

Re: Should this be an object?

Source:

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2007-06/msg01555.html>

- *From:* "Ralph" <nt_consulting64@xxxxxxxxxx>
 - *Date:* Sat, 23 Jun 2007 09:24:19 -0500
-

"MP" <Nospam@xxxxxxxxxx> wrote in message
[news:467c0388\\$0\\$3575\\$a4e5890@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:467c0388$0$3575$a4e5890@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

<snipped>

as you can see I'm still quite confused on all this...but really, I am trying to understand.. <g> maybe just not successful :-)

To paraphrase Stroustrup, poorly... <g>

Objects (classes) exist to help a programmer organize his code more logically. To make it easier to avoid mistakes and to help him find and repair mistakes if a mistake is made. Period.

An object-oriented program is simply one that is build up from the colaboration of various types of objects through well-defined interfaces. There are no specific rules as to how these interfaces should be used, or even what the interfaces should look like. The only rule is that they should provide some logical services which simplifies one's understanding of the solution and makes it easier to maintain. To that end, a whole range of mechanisms are at one's disposal – Abstraction, encapsulation, polymorphism, hierarchies, etc.

On the other hand, sometimes an object isn't needed to provide a solution. Sometimes everything a process needs is right at hand and fully understandable in place. In that case no amount of encapsulation, abstraction, polymorphism, ... is going to improve anything. (Quite the opposite usually.)

Usually the decision of whether to use an object or not, or what kind of services to support if an object is choosen, is totally dependent on service requirements outside the mechanics of OO. Anything from how many may be needed to how often you may need to revisit will likely impact your decision. One commonly has to adopt a holistic view.

Re: Should this be an object?

Your so called "unsuccessful" solution may be ideal in one case, and dismal in another. To know the difference – trust your instincts. If something feels silly or awkward – it is usually silly or awkward. If something seems right – it is probably a good working solution. Let the problem be your guide to developing a solution. Don't try and force a solution on a problem. Or make a problem out of a solution. <g>

hth
–ralph