Subject: Re: When should objects be used?
Posted by mallors on Fri, 25 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <3772CE57.6BF093C8@wellesley.edu>,
"Richard G. French" <rfrench@wellesley.edu> writes:
> Nearly all of my IDL programs started as interactive bits of code, saved
> in
> a journal file and then expanded into programs. The main power of IDL
> for me is to be able to tinker with the code quite easily, try different
> things out, and then finalize the code in the fashion that works best.
> Most of these programs are the old-fashioned kind where you specify the
> conditions at the start (the data to be
> analyzed, the conditions of the analysis, the form of the result) and

[text omitted]

Hi,

I think the experts here in this group have probably given
you enough info to peak your interest, but I just want to point
out that one of the most useful features of object oriented
programming is "code reuse". Unfortunately, for the type of
exploratory science analysis you describe, it is sometimes
difficult to decide what procedures would (or could) be suitable
to be made into objects. I have found that for a lot of data
analysis code that no one but yourself will be using, a full
blown object-oriented design is not usually acceptable.

On the other hand, keep in mind that if you can write some
useful utility objects, you can use them easily within your
"old" procedural method of programming. For example, I was
finding myself always manipulating my data file names, extracting
the path, the file extension, etc. This type of task is perfect
for encapsulation into an object, so I created a simple "File"
object. This object can then be used anywhere in IDL - you don't
have to have an super-duper full object oriented program.
Perhaps the task you mention of writing TeX tables might be a
good place to start:

```
   Table = OBJ_NEW ('teXTable')   ; Make an instance of a 'teXTable' obj
   Table->setColumns (5)          ; This table will have 5 cols.
   Table->setData (myData)        ; myData might be an array of structures
                                  ;  with 5 tags.  You could even omit the
                                  ;  setColumns() method, and just figure
                                  ;  out the number of columns
   Table->write ('myTable.tex')   ; Write a table
   OBJ_DESTROY (Table)            ; Clean up memory
```

This object could then produce TeX tables for any of your datasets,.
Come to think of it, I would like to have this object...why don't
you post it when it's done ;-)

Regards,

-bob

By the way, you can take a look at the File object on my
web page: http://cspar.uah.edu/~mallozzir/software/idl/idl.html

--

~~~~~~ ~~~~~~~~~~
Robert S. Mallozzi 256-544-0887
Mail Code SD 50
Work: http://gammaray.msfc.nasa.gov/ Marshall Space Flight Center
Play: http://cspar.uah.edu/~mallozzir/ Huntsville, AL 35812
~~~~~~ ~~~~~~~~~~