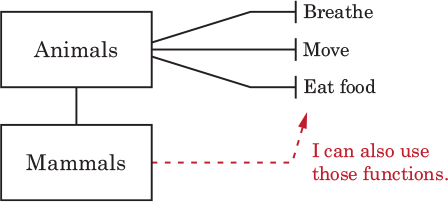
Desconocido

### Funciones heredadas

Puedes darte cuenta de que quien acabe trabajando en la clase Giraffe tiene suerte, porque cualquier función creada por las personas que trabajan en las clases Animal y Mammal también puede ser utilizada por la clase Giraffe . La clase Giraffe *hereda* funciones de la clase Mammal , que, a su vez, hereda funciones de la clase Animal . En otras palabras, cuando creamos un objeto Giraffe , podemos utilizar funciones definidas en la clase Giraffe , así como funciones definidas en las clases Mammal y Animal . Y, del mismo modo, si creamos un objeto Mammal , podemos utilizar funciones definidas en la clase Mammal , así como en su clase padre, Animal .

Observa de nuevo las relaciones entre las clases Animal , Mammal y Giraffe . La clase Animal es la clase padre de la clase Mammal , y Mammal es la clase padre de la clase Giraffe ( [Figura 8-5](ch08.xhtml#ch08fig05) ).



*Figura 8-5: Clases y funciones heredadas*

Aunque reginald sea un objeto de la clase Giraffe , podemos llamar a la función move que definimos en la clase Animal , porque las funciones definidas en cualquier clase padre están disponibles para sus clases hijas:

>>> reginald = Giraffe()  
>>> reginald.move()  
moving

De hecho, todas las funciones que definimos en las clases Animal y Mammal pueden llamarse desde nuestro objeto reginald porque las funciones son heredadas:

>>> reginald = Giraffe()  
>>> reginald.breathe()  
breathing  
>>> reginald.eat\_food()  
eating food  
>>> reginald.feed\_young\_with\_milk()  
feeding young

En este código, creamos un objeto de la clase Giraffe llamado reginald . Cuando llamamos a cada función, imprime un mensaje independientemente de si la función está definida en Giraffe o en una clase padre.

[anterior](ch08_9.html)[Subtema 10 de 14: (Ver todo)](ch08.html)[siguiente](ch08_11.html)