

Re: Declarations variables, Dim, some guidance please

Source:

<http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.programming/2005-08/msg05980.html>

- *From:* Neal Zimm <nealzimm@xxxxxxxxxx>
 - *Date:* Tue, 16 Aug 2005 13:50:02 -0700
-

Thanks to all for the sound advice, I hope y'all don't mind a common response. I'm relatively new to the VB game, self taught (with help from this bulletin board), and there's LOTS I don't know. (see later re: class variables.)

My uses of 'declared' vars in most cases, fall into all of the categories mentioned.

1. vars where the values DON'T change, but are used by many subs. My application is called "RM" and I have a sub called zRM_Values which is called at the beginning of larger macros. (BTW, I have a bad right pinky finger and typing quotes "" is a bother)

2 examples:

declared: dim Yes as string
in the zrm_values sub, Yes = "Y" , so in any other macro: if other_var_name = yes then

declared: dim PressEnter as string
in zrm_values: PressEnter = "Press Enter or click OK after typing.." & vbcr
' used in inputbox dialogs

2. I have tried mightily to have the module declared vars fall into two other categories:

a) they have the same meaning in any macro in the module so if the value is changed, it's kinda a good thing for me that other macros have access to the 'latest' value.

b) I don't care if the value changes.
example: dim x as integer

since I use option explicit, without dim'ing x in every procedure, I can use: for x = 1 to whatever

3. I have not learned what a "class with properties" is yet, nor what a collection class is. Is the Excel help adequate on this topic or can you recommend other reading sources?

Again,
Thanks to all. Your help is very much appreciated.
Neal Z.
--
Neal Z

"Jake Marx" wrote:

> Ni Neal,
>
> Neal Zimm wrote:
>> In an application that I'm developing I have dim'd quite a few
>> variables in Declarations. I'll admit some of it is not wanting to
>> take the time to put those vars that are used quite often in many
>> macros within the sub SubName(var list) parenthesis.
>>
>> 1) What advice can you offer on the pro's and cons of this technique?
>> All of the application's code is in ONE module.
>
> I don't like to declare variables at module level or globally (Public
> keyword in standard module) unless I have to. Some reasons are readability,
> ease of maintenance, and knowing your variables won't be "trounced upon" by
> various procedures. Module-level and global variables will also use more
> resources.
>
> The alternative is to pass variables via arguments to functions/subroutines.
> It is undoubtedly more work upfront, but IMO it pays off to do it this way.
>
>> 2) I got 'bitten' when testing a macro where a var called Draw was
>> dim'd as integer in Declarations, had a good value > 0 in prior
>> macros, but was 0 in the macro I was testing.
>>
>> Sure enough, I had dim'd it again, inadvertantly, also as Integer
>> in the macro being tested. Hence the 0 value, I guess. The QUESTION
>> is, why did I NOT get a duplicate Dim error?
>
> Well, since the variables had different scopes, it wasn't a duplicate
> declaration. You had one module-level variable named Draw, and you had a
> local variable named Draw with a completely different value.
>
> --
> Regards,
>
> Jake Marx

