

RESPUESTAS 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).

En la clase Pajaro redefinir el método ruido con lo que se quiere imprimir:

```
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?

Suponiendo que Pez hereda de Animal, este tendrá todos los métodos (incluyendo __init__) y atributos de su clase padre.

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?

Al no llamar al inicializador de la clase padre en la clase Animal, ningún atributo es asignado por la clase padre, para que sean los mismos, se deberá llamar el inicializador de la clase padre en la clase Animal para que este asigne los atributos.

```
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()?

De la siguiente forma:

```
def setNombre(self, 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?

En teoría no debería ser heredado ya que se pretende que sea un atributo privado con el guion bajo al inicio, pero como en Python no existe este tipo de acceso, si se hereda en las demás clases hijas, en sus clases hijas lo que ocurre es un ocultamiento al volver a definir el atributo.

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

Si, a diferencia de Java, los métodos de clase, son sobrescritos y no ocultados como en Java.

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

Hereda todos los métodos (incluyendo el inicializador) de su clase padre (Persona) que a su vez hereda todos los métodos de SerVivo, que a su vez hereda los métodos de la clase Object.

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

Se le puede pasar como parámetro cualquier objeto de tipo `Animal` o de alguna subclase de ella misma, no se le podrá pasar un objeto `SerVivo` ya que este no tiene definido el método `ruido`, por lo que saldrá un error.

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

Para obtener la cantidad de seres vivos creados, podemos hacerlo mediante el método de clase `getTotalCreados` de la clase `SerVivo`, que este accede al atributo `_totalCreados` que aumenta cada vez que se crea un objeto de tipo `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 sobrescribe o se sobrecarga?

Se haría una sobreescritura, en este caso el método de la clase hija opaca el método de la clase padre.