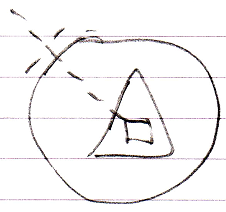
|  |
| --- |
| Circle Language Spec: Inheritance |

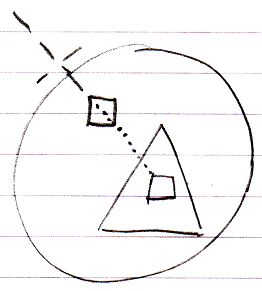
## Shadowing

A derived class can *shadow* a command of the base class. This means that when you call a base command through the derived object, the derived command is run instead.

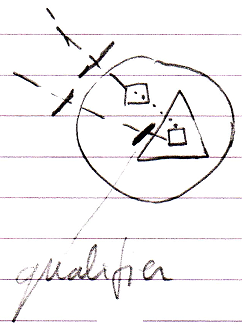
Below you see a normal base member and a call to it:



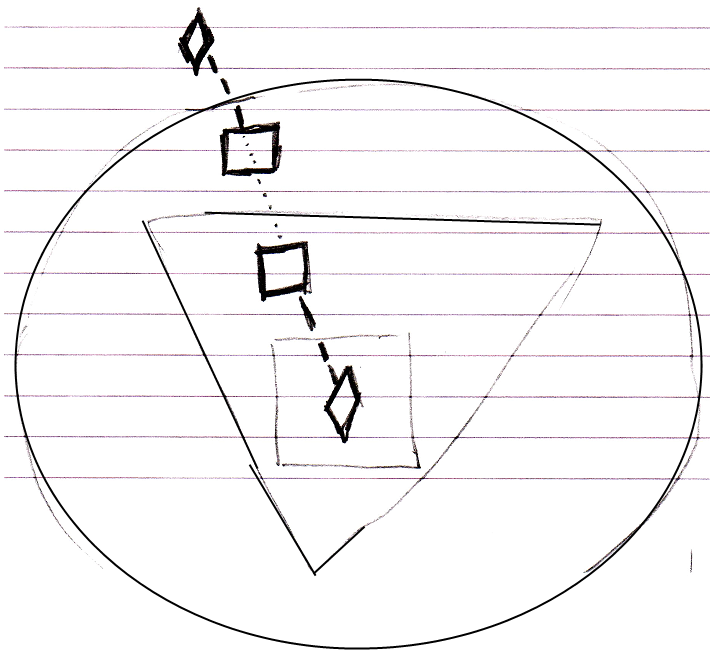
Below you see the situation when the base command is shadowed:



You can still call the base command, by qualifying the call so that the interface is adressed directly:



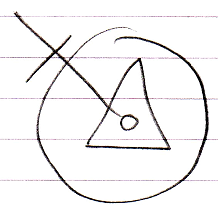
When the base class itself calls the shadowed command, the original base command is called.



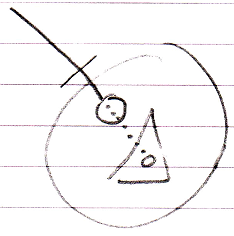
This is where the behavior of *shadowing* is different from *overriding*.

What should be mentioned is that you can also shadow objects, just like you can shadow commands.

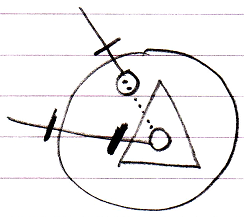
*Unshadowed:*



*Shadowed:*



*Access to base member through qualifier:*



*Base class calls base member, not derived member:*

