
Subject: Cleaning up inherited object classes

Posted by [MKatz843](#) on Wed, 03 Dec 2003 19:28:27 GMT

[View Forum Message](#) <> [Reply to Message](#)

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 to 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
