Tarea extraordinaria

Programación

Curso DAM 2025/26

Se pretende desarrollar una aplicación Java que gestione los animales de un zoológico. Los animales deben estar clasificados por tipos (mamíferos, aves, reptiles, anfibios y peces), y deben incluir información común y específica de cada tipo. Se creará un menú para permitir agregar, listar y eliminar animales:

--- MENÚ DEL ZOO ---
1. Agregar animal
2. Listar animales
3. Eliminar animal
4. Salir
Opción:

La funcionalidad será la siguiente:

- Al iniciar la aplicación se mostrará el menú propuesto.
- Dependiendo de la opción seleccionada por el usuario:
    - **Agregar animal:** Se creará un nuevo Animal, si los datos introducidos por el usuario son correctos. Se solicitará el nombre del animal, la especie, la fecha de nacimiento (dd/mm/yyyy), el peso (kg), el tipo de animal (mamífero, ave, reptil, anfibio o pez), el continente y una descripción no muy extensa. Cada tipo de especie tiene un atributo particular, que también debe ser informado (se especifica más adelante). Todos los datos serán solicitados por teclado y se validará la entrada correcta de todos ellos. Si algún dato no es correcto, se solicitará de nuevo hasta que sea correcto, sin salir al menú. **Habrá que comprobar que no existe en el zoo un animal con el mismo nombre.** En caso afirmativo, se mostrará un mensaje por pantalla y se volverá al menú.
    - **Listar animales:** Mostrar una lista de animales ordenada por nombre y filtrada por tipo de animal (mamífero, ave, reptil, anfibio, pez). Pon los de tipo reptil en primer lugar.
    - **Eliminar animal:** Permitir eliminar un animal por su nombre.
    - **Salir:** El programa finalizará.

Requisitos de la aplicación:

- Todas las clases dispondrán de los siguientes métodos:
    - **Constructor o constructores.**
    - **Métodos get y set para acceder a sus propiedades.**
    - **Método get_Edad(): Retorna un entero con la edad del animal.**

- No queremos animales con nombres duplicados y además deben mantenerse **ordenados por su nombre en la estructura de datos.** Utiliza un **TreeSet** para conseguir este objetivo.
- Crear una clase **Animal**, que **NO podrá ser instanciada**, con los siguientes atributos:
    - **Nombre** (String)
    - **Especie** (String)
    - **FechaDeNacimiento** (Date)
    - **Peso** (double)
    - **Descripcion** (String)

        - **ContinenteDeOrigen** (Tipo enumerado: África, Asia, Europa, América, Oceanía, Antártida)

- Crear una jerarquía de clases para representar diferentes tipos de animales (con sus particularidades), heredando de la clase Animal:
    - **Mamifero**, con el atributo:
        - **habilidadDeCorrer** (booleano)
    - **Ave**, con el atributo:
        - **habilidadDeVolar** (booleano)
    - **Reptil**, con el atributo:
        - **tieneEscamas** (booleano)
    - **Anfibio**, con el atributo:
        - **capacidadDeRespirarAire** (booleano)
    - **Pez**, con el atributo:
        - **tipoDeAgua** (dulce, salada)

- Los animales deben ser serializados en un archivo (zoo.dat) para persistir la información. Al iniciar el programa, los animales deben ser cargados desde el archivo si ya existen.
- Los animales deben ser guardados en el archivo automáticamente cuando se agregue o elimine uno.

TEN EN CUENTA:

- En la clase **Principal** implementarás toda la lógica del programa. Contendrá el método main y todo el código de interacción con el usuario (lectura de teclado y salida por pantalla).
- En las clases de los animales no deben hacer validaciones de datos. Los datos se validan en la clase **Principal** y si son correctos, se instancian los objetos.
- Será necesario realizar comprobaciones de los tipos de los datos solicitados por teclado.

.

Incluye en el programa comentarios con todas las consideraciones oportunas que se necesiten para entender cómo has realizado la tarea.

Criterios de evaluación:

- **Código correcto:** El sistema debe cumplir con todos los requisitos funcionales y ser capaz de agregar, listar y eliminar animales correctamente.
- **Uso de herencia y polimorfismo:** Se debe implementar correctamente la jerarquía de clases y el comportamiento de los animales según su tipo.
- **Manejo de excepciones:** El programa debe manejar adecuadamente los errores de entrada y salida.
- **Estructuras de datos:** El uso de TreeSet debe garantizar que los animales estén ordenados por nombre y no haya duplicados.
- **Persistencia de datos:** La serialización y deserialización de objetos debe funcionar correctamente, garantizando que los animales sean guardados y cargados entre ejecuciones.

- **Interfaz de usuario:** El menú debe ser claro y fácil de usar, con un manejo adecuado de los errores de entrada.

Sugerencias para el desarrollo:

- Comienza desarrollando las clases básicas de los animales, asegurándote de que la herencia funcione correctamente.
- Implementa el sistema de serialización para guardar y cargar los animales desde un archivo.
- Desarrolla el menú paso a paso, probando cada opción individualmente.
- Asegúrate de que todos los datos ingresados sean validados antes de ser procesados.

Indicaciones de entrega:

Se debe entregar un proyecto (Ant) Netbeans o no será corregido.
El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:
apellido1_apellido2_nombre_TareaExtraordinariaProg

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la quinta unidad del MP de PROG**, debería nombrar esta tarea como...

sanchez_manas_begona_ TareaExtraordinariaProg