

Materia:	Programación II		
Nivel:	2º Cuatrimestre		
Tipo de Examen:	Segundo Parcial		
Apellido ⁽¹⁾ :		Fecha:	25 nov 2024
Nombre/s ⁽¹⁾ :		Docente a cargo ⁽²⁾ :	Baus /Quiroz
División ⁽¹⁾ :	121-2	Nota ⁽²⁾ :	
DNI ⁽¹⁾ :		Firma ⁽²⁾ :	

(1) Campos a completar solo por el estudiante en caso de imprimir este enunciado en papel.

(2) Campos a completar solo por el docente en caso de imprimir este enunciado en papel.

Sistema de Gestión de Animales en un Zoológico

Se te pide implementar un sistema de gestión de animales en un zoológico. Cada animal tendrá atributos como id, nombre, especie y alimentacion. Además, se debe permitir realizar operaciones como agregar, obtener, eliminar, filtrar, ordenar, guardar y cargar desde archivos.

Requisitos:

1. Clases involucradas:

- **Animal:** Debe contener los siguientes atributos: id (entero), nombre (String), especie (String) y alimentacion (enum TipoAlimentacion). La clase Animal debe implementar Comparable<Animal> para que se pueda ordenar de manera natural por su id (en orden ascendente).
- **TipoAlimentacion:** Un enum con los tipos de alimentación: CARNIVORO, HERBIVORO, OMNIVORO, INSECTIVORO.
- **Zoologico:** Implementa un inventario genérico que pueda almacenar cualquier tipo de objeto que implemente la interfaz CSVSerializable. El zoológico debe permitir agregar, eliminar, obtener elementos, y filtrar según un criterio. También debe permitir ordenar los animales de manera natural (por id) y a través de un Comparator (por ejemplo, por nombre o especie).
- **CSVSerializable:** Interfaz con el método toCSV() para que las clases que la implementen puedan exportar sus datos en formato CSV.

2. Operaciones del Zoológico:

- Agregar animales al inventario.
- Obtener animales por índice.
- Eliminar animales por índice.
- Filtrar animales según un criterio (como por ejemplo, por tipo de alimentación o especie).

- Ordenar animales:
 - Ordenar de manera natural (por id).
 - Ordenar mediante un Comparador (por ejemplo, por nombre o especie).
 - Guardar el inventario de animales en un archivo binario.
 - Cargar el inventario de animales desde un archivo binario.
 - Guardar el inventario de animales en un archivo CSV.
 - Cargar el inventario de animales desde un archivo CSV.
3. **Filtrado:** Implementa al menos dos filtros:
- Filtrar animales por tipo de alimentación (por ejemplo, mostrar solo los animales CARNIVOROS).
 - Filtrar animales que tengan una palabra en su nombre (por ejemplo, filtrar por el nombre que contenga "León").
4. **Métodos CSV:**
- toCSV() que devuelve los datos del animal en formato CSV.
 - fromCSV(String) que recibe una cadena CSV y devuelve un objeto de tipo Animal (debe ser de clase).
5. **Entrega:** Debes entregar el código de las clases requeridas, incluyendo los métodos de filtrado, ordenamiento y CSV, así como una clase de prueba que verifique el funcionamiento de todas las operaciones del inventario, incluyendo el guardado y carga desde archivos, los filtros y el ordenamiento.

```
public static void main(String[] args) {
    try {
        // Crear un zoológico de animales
        Zoologico<Animal> zoologico = new Zoologico<>();
        zoologico.agregar(new Animal(1, "León", "Panthera leo",
TipoAlimentacion.CARNIVORO));
        zoologico.agregar(new Animal(2, "Elefante", "Loxodonta",
TipoAlimentacion.HERBIVORO));
        zoologico.agregar(new Animal(3, "Oso", "Ursus arctos",
TipoAlimentacion.OMNIVORO));
        zoologico.agregar(new Animal(4, "Zorro", "Vulpes vulpes",
TipoAlimentacion.CARNIVORO));
        zoologico.agregar(new Animal(5, "Gorila", "Gorilla gorilla",
TipoAlimentacion.OMNIVORO));

        // Mostrar todos los animales en el zoológico
        System.out.println("Inventario de animales:");
        zoologico.paraCadaElemento(animal -> System.out.println(animal));

        // Filtrar animales por tipo de alimentación CARNIVORO
        System.out.println("\nAnimales CARNIVOROS:");
        zoologico.filtrar(/*aca va expresión lambda*/)
            .forEach(animal -> System.out.println(animal));

        // Filtrar animales cuyo nombre contiene "León"
        System.out.println("\nAnimales cuyo nombre contiene 'León':");
```

```

zoologico.filtrar((*aca va expresión lambda*))
    .forEach(animal -> System.out.println(animal));

// Ordenar animales de manera natural (por id)
System.out.println("\nAnimales ordenados de manera natural (por id):");
zoologico.ordenar();
zoologico.paraCadaElemento(animal -> System.out.println(animal));

// Ordenar animales por nombre utilizando un Comparator
System.out.println("\nAnimales ordenados por nombre:");
zoologico.ordenar((*aca va expresión lambda*));
zoologico.paraCadaElemento(animal -> System.out.println(animal));

// Guardar el zoológico en un archivo binario
zoologico.guardarEnArchivo("src/data/animales.dat");

// Cargar el zoológico desde el archivo binario
Zoologico<Animal> zoologicoCargado = new Zoologico<>();
zoologicoCargado.cargarDesdeArchivo("src/data/animales.dat");
System.out.println("\nAnimales cargados desde archivo binario:");
zoologicoCargado.paraCadaElemento(animal -> System.out.println(animal));

// Guardar el zoológico en un archivo CSV
zoologico.guardarEnCSV("src/data/animales.csv");

// Cargar el zoológico desde el archivo CSV
zoologicoCargado.cargarDesdeCSV("src/data/animales.csv", linea ->
Animal.fromCSV(linea));
System.out.println("\nAnimales cargados desde archivo CSV:");
zoologicoCargado.paraCadaElemento(animal -> System.out.println(animal));

} catch (IOException | ClassNotFoundException e) {
    System.err.println("Error: " + e.getMessage());
}
}
}

```