Subject: Re: Commons, Was: can i place a job advert
Posted by Martin Downing on Tue, 27 Nov 2001 00:20:19 GMT
View Forum Message <> Reply to Message

Well I've removed all the graphics stuff -I havent got a clue what that was
about and you can add what you like later!
My solution is to link all objects as a singly linked list which has its
head pointer (named objTop in this code)
stored using a COMMON BLOCK so that it is visible to all object instances.
This could be considered a static object
member instance, rather like you can find in some C++ Classes. In C++ these
beasts have only one value but are visible from all instances of the class -
incredibly useful for stacks.
Ok so strictly the variable is not private to the class, but it takes
someone daft enough to copy the
COMMON identifier: "__TEST_PRIVATE_STATIC_MEMBERS" elsewhere to access the
pointer!
I think this is a very reasonable use of common blocks, but I await to see
alternatives.

cheers

Martin

When you run it you should get the following:
IDL> pavels_test
three
one

```
================================================
pro Test__define
result = {TEST, data_holder : ptr_new(),objPrevious:obj_new()};, inherits
'IDLgrModel'}
end

function TEST::init, data=data
 COMMON __TEST_PRIVATE_STATIC_MEMBERS, objTop

 ; if undefined, point objTop to a NULL object
 if n_elements(objTop) eq 0 then objTop = obj_new()

 self.objPrevious = objTop
 objTop = self

 ; add the rest of your initialisations
```

```
 self.data_holder = ptr_new(data)

 return, 1
end



function TEST::GET_OTHERS, index ;FromTop
 COMMON __TEST_PRIVATE_STATIC_MEMBERS, objTop

 obj=objTop
 i = 1
 while i LT index do begin
    if obj_valid(obj) then obj = obj.objPrevious
  if obj ne self then i = i+1
 endwhile

 if obj_valid(obj) then data = *(obj.data_holder) else data = 0

 return, data
end

; delete a single instance
pro TEST::CleanUp
 COMMON __TEST_PRIVATE_STATIC_MEMBERS, objTop

 ; free data
 ptr_free, self.data_holder

 ; relink object pointers
 if self eq objTop then begin
  objTop = self.objPrevious
 endif else begin
  obj=objTop
  while obj ne self do begin
   help, obj
   next = obj
     obj = obj.objPrevious
  endwhile
  next.objPrevious = self.objPrevious
 endelse
end

; delete all instances of this object class
pro TEST::CleanUpAll
 COMMON __TEST_PRIVATE_STATIC_MEMBERS, objTop

 obj=objTop
```

```
 while obj_valid(obj) do begin
    objPrev = obj.objPrevious
  obj_destroy, obj
  obj = objPrev
 endwhile
end


pro pavels_test

a = obj_new('TEST', data = "one")
b = obj_new('TEST', data = "two")
c = obj_new('TEST', data = "three")

;then it is possible to do something like
print, b->get_others(1)
print, b->get_others(2)

c->CleanUpAll

end
```

 ========== =

--

Martin Downing,
Clinical Research Physicist,
Grampian Orthopaedic RSA Research Centre,
Woodend Hospital, Aberdeen, AB15 6LS.
Tel. 01224 556055 / 07903901612
Fax. 01224 556662

m.downing@abdn.ac.uk

"Pavel A. Romashkin" <pavel.romashkin@noaa.gov> wrote in message
news:3C02996F.ABD6D1D8@noaa.gov...
> This topic is irresistable.
> How about we issue a Challenge:
>
> Please modify the object definition below or create a method for objects
> to be aware of each other after creation:
> ;**********
> pro Test__define
> result = {'TEST', data_holder : ptr_new(), inherits 'IDLgrModel'}
> end
>

> ;so that if
>
> a = obj_new('TEST')
> b = obj_new('TEST')
>
> ;then it is possible to do something like
>
> b_CTM_matrix = a -> Get_others_CTM(a_keyword = '?')
>
> ; without passing B to the call?
> ;**********
> Submissions using bulk heap searches and .sav files are not allowed :)
>
> Cheers,
> Pavel
>
> Richard Younger wrote:
>>
>> Mind you, I'm not supporting banning commons as dogma, but I think there
>> are enough general objections to them to ask people to think a bit
>> before they rush out and use them everywhere they can.
