**Taller #2 / POO**

**a)** ¿Cuántas clases se están definiendo en este ejercicio?

R/ Se están definiendo 3 clases, Animal, Familia y FamiliaAnimales.

**b)** ¿En cuál clase se define el programa principal? Corra el programa principal.

R/ El programa principal se define en la ultima clase “public class FamiliaAnimales”.

**c)** ¿Cuántos objetos de la clase Animal se están creando en la clase que define el programa principal?

R/ En la clase principal se crearon 2 objetos de la clase Animal.

**d)** ¿Cuáles objetos se están creando de la clase Animal en la clase que define el programa principal?

R/ Se están creando “Animal animal1 = new Animal();” y “Animal animal2 = new Animal();”.

**e)** ¿Cuáles atributos tiene la clase Animal?

R/ La clase animal tiene los sigs. atributos: nombre, genero, peso, pareja y totalAnimales.

**f)** ¿Cuáles atributos de la clase Animal están haciendo referencia a tipos primitivos?

R/ Los únicos atributos que son tipo primitivos son peso, ya que es un “double”, y el totalAnimales ya que es un “int”.

**g)** ¿Cuáles atributos de la clase Animal están haciendo referencia a objetos?

R/ Los atributos que están haciendo referencia a objetos son nombre, genero y pareja, ya que son strings.

**h)** ¿Con qué valor se inicializa el atributo pareja de la clase Animal?

R/ Comienza con null, ya que aun no se le ha asignado ningún valor.

**i)** ¿Cuál es el nombre que tienen los objetos animal1 y animal2 antes de la línea 7 en la clase FamiliaAnimales?

R/ Aun no tienen un nombre asignado, son variables existentes, pero sin un valor atribuido.

**j)** ¿Cuál es el peso de animal1 en la clase FamiliaAnimal?

R/ El peso del animal1 en la clase FamiliaAnimal es igual a 0.0 por ser un dato double.

**k)** Dibuje el estado de memoria luego de establecer como pareja del caballo a la cebra y viceversa.

R/

|  |  |
| --- | --- |
| Animal1 | Animal2 |
| Nombre: Cebra | Nombre: Caballo |
| Género: F | Género: M |
| Peso: 0.0 | Peso: 98.0 |
| Pareja: Animal2 | Pareja: Animal1 |
| totalAnimales: 2 | totalAnimales: 2 |

**l)** ¿Cuál es el género del animal2 en la clase FamiliaAnimales?

R/ El genero del animal2 es M.

**m)** Qué sucede si…

- a…se comenta la línea 15 de la clase FamiliaAnimales

- b…se comenta la línea 16 de la clase FamiliaAnimales

- c…se comentan las líneas 15 y 16 de la clase FamiliaAnimales

R/ a) Si se comenta la línea 15 el animal1 no tendría una pareja designada

b) Si se comenta la línea 16 el animal2 no tendría una pareja definida.

c) Si se comentan las líneas 15 y 16, ni el animal1 y el animal2 tendría una pareja asignada.

**n)** ¿En el contexto de cuál objeto se está ejecutando el método procrear cuando es invocado en la línea 18 de la clase FamiliaAnimales?

R/ El objeto que se está ejecutando es animal1, es decir, cebra.

**o)** ¿Qué sucede si al atributo pareja de la clase Animal se le coloca el modificador final?

R/ Luego de agregar el modificador “final” al momento de haberle asignado un valor al atributo no se podrá cambiar a otro valor.

**p)** ¿Se puede eliminar el modificador static del método morir de la clase Animal sin que se afecte el funcionamiento del programa?

R/ Se puede eliminar, pero el método morir pertenecerá a cada instancia individual de la clase Animal en lugar de a la clase en su totalidad, además de esto, para invocar este método se deberá llamar así: animal.morir.

**q)** ¿Qué sucede si modifica la línea 3 de la clase Animal como se indica a continuación?

Línea original: String genero = "M";

Nueva línea: static final String genero = "M";

R/ El género solo podrá tener el string “M” como valor del atributo, además no se podrá cambiar, también será una constante compartida por todas las instancias de la clase “Animal”.

**r)** ¿Por qué no es necesario asignarle el valor inicial al atributo totalAnimales de la clase Animal? Explique.

R/ Porque es un atributo contador que va a determinar el total de animales, por lo cual es un int estático, gracias a eso su valor va a ser 0 por defecto.

**s)** ¿Por qué razón, si se reemplaza la línea 28 de la clase FamiliaAnimales por la línea indicada a continuación, el resultado se mantiene igual?

Línea original: System.out.println("Nuevo total de animales: " + Animal.totalAnimales);

Nueva línea: System.out.println("Nuevo total de animales: " + animal1.totalAnimales);

R/ Da el mismo resultado ya que se esta accediendo al mismo atributo, solo al final es asignado por medio del objeto animal1.

**t)** ¿Por qué razón, no se afecta el resultado, si se reemplaza la línea 27 de la clase FamiliaAnimales por la indicada a continuación?

Línea original: Animal.morir(animal2);

Nueva línea: animal1.morir(animal2);

R/ Al ser un método estático puede ser llamado por medio de la clase o del objeto, con ello las dos interpretaciones son correctas.

**u)** ¿Cuántos métodos tiene la clase Animal?

R/ La clase animal tiene 5 métodos, setPareja, tenerHijo, procrear, morir y toString.

**v)** ¿Cuál es el tipo de retorno de los métodos procrear() y tenerHijo() de la clase Animal?

R/ El método procrear y tenerHijo tienen un tipo de retorno Animal.

**w)** ¿Al método setPareja() de la clase Animal se le está pasando el parámetro por valor o por referencia?

R/ Se esta pasando por valor.

**x)** ¿A quién está haciendo referencia la variable this de la línea 29 de la clase Animal cuando se ejecuta el programa principal? ¿Podría omitirse el uso de la variable this en este caso?

R/ Esta haciendo referencia al atributo nombre, del objeto Animal; y si podría omitirse, ya que no hay otra variable que se identifique igual “nombre”, por lo tanto, no habrá conflicto alguno.

**y)** ¿Por qué no se afecta la ejecución del programa si se reemplaza la línea 38 de la clase Animal por la indicada a continuación?

Línea original: System.out.println(nombre + " y " + pareja.nombre + " van a tener un hijo");

Nueva línea: System.out.println(this.nombre + " y " + this.pareja.nombre + " van a tener un hijo");

R/ No se ve afectado puesto que se esta refiriendo a los mismos atributos de la clase Animal.

**z)** ¿Podría eliminarse el modificador static del método morir() de la clase Animal sin que se afecte el programa?

R/ Solo se vería afectado en la sintaxis, ya que pasaría de ser llamado de forma general en la clase, a vernos en la obligación de llamarlo directamente a una instancia individual, así: “animal1.morir”

**aa)** ¿A quién hace referencia la variable this de las líneas 44 y 46 de la clase Animal cuando se ejecuta el programa principal? ¿Por qué es necesario?

R/ La variable “this” hace referencia al objeto Animal que se esta utilizando para formar una familia, es decir, el papa y la mama; es necesario su uso ya que está citando explícitamente esas dos instancias.

**bb)** Modifique el método imprimir de la clase Familia para que sea un método de clase.

R/ static void imprimir(Familia familia) {

String genero;

if (familia.hijo.genero.equals("M"))

genero = "masculino";

else

genero = "femenino";

Luego de esto, el método se solicitará Familia.imprimir(familia);.

**cc)** ¿Por qué razón es útil que el atributo totalAnimales sea un atributo de clase y no un atributo de ejemplar?

R/ Hace que sea un atributo de clase, se promueve la coherencia, el acceso uniforme y la simplicidad en la gestión del conteo de animales en la aplicación.

**dd)** ¿Se puede colocar el modificador static al método tenerHijo() de la clase Animal sin que se afecte el funcionamiento del programa? Explique.

R/ Afectaría la forma en que se llama el método, además no podría tener acceso directo a las variables de instancia.

**ee)** ¿Qué hace el método toString() de la línea 59 de la clase Animal?

R/ Este método concatena e imprime texto con objetos Animales.