Subject: Re: Inheritance query
Posted by J.D. Smith on Wed, 10 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Bernard Puc wrote:
>
> Hello
>
> For the object programming gurus: I'm creating a class called data.
> I'm then creating subclasses of data called type1, type2, etc. The
> type1 class inherits the data class attributes. Now, is it possible to
> inherit, lets say, the data::INIT method and somehow add to it? Or, do
> I have to write an entirely new INIT method for type1 class which
> incorporates the statements in the data::INIT method?

By default, all methods are inherited. To add to the method, you need to "chain up" to the superclass, like this:

```
function SubClass::Init,_EXTRA=e
  if (self->SuperClass::Init(_EXTRA=e) ne 1) then return,0
  ;;; do more stuff
  return,1
end
```

This is called "extending" a method, and works for any method, not just Init. If you don't chain up, it's called "overriding" a method (which will never be called then). If you omit Init altogether, SuperClass::Init is called automatically, which I call "defaulting".

A note on where to chain: you generally want to chain-up first in Init, and last in Cleanup. In other methods, you'll have to choose the best place to chain.

Most good designs will have a fair number of defaulting, fewer extending, and a small number of overriding methods. Overridden methods represent new code which doesn't benefit from the work done in making its superclasses. I could inherit a class called "Autos" into my new class "FlatWare", override every single method from Autos, and never even know it was there.... clearly not too useful. This is not to say that overriding doesn't have its purposes though.

Good Luck,

JD

--
J.D. Smith |*| WORK: (607) 255-5842

Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|