



SOLUCIÓN TALLER # _ (PYTHON/JAVA)

ASIGNATURA

Programación Orientada a Objetos

PROFESOR

[Jaime Alberto Guzmán Luna](#)

ESTUDIANTE

SAMUEL TORO AGUDELO

NÚMERO DE IDENTIFICACIÓN

1020107389

Universidad Nacional de Colombia

Sede Medellín

2024



RESPUESTA PREGUNTAS

- a) Se están definiendo tres clases en este ejercicio: Animal, Familia y FamiliaAnimales.
- b) El programa principal se define en la clase FamiliaAnimales.
- c) Se están creando dos objetos de la clase Animal en la clase que define el programa principal.
- d) Los objetos que se están creando de la clase Animal en la clase que define el programa principal son animal1 y animal2.
- e) Los atributos que tiene la clase animal son: nombre, genero, peso, pareja y totalAnimales.
- f) Los atributos de la clase Animal que están haciendo referencia a tipos primitivos son: nombre (String), genero (String), y peso (double).
- g) El atributo de la clase Animal que está haciendo referencia a objetos es pareja (Animal).
- h) El atributo pareja de la clase Animal se inicializa con null.
- i) Antes de la línea 7 en la clase FamiliaAnimales, los objetos animal1 y animal2 no tienen un nombre asignado.
- j) El peso de animal1 en la clase FamiliaAnimales no se especifica, por lo que sería 0.0 por defecto.
- k) Después de establecer como pareja del caballo a la cebra y viceversa, animal1.pareja apunta a animal2 y animal2.pareja apunta a animal1.
- l) El género de animal2 en la clase FamiliaAnimales es "M" (masculino).
- m) Si se comentan las líneas 15 y/o 16 de la clase FamiliaAnimales, los animales no tendrán pareja asignada, por lo que no podrán procrear.
- n) El método procrear se está ejecutando en el contexto del objeto animal1 cuando se invoca en la línea 18 de la clase FamiliaAnimales.



- o) Si al atributo pareja de la clase Animal se le coloca el modificador final, no se podrá cambiar su valor una vez que se haya inicializado.
- p) Sí, se puede eliminar el modificador static del método morir de la clase Animal sin que se afecte el funcionamiento del programa. Sin embargo, se tendría que llamar a este método en un objeto específico de la clase Animal.
- q) Si se modifica la línea 3 de la clase Animal como se indica, todos los animales tendrán el género "M" y no se podrá cambiar.
- r) No es necesario asignarle el valor inicial al atributo totalAnimales de la clase Animal porque los atributos static de tipo int se inicializan a 0 por defecto en Java.
- s) Si reemplazas la línea 28 de la clase FamiliaAnimales por la línea indicada, el resultado se mantiene igual porque totalAnimales es un atributo static y su valor es el mismo para todas las instancias de la clase Animal.
- t) No se afecta el resultado si se reemplaza la línea 27 de la clase FamiliaAnimales por la indicada porque morir es un método static y puede ser llamado desde cualquier objeto de la clase Animal o desde la clase misma.
- u) La clase Animal tiene seis métodos: setPareja, tenerHijo, procrear, morir, toString y el constructor por defecto.
- v) Los métodos procrear() y tenerHijo() de la clase Animal retornan objetos de las clases Familia y Animal, respectivamente.
- w) Al método setPareja() de la clase Animal se le está pasando el parámetro por referencia, ya que Animal es un tipo de objeto.
- x) La variable this de la línea 29 de la clase Animal hace referencia al objeto actual en el que se está ejecutando el método. En este caso, podría omitirse el uso de la variable this ya que no hay ambigüedad con otras variables.
- y) No se afecta la ejecución del programa si se reemplaza la línea 38 de la clase Animal por la indicada porque this hace referencia al objeto actual y es implícito.
- z) Sí, podría eliminarse el modificador static del método morir() de la clase Animal sin que se afecte el programa. Sin embargo, tendría que llamar a este método en un objeto específico de la clase Animal.



aa) La variable `this` de las líneas 44 y 46 de la clase `Animal` hace referencia al objeto actual en el que se está ejecutando el método. Es necesario para diferenciar entre el atributo de la clase y el parámetro del método.

bb) Para modificar el método `imprimir` de la clase `Familia` para que sea un método de clase, tendría que hacer que todos los atributos que usa (`papa`, `mama` y `hijo`) sean también atributos de clase, en dado caso sería así:

```
public class Familia {  
    static Animal papa;  
    static Animal mama;  
    static Animal hijo;  
  
    static void imprimir() {  
        String genero;  
        if (hijo.genero.equals("M"))  
            genero = "masculino";  
        else  
            genero = "femenino";  
        System.out.println(papa.nombre + " es el papá, "  
            + mama.nombre + " es la mamá, y " + hijo.nombre  
            + " es el hijo de genero " + genero);  
    }  
}
```

cc) Es útil que el atributo `totalAnimales` sea un atributo de clase y no un atributo de ejemplar porque representa una propiedad que es común a todos los animales, no solo a un ejemplar específico.

dd) No, no se puede colocar el modificador `static` al método `tenerHijo()` de la clase `Animal` sin que se afecte el funcionamiento del programa. Esto se debe a que este método usa el atributo `genero`, que es un atributo de instancia.

ee) El método `toString()` de la línea 59 de la clase `Animal` retorna una cadena de texto que representa al objeto `Animal`. En este caso, retorna el nombre y el peso del animal.