

samuel botero rivera

cc: 1000654007

parte 2 Taller 5 python

1)

Es necesario sobrescribir el método ruido dentro de la clase Pajaro.:

```
class Pajaro(Animal):  
    def ruido(self):  
        print("cantar y silbar")
```

2)

La clase Animal cuenta con un constructor que recibe sus atributos y métodos heredados de su clase base.

3)

Al definir nuevamente nombre y edad en Animal, se ocultan los atributos de la clase SerVivo.

Para evitar duplicación y mantener los mismos atributos, puedes usar el constructor de SerVivo con super() dentro de Animal:

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

```
Animal._totalCreados += 1
```

4)

Para que los métodos no oculten el comportamiento de la clase padre, se puede usar `super()` para llamar los métodos correspondientes:

```
class Animal(SerVivo):  
    def setNombre(self, nombre):  
        super().setNombre(nombre)  
  
    def getNombre(self):  
        return super().getNombre()
```

5)

El atributo `_totalCreados` es heredado por las subclases. Sin embargo, si se redefine en las subclases (por ejemplo, en `Animal`, `Gato`, etc.), el atributo de la clase padre queda oculto. De este modo, cada subclase tendrá su propia versión independiente de `_totalCreados`.

6)

Los métodos `getTotalCreados` definidos en las subclases reemplazan al de la clase base (`SerVivo`). Cada subclase utiliza su propia versión de `_totalCreados`.

7)

Una clase que herede de `Persona` heredará todos los métodos y atributos de `Persona`, incluyendo:
`aduenarAnimal`

Métodos y atributos de SerVivo (como setName, getName, _totalCreados).

8)

Se pueden pasar objetos de cualquier clase que implemente el método ruido, ya que aduenarAnimal llama a x.ruido().

No se puede pasar un objeto SerVivo directamente, a menos que SerVivo implemente el método ruido.

9)

Llamando al método getTotalCreados de la clase SerVivo:

```
totalSeresVivos = SerVivo.getTotalCreados()
```

10)

El método getRaza en la subclase sobrescribe al método del mismo nombre en la clase Animal. Sin embargo, al añadir un parámetro adicional (tipo), cambia su firma, lo que lo hace incompatible con las llamadas esperadas por el método de Animal.