Subject: Re: Cleaning up inherited object classes
Posted by btt on Wed, 03 Dec 2003 20:17:02 GMT
View Forum Message <> Reply to Message

M. Katz wrote:
> Cleaning up is my least favorite activity. Were my living room an IDL
> object I'm sure it'd be full of dangling pointer references. Here's a
> question regarding objects' Cleanup methods and inheritance.
>
> When an object inherits another object, methods can be overridden. So
> what happens to the CleanUp method? It is special.
>
> If my House object inherits the Living_Room and Bathroom object
> classes, will a call to HOUSE::CleanUp also call Living_Room::CleanUp
> and Bathroom::Cleanup when obj_destroy, self is called?
>
> Let me put that another way. Suppose an object class, A, has pointer
> fields. Unless someone tells me otherwise, I assume it's a good idea
> two specifically free the pointers in that object's Cleanup routine.
> Now, suppose another object class, B, inherits A. B has its own
> pointers to clean up as well, so I write that into its cleanup
> routine.
>
> It is sufficient to write the Cleanup methods like this?
>
> pro Bobj::CleanUp
> ptr_free, self.Bpointer
> obj_destroy, self
> end
>
> pro Aobj::CleanUp
> ptr_free, self.Apointer
> obj_destroy, self
> end
>
> Will Bobj::CleanUp's call to "obj_destroy, self" also call
> Aobj::Cleanup so that self.Apointer can be freed as the object is
> destroyed?
>
> Also, does the destruction of an object that contains a pointer field
> also inherently free the pointer? or is it necessary to specifically
> ask for that in the Cleanup?
>
> Now if I could only get the House::TakeOutTheTrash method to work
> reliably my wife would be thrilled.
>
> M. Katz
Hello,

I think you simply call the cleanup method for each superclass.
The following is the way I do it.

```
PRO House::Cleanup

DoMyOwnCleanUpofLocalPointersAndObjects

Self->Living_Room::CleanUp
Self->Bath_Room::Cleanup

END
```

Provided that HOUSE was defined this way.

```
PRO House__Define

struct = {House, $
 INHERITS Bath_Room, $
 INHERITS Living_Room}
END
```

If Bath_Room inherits from some other object, such as READING_ROOM, then
it will call that superclass' cleanup method if you have Bath_Room's
cleanup as ...

```
PRO Bath_Room::Cleanup

self->Reading_Room::Cleanup
END
```

I would defer to other's on the question if a subclass MUST have its own
CLEANUP method. I haven't tried it, but I'm not sure that it does
(unless it has its own pointers and objects to cleanup.)

Cheers,
Ben
