

Re: OOB or procedural

Source: <http://coding.derkeiler.com/Archive/General/comp.programming/2003-12/0074.html>

From: Philip (*phlip_cpp_at_yahoo.com*)

Date: 12/01/03

Date: Mon, 01 Dec 2003 14:03:03 GMT

Gerry Quinn wrote:

> >If that breed of non-Standard C supported a kind of inheritance, and if
you
> >define OO as inheritance, then saying it's not OO because it's C is empty
> >dogma.
>
> But you referred to an OS that supported it, not a dialect of C.
> Language features are not defined by those of the APIs they can access,
> or the code that is written in them!

Implementors tweak the C languages to match OSs all the time.

> >Of course Standard C doesn't have inheritance!
> >
> >MS Windows's SendMessage architecture takes this only one step further.
> >Applications register pointers to functions with each Window object
> >(including the little Control Window objects embedded in top-level
windows).
> >Then the system dispatches messages to individual windows, and these
route
> >individual events in a big switch-case block. Modern wrappers hide this
> >inside the MESSAGE_HANDLER() macros, but it's still the exact same thing.
> >
> >These systems use C to explicitly create the same virtual dispatch
system –
> >function pointers and switch-case blocks – that OO languages implicitly
> >provide. But making those systems much lighter-weight allows one to use
them
> >much more strategically. The engineering expense of using them goes way
> >down; they become a viable alternative to individual 'if' statements.
>
> We all agree that one can implement OO ideas in non-OO languages. I
> don't know what you mean by making them lightweight, though. I doubt
> there are major differences in speed between the implementations.

By "lightweight" I meant easy and safe for a programmer to type in.

comp.programming: Re: OOB or procedural

>*From the top: Compilers implement Polymorphism to lower the cost (in terms of programmer editing time & overhead) of installing a virtual dispatch. The cost becomes lower than their naive implementation as function pointers and/or conditionals would be. This cost becomes so low that they become preferable to 'if' statements.*

--

Philip