Subject: Need advice about inheritance
Posted by wcramer on Sat, 21 May 2005 15:03:52 GMT
View Forum Message <> Reply to Message

I have two classes:

"data": maintains a list of x, y data values and provide methods for
operations between two "data" classes (i.e., "multiply", "add", etc.)

"timedata": inherits "data" and treats x as time and y as value data
------

Methods such as "add" should return another object instance of the same
class as "self". Is there a way to get "add" to return a "timedata"
object without having to copy the method verbatim from the "data" class
to the "timedata" class? The functionality is exactly the same, so I
really don't want to do that. Also, I can't dynamically define the
class name for my call to obj_new because the init parameters are
different.

Any advice would be greatly appreciated.

Thanks,
Doug

Subject: Re: Need advice about inheritance
Posted by David Fanning on Sat, 21 May 2005 23:46:06 GMT
View Forum Message <> Reply to Message

wcramer writes:

> I have two classes:
>
> ------
> "data": maintains a list of x, y data values and provide methods for
> operations between two "data" classes (i.e., "multiply", "add", etc.)
>
> "timedata": inherits "data" and treats x as time and y as value data
> ------
>
> Methods such as "add" should return another object instance of the same
> class as "self". Is there a way to get "add" to return a "timedata"
> object without having to copy the method verbatim from the "data" class
> to the "timedata" class? The functionality is exactly the same, so I
> really don't want to do that. Also, I can't dynamically define the
> class name for my call to obj_new because the init parameters are

> different.

Maybe I'm missing something, but if you want to copy the code
"verbatim", and the functionality is "exactly the same", can
it also be true that "the init parameters are different"!?
I'm confused. :-(

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Need advice about inheritance
Posted by Antonio Santiago on Mon, 23 May 2005 06:28:08 GMT
View Forum Message <> Reply to Message

Before programming you need to stop and thing about your classes,
inheritance and relations.

> Methods such as "add" should return another object instance of the
> same class as "self".

Only for curiosity. ADD returns a new instance or the same instance with
the addition?

> Is there a way to get "add" to return a "timedata"
> object without having to copy the method verbatim from the "data"
> class to the "timedata" class?

Yes, don't write an "ADD" method for TIMEDATA and use the inherited from
DATA.

> The functionality is exactly the same, so I really don't want to do
> that.

what? Perhaps it hasn't got the same functionality. (Perhaps here, my
poor english doesn't bring me to use the correct words). That is, maybe
the concept is the same (the addition) but it is not the same integer
addition than "house" (objects) addition.

> Also, I can't dynamically define the class name for my call to obj_new
> because the init parameters are different.

Perhaps you can use keyword parameters in addition to some kind of
"template pattern". That is, you need your DATA::Init and TIMEDATA::Init
has the same positional parameters in addition to their respective
keyword parameters.
Now, you can call OBJ_NEW('DATA', d1) or OBJ_NEW('TIMEDATA', d1, XX=d2)
because they have the same parameters.

I don't know if this can be hopeful for you.
Bye.

--

Antonio Santiago Pï¿½rez
(email: santiago<<at>>grahi.upc.edu)
(www: http://www.grahi.upc.edu/santiago)
(www: http://asantiago.blogsite.org)

GRAHI - Grup de Recerca Aplicada en Hidrometeorologia
Universitat Politï¿½cnica de Catalunya

Subject: Re: Need advice about inheritance
Posted by wcramer on Mon, 23 May 2005 15:00:18 GMT
View Forum Message <> Reply to Message

I DON'T want to copy the code verbatim.

For the "data" class, all parameters have names that include "X" and
"Y" (like "MAXX", "XVALUES", etc.). For the "timedata" class, I
overwrite many functions to modify the parameter names to include
"TIME" and "VALUE" (like "MAXTIME", "TIMES", etc.). That's why the init
parameters are different. Is there a better way to handle this?

Thanks,
Doug

Subject: Re: Need advice about inheritance
Posted by wcramer on Mon, 23 May 2005 15:13:58 GMT
View Forum Message <> Reply to Message

>> Only for curiosity. ADD returns a new instance or the same instance
with
the addition?
>> Yes, don't write an "ADD" method for TIMEDATA and use the inherited

from
DATA.

ADD returns a new instance. I use OBJ_NEW("data",...) to create the new instance. How do I get the ADD method of the DATA class to return a TIMEDATA without copying the code to the TIMEDATA class and changing the OBJ_NEW call? Is it proper to get the current class type string (OBJ_CLASS(self)) and use that in the OBJ_NEW call, or is there a better way?

>> Perhaps you can use keyword parameters in addition to some kind of "template pattern". That is, you need your DATA::Init and TIMEDATA::Init has the same positional parameters in addition to their respective keyword parameters. Now, you can call OBJ_NEW('DATA', d1) or OBJ_NEW('TIMEDATA', d1, XX=d2) because they have the same parameters.

Althought this would work, I was hoping that I could set up the TIMEDATA class such that a user wouldn't have to know anything about "X" and "Y" and would only deal with "TIME" and "VALUE" (even though the data is stored the same). I suppose that I'm looking for a perfect solution that doesn't exist.

Thanks,
Doug

Subject: Re: Need advice about inheritance
Posted by Antonio Santiago on Mon, 23 May 2005 15:34:05 GMT
View Forum Message <> Reply to Message

> ADD returns a new instance. I use OBJ_NEW("data",...) to create the new
> instance. How do I get the ADD method of the DATA class to return a
> TIMEDATA without copying the code to the TIMEDATA class and changing
> the OBJ_NEW call? Is it proper to get the current class type string
> (OBJ_CLASS(self)) and use that in the OBJ_NEW call, or is there a
> better way?
>
Sorry, but I don't understand well (perhaps because I am a bad english reader).

I suposse when you want to create a new object you know what type of object create, no? Then if you want to create a TIMEDATA object the call is:
OBJ_NEW('TimeData', x, y)

Why do you need to create an OBJ_NEW('Data', ...) to create an TIMEDATA?? On a DATA object you can't invoque TIME operation, this is responsability of TIMEDATA class.

Inside TIMEDATA class (I suposse) you invoke the DATA::Init method, also I suposse the DATA class is more closely to be an storage site (stores x, y) where its subclases retrieves the x, y values and interprets on its own way.

Then the TIMEDATA::Add method gets x, y values and makes a time addition. If the code of TIMEDATA::Add is the same as the DATA:Add then you dont need to write the code, simply when you invoke TIMEDATA::Add really you are invoqing the DATA:Add method.

> For the "data" class, all parameters have names that include "X" and
> "Y" (like "MAXX", "XVALUES", etc.). For the "timedata" class, I
> overwrite many functions to modify the parameter names to include
> "TIME" and "VALUE" (like "MAXTIME", "TIMES", etc.). That's why the
> init parameters are different. Is there a better way to handle this?

Finally, in my own opinion, you shouln't put this names to your methods. When you creates a TIMEDATA object you known this is related with time and then Add method makes an addition with time values. This notation can complicate your code when use polymorphym (I think this is what you want with you ADD method in DATA and TIMEDATA classes).

Bye Mr. W. :)
(from Mr. A)

--

Antonio Santiago Pï¿½rez
(email: santiago<<at>>grahi.upc.edu)
(www: http://www.grahi.upc.edu/santiago)
(www: http://asantiago.blogsite.org)

GRAHI - Grup de Recerca Aplicada en Hidrometeorologia
Universitat Politï¿½cnica de Catalunya

Subject: Re: Need advice about inheritance
Posted by wcramer on Tue, 07 Jun 2005 17:48:36 GMT
View Forum Message <> Reply to Message

Sorry that it's taken me so long to respond to your post, "Mr. A"; I got sidetracked with something else.

Let's do this with code so it will be more clear. Here's what I have:

```
function data::add,data2
  ...
  result = obj_new('data', <data init params>)
  ...
  return, result
end
```

I want this function to return a TIMEDATA object if called with a TIMEDATA object. There are two ways that I can think of to do this:

(1) Override the ADD function in the TIMEDATA class:

```
function timedata::add,timedata2
  ...
  temp = data::add(timedata2)
  result = obj_new('timedata', data = temp)
  ...
  return, result
end
```

I really don't want to do this.

(2) Dynamically pass the class type to OBJ_NEW:

```
function data::add,data2
  ...
  result = obj_new(obj_class(self), <common data/timedata init params>)
  ...
  return, result
end
```

The problem with this method is that I define the INIT parameters as keyword parameters, and the parameter names are different for DATA and TIMEDATA. I can't convert the parameters to positional parameters because INIT allows multiple possibilities of initialization (MATRIX, SIZE, or X/TIMES and Y/VALUES)

Is there a better way to do this that I haven't thought of?

Thanks,
Doug

Subject: Re: Need advice about inheritance
Posted by marc schellens[1] on Fri, 10 Jun 2005 08:06:06 GMT
View Forum Message <> Reply to Message

You *could* use

function data::add,data2

;; common code

if OBJ_CLASS(self) eq 'DATA' then begin

;; do everything for DATA ADD

endif else begin

;; do everything for TIMEDATA ADD

endelse

end

But this is goes again the concept of inheritance. Overloading ADD is
the much cleaner
way. Instead you should factorize you code, ie. define a
DATA::ADD_COMMON and put in there everything both ADD functions share
and
call it from both ADD functions leaving only the differnces there.

Cheers,
marc

Subject: Re: Need advice about inheritance
Posted by Antonio Santiago on Fri, 10 Jun 2005 09:21:38 GMT
View Forum Message <> Reply to Message

Like Marc I think the best solution is to override the add method. As he
says you can use the template pattern, that is, put the common code into
an auxiliar method.

Anyway, I think it is more a concept problem. Supose the class PERSON
and subclasses MAN and WOMAN. It is a bit strange to call:

man = some_person->set_address(xxx)

it must be

person = some_person->set_addres(xxx)

Perhaps this example isn't the best, but applied on your case (and in my
own opinion) is a bit strange to add two datas and obtain a time_data.
That is if you add two integers is a bit strange get a float as result

(possible but strange). You can is something like a cast after an operation.

I don't know how you model the problem, I supose something like:

DATA <----- TIMEDATA

Note that a TIEMDATA object also is a DATA object, but a DATA object maybe isn't a TIMEDATA object. A method on TIMEDATA can return a reference to TIMEDATA object, that in fact is a DATA object, but is dangerous (i think so) to execute a method on DATA and return a TIMEDATA object.

Leaving theory on a side (but remember we must follow it), why not implement a "clone" method in both classes and:

```
FUNCTION Data::Add, some_data_or_timedata_object

 new_object = some_data_or_timedata_object->Clone()

 self->GetProperty, MONTH=m1, DAY=d1, YEAR=y1
 new_object->GetProperty, MONTH=m2, DAY=d2, YEAR=y2

 new_object->SetProperty, MONTH=m1+m2, DAY=d1+d2, YEAR=y1+y2

 RETURN, new_object
END
```

In this case, you can pass to Data::Add any DATA or TIMEDATA oboject and you return the same type of object. The pproblem becomes with TIMEDATA class. Has it got an Add method? If yes, which type of object you can pass DATA or TIMEDATA. Depends of its types TIMEDATA::Add method must work different (getting the properties of TIMEDATA plus theb month, day and year).

If you can model your problem first (UML) and all is ok, the only (sometimes not :)) final task is to imlpement it.

I hope this crazy email will be useful for you, bye :)

--

Antonio Santiago Pï¿½rez
(email: santiago<<at>>grahi.upc.edu)
(www: http://www.grahi.upc.edu/santiago)
(www: http://asantiago.blogsite.org)

GRAHI - Grup de Recerca Aplicada en Hidrometeorologia
Universitat Politï¿½cnica de Catalunya

Subject: Re: Need advice about inheritance
Posted by Michael Wallace on Fri, 10 Jun 2005 18:46:35 GMT
View Forum Message <> Reply to Message

wcramer wrote:
> I DON'T want to copy the code verbatim.
>
> For the "data" class, all parameters have names that include "X" and
> "Y" (like "MAXX", "XVALUES", etc.). For the "timedata" class, I
> overwrite many functions to modify the parameter names to include
> "TIME" and "VALUE" (like "MAXTIME", "TIMES", etc.). That's why the init
> parameters are different. Is there a better way to handle this?
>

I don't know that I really understand what you're trying to do. If I happen to be really off base, just ignore me. Everyone else does. ;-)

Why go to the trouble of overwriting your parameter names? One of the nice things about OO programming is that you can use the same interface among many classes in a class hierarchy. This allows easy switching between data, timedata, foodata and bardata without the need of changing substantial areas of code or at least not needing to code special cases all the time. Also, there is only one core interface to learn. While the subclasses may and probably will have their own implementations of methods and attach their own meanings to certain properties, the core interface of the classes is the same.

It seems that instead of extending the data interface, you're trying to redefine the interface. If the interface of data and timedata is the same, you should be able to do what you want to do (dynamically determine the class and return a new object of that class). By redefining the interface, you're just causing more problems for yourself. It seems that the only reason you're doing this is to give more precise names to the inputs and keywords. By doing that, you're uncoupling interfaces that should be related and consequently running into problems because the interfaces don't match.

Is there anything else preventing you from having timedata and data use the same interface?

-Mike