

Re: Static vs. Dynamic typing (big advantage or not)---WAS: c.programming: OOP and memory management

Source: <http://coding.derkeiler.com/Archive/General/comp.programming/2004-07/2247.html>

From: Thomas Gagne (tgagne_at_wide-open-west.com)

Date: 07/27/04

Date: Mon, 26 Jul 2004 20:10:32 -0400

Michael N. Christoff wrote:

<snip>

>

>

> *This makes no sense, to me at least. How is the user of some software*
> *'always' more knowledgeable than the person who wrote it? This implies that*
> *before you send an object to another bit of code, you have to analyze the*
> *receiver to the point where you know more about it than the authors*
> *themselves. Seems like a waste of time to me, when we could simply have the*
> *receiver specify what objects are appropriate. ie: everyone takes*
> *responsibility for their own code. Its almost a restatement of the idea*
> *behind encapsulation. My object has a protected state and I (the receiver)*
> *take responsibility for protecting that state. I do not pass off that*
> *responsibility to the users of my code. Also, there is the whole issue of*
> *the use of commercial closed source software where there is no way the*
> *sender could be more knowledgeable about the receiver than the receiver*
> *itself.*

>

They don't have to know the receiver inside and out---only enough to know what it's supposed to do, then they should be able to send it anything that meets the /minimum/ requirements. In another subthread there's discussion about the effort required in most statically typed languages to define what a minimum interface is.

Don't limit consumers' knowledge of your objects to only a brief acquaintance.

They may be using it intimately for weeks, months, or years. In that amount of time they may have found more reasons and places to use it and perhaps have become more intimate than even its parents. These are typically long-term relationships and not one-night stands.

>

>> *Who is better equipped to determine the utility of a screwdriver,*
>> *the screwdriver or the*