
Subject: Object Programming in IDL

Posted by [graham_wilson_1234](#) on Tue, 21 May 2002 12:26:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

Just to appease Craig, I have started a new thread so I can avoid putting my comments after David's 'gosh golly' post ;) I am interested in hearing others comments...

The first point that we should all be very clear on is that IDL is NOT a particularly good example of an object oriented language. You can certainly emulate OOP concepts using IDL's objects and a select few functions/procedures but it often defeats the purpose of the OOP style. When someone mentions IDL objects, it is universally assumed that they really mean 'object graphics' which leads directly to point number 2; Object oriented programming != object graphics. Unfortunately, it is very difficult to dispel this myth using IDL because of point number 1.

With regard to writing object oriented code in IDL we are all rather stuck until RSI implements a more complete feature set. I generally define polymorphism as the ability to process objects differently depending on their data type or class. In this respect, the lack of operator overloading is an example where IDL fails to offer the full OOP tool set. Yes, you can overload methods, but operators should be no different. To compensate for this missing functionality one can write functions and/or procedures but this is better described as an overlay and you must rely on a naming convention or a path precedence to avoid conflicts. Personally, I'd like to see true polymorphism (with overloading) and public/private methods sooner rather than later (is anyone at RSI listening?).

A good technical book describing the merits of using objects in data analysis is "Programming with Data: A Guide to the S Language" (ISBN: 0-387-98503-4). The concepts described are specific to S-Plus but can be adapted to any OOP language. While they may seem abstract at first, they are a very powerful way of manipulating and modelling data. A free alternative to S-Plus is R (www.r-project.com).

For what it is worth, Matlab has a slightly more complete implementation of OOP. The one glaring (and annoying) feature missing from Matlab, however, is the absence of pointers and therefore dynamic structures/sizing. This, of course, is a gripe for a different newsgroup...

I lurk therefore I am.
Graham

Subject: Re: Object Programming in IDL

Posted by [Pavel A. Romashkin](#) on Thu, 23 May 2002 15:47:34 GMT

I think by now only a few people equate objects with object graphics.
This topic was brought up way too many times for that.

I think objects in IDL are very useful. They speed up development, help write reuseable code and make it easy to maintain. They may not be perfect but again - if anyone knows of any tool, be it programming language, a car or anything else - let us all know so we can all switch.

I think I will stay with IDL over C++ for data processing applications, despite the lack of operator overloading in IDL.

Cheers,

Pavel

Graham Wilson wrote:

>

> Just to appease Craig, I have started a new thread so I can avoid putting my
> comments after David's 'gosh golly' post ;) I am interested in hearing
> others comments...

>

> The first point that we should all be very clear on is that IDL is NOT
> a particularly good example of an object oriented language. You can
> certainly emulate OOP concepts using IDL's objects and a select few
> functions/procedures but it often defeats the purpose of the OOP style.
> When someone mentions IDL objects, it is universally assumed that they
> really mean 'object graphics' which leads directly to point number 2;
> Object oriented programming != object graphics. Unfortunately, it
> is very difficult to dispel this myth using IDL because of point number 1.

>

> With regard to writing object oriented code in IDL we are all rather stuck
> until RSI implements a more complete feature set. I generally define
> polymorphism it as the ability to process objects differently depending on
> their data type or class. In this respect, the lack of operator overloading
> is an example where IDL fails to offer the full OOP tool set. Yes, you can
> overload methods, but operators should be no different. To compensate for
> this missing functionality one can write functions and/or procedures but
> this better described as an overlay and you must rely on a naming
> convention or a path precedence to avoid conflicts. Personally, I'd like
> to see true polymorphism (with overloading) and public/private methods
> sooner rather than later (is anyone at RSI listening?).

>

> A good technical book describing the merits of using objects in data
> analysis is "Programming with Data: A Guide to the S Language"
> (ISBN: 0-387-98503-4). The concepts described are specific to S-Plus but
> can be adapted to any OOP language. While they may seem abstract at first,
> they are very powerful way of manipulating and modelling data. A free
> alternative to S-Plus is R (www.r-project.com).

>

> For what it is worth, Matlab has a slightly more complete implementation of
> OOP. The one glaring (and annoying) feature missing from Matlab, however,

> is the absence of pointers and therefore dynamic structures/sizing. This,
> of course, is a grip for a different newsgroup...
>
> I lurk therefore I am.
> Graham

Subject: Re: Object Programming in IDL
Posted by [K. Bowman](#) on Thu, 23 May 2002 21:00:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <3CED0F15.2FA1F456@hotmail.com>,
"Pavel A. Romashkin" <pavel_romashkin@hotmail.com> wrote:

> I think by now only a few people equate objects with object graphics.

I thought that there were four kinds of objects: objects, graphics objects, sex objects, and graphic sex objects. The first two are covered in this newsgroup (for IDL at least). The last one, especially, is found in the alt.binaries hierarchy.

Ken

Subject: Re: Object Programming in IDL
Posted by [Craig Markwardt](#) on Fri, 24 May 2002 02:52:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

graham_wilson_1234@yahoo.ca (Graham Wilson) writes:

> Just to appease Craig, I have started a new thread so I can avoid putting my
> comments after David's 'gosh golly' post ;) I am interested in hearing
> others comments...

Ahh, don't worry Graham, I don't normally read object postings :-)

> The first point that we should all be very clear on is that IDL is NOT
> a particularly good example of an object oriented language. You can

I am an IDL object not-even-newbie, but I've done some other "real" object oriented programming. Your desires and suggestions seem pretty reasonable.

Craig

Subject: Re: Object Programming in IDL
Posted by [R.Bauer](#) on Fri, 24 May 2002 08:53:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Graham,

I hope you already have done a feature request to RSI directly.

At the moment it's not clear to me why you believe this feature is so important for the IDL language.

The parts of Object Language already in IDL implemented are full working or not ?

regards
Reimar

Graham Wilson wrote:

>
> Just to appease Craig, I have started a new thread so I can avoid putting my
> comments after David's 'gosh golly' post ;) I am interested in hearing
> others comments...
>
> The first point that we should all be very clear on is that IDL is NOT
> a particularly good example of an object oriented language. You can
> certainly emulate OOP concepts using IDL's objects and a select few
> functions/procedures but it often defeats the purpose of the OOP style.
> When someone mentions IDL objects, it is universally assumed that they
> really mean 'object graphics' which leads directly to point number 2;
> Object oriented programming != object graphics. Unfortunately, it
> is very difficult to dispel this myth using IDL because of point number 1.
>
> With regard to writing object oriented code in IDL we are all rather stuck
> until RSI implements a more complete feature set. I generally define
> polymorphism as the ability to process objects differently depending on
> their data type or class. In this respect, the lack of operator overloading
> is an example where IDL fails to offer the full OOP tool set. Yes, you can
> overload methods, but operators should be no different. To compensate for
> this missing functionality one can write functions and/or procedures but
> this better described as an overlay and you must rely on a naming
> convention or a path precedence to avoid conflicts. Personally, I'd like
> to see true polymorphism (with overloading) and public/private methods

> sooner rather than later (is anyone at RSI listening?).

>
> A good technical book describing the merits of using objects in data
> analysis is "Programming with Data: A Guide to the S Language"
> (ISBN: 0-387-98503-4). The concepts described are specific to S-Plus but
> can be adapted to any OOP language. While they may seem abstract at first,
> they are very powerful way of manipulating and modelling data. A free
> alternative to S-Plus is R (www.r-project.com).
>
> For what it is worth, Matlab has a slightly more complete implementation of
> OOP. The one glaring (and annoying) feature missing from Matlab, however,
> is the absence of pointers and therefore dynamic structures/sizing. This,
> of course, is a grip for a different newsgroup...
>
> I lurk therefore I am.
> Graham

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml
=====

Subject: Re: Object Programming in IDL
Posted by [Paul Van Delst\[1\]](#) on Fri, 24 May 2002 11:59:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Reimar Bauer wrote:

>
> Dear Graham,
>
> I hope you already have done a feature request to RSI directly.
>
> At the moment it's not clear to me why you believe this feature
> is so important for the IDL language.

I use operator overloading in Fortran 90/95 all the time for derived type (structure) arithmetic. Not just overloading existing operators either - you can define new ones too.

Very handy. If it's useful in Fortran, well then....

- > The parts of Object Language already in IDL implemented
- > are full working or not ?

I think that (and here I'm wildly speculating on Grahams POV) is that the emphasis is on the word "parts" in the above sentence.

paulv

--

Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP/EMC purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul

Subject: Re: Object Programming in IDL
Posted by [Dick Jackson](#) on Fri, 24 May 2002 14:51:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Craig Markwardt" <craigmnet@cow.physics.wisc.edu> wrote in message
news:on7klui7fu.fsf@cow.physics.wisc.edu...

- >
- > graham_wilson_1234@yahoo.ca (Graham Wilson) writes:
- >
- >> Just to appease Craig, I have started a new thread so I can avoid
putting my
- >> comments after David's 'gosh golly' post ;) I am interested in hearing
- >> others comments...
- >
- > Ahh, don't worry Graham, I don't normally read object postings :-)

This reminds me of one posting I read in comp.object, a group for
discussions of O-O issues in any language. During a rather intense argument
among people with very strong opinions, someone pointed out that in the name
of the newsgroup, "object" was the noun, not the verb.

Which did you mean, Craig? :-)

Cheers,

--

-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

Subject: Re: Object Programming in IDL
Posted by [Craig Markwardt](#) on Fri, 24 May 2002 18:14:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Dick Jackson" <dick@d-jackson.com> writes:

>>

>> Ahh, don't worry Graham, I don't normally read object postings :-)

>

> This reminds me of one posting I read in comp.object, a group for
> discussions of O-O issues in any language. During a rather intense argument
> among people with very strong opinions, someone pointed out that in the name
> of the newsgroup, "object" was the noun, not the verb.

>

> Which did you mean, Craig? :-)

Heh, I like the ambiguity. :-)

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
