

Taller 5 python ejercicio 2 Andrés Jacobo Leal Aguirre

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).

Solo debo agregarle esto a la clase Pajaro, ocultando el método anterior.

```
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?

Heredará todo de la clase Animal, tanto su constructor con sus mismos parámetros y sus métodos y atributos

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?

Cuando se definen en la clase Animal se ocultan los de SerVivo, para que esto no pase llamamos al constructos de SerVivo desde el de Animal así

```
def __init__(self, nombre, edad, raza):
```

```
    super().__init__(nombre, edad)
```

```
    self._raza = raza
```

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()?

Si podría 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?

Si, este atributo es accesible desde las clases hijas, si se define este atributo en clases hijas si se oculta

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

Si, pero más que sobrescribirse se ocultan

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

Una clase que hereda de `Persona` tendrá los siguientes métodos:

`__init__`

`aduenarAnimal`

`getTotalCreados`

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 cualquier tipo de objeto que tenga un método `ruido()`, como la clase `SerVivo` no lo tiene, no lo podemos pasar

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

Llamando al método `getTotalCreados()` desde 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 sobrescribe o se sobrecarga

El método se sobrescribe por lo que si llamas ahora al método `getRaza()` desde `Perro` sin parámetros esto generaría un error