted by J.D. Smith on Fri, 03 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Struan Gray wrote:
>
> Reading this thread has piqued my curiosity as to what people are
> using arrays of objects for. I like to use IDL_containers instead, as
> they simplify adding and removing objects while maintaining a logical
> order that can be related to numerical array elements if you choose.
> As JD points out you can easily subclass the container, modify the add
> method so it gets picky as to what sort of objects it accepts, and
> provide a DoMethod method to call the relevant method for all the
> contained objects. Are there applications where doing it this way is
> a bad idea (or impossible)?
>
> Struan

If you implement your own IDL_Container-like class, it often contains an internal object array, though the one I use most contains merely a pointer array, so the array elements need not be simply objects. Basically, if you have a collection of objects which need not be intimately related (e.g. if you're just going to kill them all), an objarr is fine. It's lighter weight than a container, though less fully featured.

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: do I really need to use loops on objects?
Posted by Struan Gray on Fri, 03 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Reading this thread has piqued my curiosity as to what people are using arrays of objects for. I like to use IDL_containers instead, as they simplify adding and removing objects while maintaining a logical order that can be related to numerical array elements if you choose. As JD points out you can easily subclass the container, modify the add method so it gets picky as to what sort of objects it accepts, and provide a DoMethod method to call the relevant method for all the contained objects. Are there applications where doing it this way is a bad idea (or impossible)?

Struan
