Subject: Re: Object Programming
Posted by penteado on Thu, 12 May 2011 19:41:09 GMT
View Forum Message <> Reply to Message

On May 11, 4:23 pm, David Fanning <n...@idlcoyote.com> wrote:
> I guess that's right. I usually check to see if parameters
> are undefined or not. In this case, because the actual programs
> are wrappers to the PLOT command, I wanted to know how many
> there were. Odd that I haven't run into this before, though. :-)

Do you DG's plot (what I would guess by 'command')?

I ask because I know there are some peculiar pitfalls in NG's plot():
instead of (what I expected) every class inheriting Graphic, and being
able to inherit from them, in the usual way, it is a very convoluted
system. I suspect the cause was to make it work while minimizing the
changes or additions to the way the iTools were organized.

The most confusing part is how the classes (like Plot) are created:
The functions (like plot()) are not the usual init functions, they are
separate functions that create a Graphic object (itself not created by
a init function), informing it of what kind of graphic to make. The
graphic() function creates the proper iTool, and and an object of the
proper class, which is just a wrapper and contains the Graphic object
inside.

This is why I gave up on trying to inherit Window for a class, and
contained it instead (you can see it at
http://www.ppenteado.net/idl/pp_lib/doc/pp_multiplot__define .html). I
do not remember whether it would be impossible or just complicated to
inherit without changing the code in some of IDL's routines (which
would be a very weird inheritance, if I had to change other code).