
Subject: Call a method from a class

Posted by [vcarlos](#) on Wed, 15 Feb 2006 19:25:15 GMT

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

Hi all,

In some languages is possible to call methods from a Class, instead of a object. For instance, the class Messenger could call a method `reportError`, `message`. This is useful when I just need a "instance" of that object and everybody takes advantage of that (I think that is similar to Singleton design pattern). Is there any way to the same in IDL? Or I should set up some kind of library procedures/functions and use through my program?

Thanks a lot

Vinicius

Subject: Re: Call a method from a class

Posted by [Michael Galloy](#) on Sat, 01 Apr 2006 09:24:24 GMT

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

vcarlos wrote:

> In some languages is possible to call methods from a Class, instead of
> a object. For instance, the class Messenger could call a method
> `reportError`, `message`. This is useful when I just need a "instance" of
> that object and everybody takes advantage of that (I think that is
> similar to Singleton design pattern). Is there any way to the same in
> IDL? Or I should set up some kind of library procedures/functions and
> use through my program?

IDL doesn't provide you with a lot of extra syntax/features for object-oriented programming, but you can get around most of what's missing with good conventions and discipline.

For class methods, I usually just use the name of the class as a prefix to a normal routine name. So for example, the class method `sin` of the `Math` class would be:

```
function math__sin, x
```

The hard part is figuring out where to put it. I normally put my method definitions and the member variable definition all in one file. If you put the class method definitions in there too you have to instantiate an object of that class in order to call them. So I usually put them in their own files.

Class variables are a bit trickier, I usually use a COMMON block with a special name defined in the math__define routine.

-Mike
