PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

Preguntas de análisis

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**).

   Se debe sobrescribir el método ruido() en la clase Pajaro
   def ruido(self):
   print("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?

   La clase Pez hereda el constructor de la clase Animal, el cual recibe los argumentos nombre, edad y raza, en general todos los métodos y atributos 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?

   Los atributos nombre y edad de la clase SerVivo son redefinidos en la clase Animal cuando se crea un objeto de tipo Animal, se debe eliminar la privacidad de estos atributos nombre y edad tanto en la clase SerVivo como en la clase Animal

```
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, ¿Como modificaría estos métodos para que su funcionamiento no oculte algún valor de la clase padre? ¿Podría plantearse esta solución usando super()?
   Sí se podría, con super() se llamarían los métodos setNombre y getNombre

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

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

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?
   Sí se hereda, pero al redefinirse en las subclases se oculta el atributo de la clase padre. Entonces cada subclase mantiene su propio contador independiente

6. ¿Los métodos getTotalCreados sobrescriben al metodo de su padre?

Sí, los métodos getTotalCreados definidos en las clases hijas sobrescriben el método de la clase padre (SerVivo) porque tienen el mismo nombre

7. ¿Qué métodos hereda una clase que hereda de la clase Persona?

aduenarAnimal, getTotalCreados y los métodos set y get correspondientes a los atributos privados nombre y edad.

8. ¿Qué tipo de objetos podrían pasársele al método aduenarAnimal de la clase Persona?¿Se le puede pasar un objeto serVivo?

Todo objeto que sea de tipo Animal o sus subclases

9. ¿Cómo podría obtener la cantidad de seres vivos que se creen (**objetos creados**)?

Llamando al método getTotalCreados de la clase SerVivo:
totalSeresVivos = SerVivo.getTotalCreados()

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?

El método getRaza en la clase Perro sobrescribe al método con el mismo nombre en la clase Animal porque tienen el mismo nombre y firma y se prioriza el de la clase hija