Subject: Re: What about real polymorphism ??
Posted by tam on Thu, 09 Dec 2004 20:11:19 GMT
View Forum Message <> Reply to Message

>
> Thats ok. But you know that your object "is a" WOMEN or a MAN because:
> 1) you have created it "o=obj_new('WOMAN')" and then you know its
> methods or
> 2) you request its type with OBJ_ISA() funtion and then need to know it
> methos to invoque one.
>
>

I suspect that the real difference between IDL and Java is that in Java
you need to deal with two different classes/types: In Java if you have some object O,
you have the actual class of O, but you also have the type associated
with the variable that references O, i.e., you can say:

```
    Superclass o = new Subclass();
```

which creates an object that has an actual class of 'Subclass' but is referenced
as if it were an object of type 'Superclass'. Once you have this distinction
a language needs to have rules as to whether the methods that will be invoked
with a reference of the form
     o.method() [or in IDL o->method()]
is the method associated with the Superclass or the Subclass. In Java it can be
either depending upon the details of the method.

In IDL (similarly in Smalltalk or Perl)
you don't have this distinction, since there are no typed variables.
The only class you need to worry about is the class of the object itself and
everything works polymorphically by default.

Of course occasionally you want to be able to explicitly use a superclass's methods
even though a subclass overrides it (e.g., when adding just a little functionality
to the method). In Java you do this with the 'super' prefix. In IDL you use
the name of the superclass.

```
    o->Superclass::method()
```

Hope this clarifies things a bit.

Regards,
Tom McGlynn