
Subject: array of objects?

Posted by [thomas.bielser](#) on Tue, 12 Feb 2002 07:27:17 GMT

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

I decided to change IDL-routines into oo. Simplified therefore I had to create an own class:

```
; creating an own class
pro own_class__define
    struct = { own_class, $
              integer_value : 0L $
            }
end
```

...containing two methods for setting and reading the integer_value of an instance of this class:

```
; setproperty-method of the 'own class'
pro own_class::setproperty, integer_value=integer_value

    if ( n_elements( integer_value ) ne 0 ) then
self.integer_value = integer_value

end

; getproperty-method of the 'own class'
function own_class::getproperty, integer_value=integer_value

    if keyword_set( integer_value ) then begin

        return, self.integer_value

    endif

end
```

... now I create an array of 100 elements of this class with the replicate-command:

```
; array for holding 100 elements of the type 'own_class'
a = replicate( OBJ_NEW( 'own_class' ), 100)
```

... I label the integer_value-property of every element of this array with its index (instead of a name, or whatever):

```
; label each array-element with it's number
for i=0, 99 do begin

    ; call the method setproperty
```

```
a[ i ] -> setproperty, integer_value = i
```

```
endfor
```

... here now something strange happens: From IDL I would expect from the following commands, the following output:

```
IDL> print, a[0] -> getproperty( /integer_value )  
0  
IDL> print, a[50] -> getproperty( /integer_value )  
50  
IDL> print, a[99] -> getproperty( /integer_value )  
99
```

...but IDL comes up with:

```
IDL> print, a[0] -> getproperty( /integer_value )  
99  
IDL> print, a[50] -> getproperty( /integer_value )  
99  
IDL> print, a[99] -> getproperty( /integer_value )  
99
```

Is this a feature or a bug? Many thanks in advance!

Thomas
