

PRÁCTICA 06 - PROGRAMACIÓN ORIENTADA A OBJETOS

Alejandro Vaquera López

Ejercicio 1: Realiza las clases de Animales, Perros y Gatos. Implemente un programa que cree un objeto para * cada uno de ellos y llame a sus métodos de dormir y comer. ¿Qué líneas corresponden a los métodos * definidos en la clase principal y cuáles a las clases secundarias?

```
38 public class Animales{
    Run | Debug
39     public static void main(String [] args){
40
41         Cat cat = new Cat(name:"Snow", color:"White");
42         Dog dog = new Dog(name:"Popi", color:"Black");
43         cat.sleep();
44         cat.eat();
45         System.out.println();
46         dog.sleep();
47         dog.eat();
48     }
49 }
50
51
```

Ejercicio 2: Cree una variable de clase "Animal" e inicialícela con un objeto "Dog", use todos los métodos * de la clase Animal" y "Dog" y vea lo que sucede cuando llama a: -Un método escrito exclusivamente en el " Clase Animal-La clase "Dog" anula un método.-Y un método exclusivo de la clase "Dog"

```
public class Animales{
    Run | Debug
    public static void main(String [] args){

        Cat cat = new Cat(name:"Snow", color:"White");
        Dog dog = new Dog(name:"Popi", color:"Black");
        cat.sleep();
        cat.eat();
        System.out.println();
        dog.sleep();
        dog.eat();

        //Animal animal = new Animal("leopardo", "gris");
        Animal animal = new Dog(name:"salchicha",color:"cafe");
        animal.sleep();
        animal.eat();
        animal.bark();
    }
}
```

Ejercicio 3: Pruebe el bucle for-each con la matriz de animales.

```
Animal [] animalArray = new Animal[3];
animalArray[0] = new Animal(name:"pepe", color:"gris");
animalArray[1] = new Animal(name:"pipo", color:"cafe");
animalArray[2] = new Animal(name:"papa", color:"negro");

for (Animal a: animalArray){
    a.eat();
}
```

```
comiendo
comiendo
comiendo
```

C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Practica-06>

Ejercicio 4: Pruebe la salida del código anterior.

```
Animal [] animalArray = new Animal[3];
animalArray[0] = new Animal(name:"pepe", color:"gris");
animalArray[1] = new Animal(name:"pipo", color:"cafe");
animalArray[2] = new Dog(name:"papa", color:"negro");

for (Animal a: animalArray){
    a.eat();
    if (a instanceof Dog){
        ((Dog) a).bark();
    }
}
```

```
comiendo
comiendo
comiendo
Barking
```

C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Practica-06>

Desafío:

- Implementar una clase "Zoo" que contenga una ArrayList de diferentes objetos Animal y métodos para agregar, eliminar y mostrar animales.
- Agregue más funcionalidad a la clase de veterinario.
- Crear diferentes subclases de animales con sus propios métodos.

Las clases Animal, Tiger y Zoo están interconectadas a través de una interfaz

```
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Zoologico\Zoologico>javac Animal.java
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Zoologico\Zoologico>javac Tiger.java
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Zoologico\Zoologico>javac Zoo.java
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Zoologico\Zoologico>java Zoo
match
Vengala 4 155 45 White
Vengala was vaccinated
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Zoologico\Zoologico>
```

CRD de animales

```
public class Zoo{
    private List<Animal> animalArray = new ArrayList<>();

    public void addAnimal(Animal animal){
        animalArray.add(animal);
    }

    public void readAnimal(){
        Veterinarian veterinarian = new Veterinarian(name:"roberto", salary:3000);
        for (Animal animal : animalArray){
            System.out.println(animal.getName() + " " + animal.getSize() + " " + animal.getWeight() +
                veterinarian.vaccinate(animal);
        }
    }

    public void deleteAnimal(String nombre){
        int index = 0;
        for (int cont = 0; cont < animalArray.size(); cont++){
            if(nombre == animalArray.get(cont).getName()){
                index = cont;
                System.out.println(x:"match");
            }
        }
        animalArray.remove(index);
    }
}
```

Se adjuntaran todos los archivos de este programa. (favor de ignorar las marcas de error, según una breve investigación, es un bug del editor de código, el programa funciona como se espera).

Práctica 06

```
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Zoologico\Zoologico>javac Animal.java
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Zoologico\Zoologico>javac Tiger.java
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Zoologico\Zoologico>javaC Zoo.java
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Zoologico\Zoologico>java Zoo
roar
match
Vengala 4 155 45 White
Vengala was vaccinated
C:\Users\Vacintosh\Documents\projects\P00-Claudia-tona\Zoologico\Zoologico>
```

1. Se utilizó una lista para rellenarla de objetos de animales
2. Se implementó la clase veterinario y su método de vacunación
3. Se creó la subclase "Tigre" que contiene métodos @override y 1 método propio.
Todo adjunto en los archivos de código fuente.