Andrés Felipe Guerra Amaris.
cc.1097783416
anguerraa@unal.edu.co

Respuestas ejercicio 2 taller 5: Python

1. Si deseo modificar el mensaje del método ruido al crear un objeto Pajaro sin alterar la clase Animal ¿Qué debo agregarle al código? (Por ejemplo, al llamar el método ruido imprima, cantar y silbar).
   R/ Le debo agregar un método ruido a la clase Pajaro que sobrescriba al de Animal, algo asi:
   def ruido (self):
   return "cantar y silbar"

2. Si se crea una nueva clase Pez, y no se definen nuevos métodos, constructor y atributos. ¿Qué constructor tendrá esta clase, qué argumentos recibe? ¿Qué otros métodos y atributos tendrán estos mismos?
R/ si esta nueva clase Pez se declara como una subclase de Animal, todo lo heredará de la misma, incluyendo el constructor, ya que al ser un método común en Python, se hereda como cualquier otro, recibiendo los mismos argumentos. Los métodos y atributos son los mismos de la clase Animal.

3. ¿Qué ocurre con el atributo nombre y edad de la clase SerVivo, al momento de definirse en la Clase Animal? ¿Cómo cambiaría el código del constructor para que estos atributos sean el mismo?
R/ nombre y edad se están sobrescribiendo en la clase Animal. Para que ambos atributos sean el mismo, al constructor de animal se le cambia la asignación de nombre y edad y se pasan estos parámetros al constructor de SerVivo, así:

```
class Animal(SerVivo):
    _totalCreados = 0
    def __init__ (self, nombre, edad, raza):
        super().__init__(nombre,edad)
        self._raza = raza
        Animal._totalCreados += 1
```

4. En la clase Animal se sobrescribieron los métodos setNombre y getNombre, ¿Cómo modificaría estos métodos para que su funcionamiento no oculte algún valor de la clase padre? ¿Podría plantearse esta solución usando super()?
R/ Sí, usando el super() se pueden acceder a los métodos de la clase padre sin sobrescribirlos. En el caso del getter y setter de nombre, sería algo así:

```
def setNombre(self, nombre):
    super().setNombre(nombre)

def getNombre(self):
    super().getNombre
```

Andrés Felipe Guerra Amaris.
cc.1097783416
anguerraa@unal.edu.co

5. El atributo totalCreados de la clase SerVivo ¿es heredado por las demás clases? ¿En este caso ocurre ocultación de los atributos al definirlo de nuevo en las clases hijas?
R/ Sí, este atributo si se hereda por las subclases de SerVivo y Animal. Al definirse de nuevo en las clases hijas, sí ocurre una ocultación del valor del atributo.

6. ¿Los métodos getTotalCreados sobrescriben al metodo de su padre?
R/ sí, esto se debe a que En el caso de métodos de clase ó estáticos heredados, cuando se sobreescribe el método, el viejo método es sobreescrito y no ocultado como en JAVA.

7. ¿Qué métodos hereda una clase que hereda de la clase Persona?
R/ Heredará los métodos que Persona hereda de SerVivo, estos son, los getters y setters de nombre y edad y el getTotalCreados, los métodos que Persona hereda de la clase Object como el __str__, y los métodos de la propia clase Persona, es decir, el aduenarAnimal

8. ¿Qué tipo de objetos podrían pasársele al método aduenarAnimal de la clase Persona?¿Se le puede pasar un objeto serVivo?
R/ Solo se le pueden pasar objetos de tipo Animal, o sea, de la clase Animal o cualquiera de sus subclases, ya que son las que tienen el método ruido() definido. Si se le pasa un objeto SerVivo, al no tener el método ruido definido, se produce un error.
AttributeError: 'SerVivo' object has no attribute 'ruido'

9. ¿Cómo podría obtener la cantidad de seres vivos que se creen (**objetos creados**)?
Si dentro de las subclases de SerVivo y Animal el inicializador invoca al de su clase padre usando el super(), todos llegarán al constructor de SerVivo para definir sus atributos nombre y edad, y a su vez, cada vez que se llama este inicializador, se aumenta el contador de SerVivo, por lo que para acceder a los objetos creados de esta clase, basta con hacer uso del método getTotalCreados() de la clase SerVivo.

10. Si se define el siguiente método en la clase Perro:
```
def getRaza(self, tipo):
return self._raza + " " + tipo
```
¿Qué ocurre con el método getRaza de la clase Animal? ¿Este método se sobreescribe o se sobrecarga?
R/ El método se sobrescribe. En Python no existe la sobrecarga.