Desconocido

### Añadir características de clase

Considera las clases hijas de la clase Animate que definimos en la sección Hijos y padres. Podemos añadir características a cada clase que describan lo que es y lo que puede hacer. Una *característica* es un rasgo que comparten todos los miembros de la clase (y sus hijos).

Por ejemplo, ¿qué tienen en común todos los animales? Para empezar, respiran, se mueven y comen. ¿Y los mamíferos? Los mamíferos alimentan a sus crías con leche, y también respiran, se mueven y comen. Sabemos que las jirafas comen hojas de lo alto de los árboles. Y como todos los mamíferos, alimentan a sus crías con leche, respiran, se mueven y comen. Cuando añadimos estas características a nuestro diagrama de árbol, obtenemos algo como [la Figura 8-2 .](ch08.xhtml#ch08fig02)

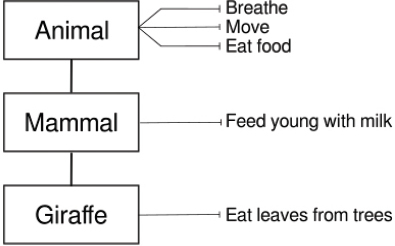


Figura 8-2*: Clases con* características

Estas características pueden considerarse acciones o funciones, cosas que puede hacer un objeto de esa clase.

Para añadir una función a una clase, utilizamos la palabra clave def . Así, la clase Animal tendrá este aspecto:

>>> class Animal(Animate):  
 def breathe(self):  
 pass  
 def move(self):  
 pass  
 def eat\_food(self):  
 pass

En la primera línea de este listado, definimos la clase como antes, pero en lugar de utilizar pass en la línea siguiente, definimos una función llamada breathe y le damos el parámetro self . El parámetro self es una forma de que una función de la clase llame a otra función de la clase (y de la clase padre). Veremos cómo se utiliza este parámetro más adelante.



En la línea siguiente, pass indica a Python que no vamos a dar más información sobre la función breathe porque no va a hacer nada por ahora. A continuación, añadimos las funciones move y eat\_food y utilizamos la palabra clave pass para ambas. En breve recrearemos nuestras clases y pondremos código adecuado en las funciones. Esta es una forma habitual de desarrollar programas.

**NOTA**

*A menudo, los programadores crean clases con funciones que no hacen nada, como forma de averiguar qué debe hacer la clase antes de entrar en los detalles de las funciones individuales.*

También podemos añadir funciones a las clases Mammal y Giraffe . Cada clase podrá utilizar las características (o funciones) de su clase madre, lo que significa que no necesitas crear una clase realmente complicada. En su lugar, puedes poner tus funciones en la clase padre más alta en la que se aplique la característica. (Esto hace que tus clases sean más sencillas y fáciles de entender).

>>> class Mammal(Animal):  
 def feed\_young\_with\_milk(self):  
 pass  
  
>>> class Giraffe(Mammal):  
 def eat\_leaves\_from\_trees(self):  
 pass

En el código anterior, la clase Mammal proporciona una función feed \_young\_with\_milk . La clase Giraffe es una clase hija (o *subclase* ) de Mammal y proporciona otra función: eat\_leaves\_from\_trees .

[anterior](ch08_5.html)[Subtema 6 de 14: (Ver todo)](ch08.html)[siguiente](ch08_7.html)