PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

Preguntas de análisis

1. Si se desea modificar el mensaje del método **ruido** al crear un objeto **Pajaro** sin alterar la clase **Animal** se debe agregar al código en la clase Pajaro:
   def ruido(self):
   print("**cantar y silbar"**)
   Para de esa manera sobreescribir el método ruido desde Pajaro y que no se tenga que afectar la clase Animal

2. Si se crea una nueva clase Pez, y no se definen nuevos métodos, constructor y atributos. la clase tendrá el inicializador que hace las veces de constructor en python (__init__) de la clase Animal, que recibe 3 argumentos: nombre, edad y raza, tendrá también los atributos get y set para raza y nombre que hereda de la clase animal, y para edad que no está implícito en animal, pero que esté a su vez hereda de ser vivo, así mismo los métodos caminar, correr y ruido, y finalmente, tanto el atributo de clase como el método de clase totalCreados y getTotalCreados.

3. Lo que ocurre con el atributo nombre y edad de la clase SerVivo, al momento de definirse en la Clase Animal es que se redefinen ambos atributos para la clase hijo, para que estos atributos sean el mismo, cambiaría el código del constructor de manera que en el inicializador de Animal se llame el __init__ de la clase padre para que los atributos sean los mismos en ambas clases, así:
   super().__init__(nombre, edad)

4. En la clase Animal se sobreescriben los métodos setNombre y getNombre, se podría modificar estos métodos para que su funcionamiento no oculte algún valor de la clase padre al utilizar super().getNombre desde el método getNombre de la clase hijo y haciendo referencia al método de la clase padre, de igual manera para setNombre.

5. El atributo totalCreados de la clase SerVivo es heredado por las demás clases hijas al ser un atributo propio de la clase, en este caso NO ocurre ocultación de los atributos al definirlo de nuevo en las clases hijas, ya que esto no sucede en el contexto de python, simplemente el atributo se sobreescribe por la nueva clase.

6. Los métodos getTotalCreados sobrescriben al metodo de su padre, sucede una redefinición de dicho método de clase.

7. Los métodos que hereda una clase que hereda de la clase Persona serían: el metodo de inicialización __init__, get y set para nombre y edad, aduenarAnimal y getTotalCreados que es un método de clase.

8. El tipo de objetos podrían pasársele al método aduenarAnimal de la clase Persona tiene que ser de tipo animal, ya que en el método se hace llamado al método ruido del objeto ingresado en el parámetro, método que está definido únicamente para la clase Animal y sus subclases, no se le puede pasar un objeto serVivo por lo mismo mencionado anteriormente, este no cuenta con el método ruido lo que generaría un error de compilación.

9. Se podría obtener la cantidad de seres vivos que se creen (**objetos creados**) al hacer uso del método de clase obtenerTotalCreados de la clase serVivo

10. Si se define el siguiente método en la clase Perro:

```
def getRaza(self, tipo):
    return self._raza + " " + tipo
```

El método getRaza de la clase Animal se sobreescribe únicamente para la clase Perro.