Jorge Andrés Pardo Rea

1. Introducción

Descripción del Proyecto: El proyecto de "Gestión de Mascotas Virtuales" es una aplicación sencilla en Java que permite gestionar una lista de mascotas. Los usuarios pueden agregar, eliminar y listar mascotas a través de una interfaz de línea de comandos. La aplicación utiliza principios de inyección de dependencias y un enfoque modular para manejar las operaciones de mascotas.

Objetivo: El objetivo es proporcionar una solución básica para gestionar mascotas, demostrando conceptos de programación orientada a objetos y la implementación de un sistema de gestión simple.

2. Estructura del Proyecto

2.1 Clases Principales:

- **Pet**
    - **Descripción:** Representa una mascota con atributos como nombre, edad y tipo.
    - **Atributos:**
        - name: El nombre de la mascota (String).
        - age: La edad de la mascota (int).
        - type: El tipo de la mascota (String).
    - **Métodos:**
        - getName(): Obtiene el nombre de la mascota.
        - getAge(): Obtiene la edad de la mascota.
        - getType(): Obtiene el tipo de la mascota.
        - setName(String name): Establece el nombre de la mascota.
        - setAge(int age): Establece la edad de la mascota.
        - setType(String type): Establece el tipo de la mascota.
        - toString(): Devuelve una representación en texto de los atributos de la mascota.
- **PetService**

    - **Descripción:** Interfaz que define las operaciones para gestionar mascotas.
    - **Métodos:**
        - addPet(Pet pet): Agrega una nueva mascota.
        - removePet(String name): Elimina una mascota por nombre.
        - listPets(): Lista todas las mascotas.

- **PetServiceImpl**
    - **Descripción:** Implementación de la interfaz PetService que maneja la lógica real para gestionar mascotas.
    - **Atributos:**
        - pets: Una lista para almacenar las mascotas.
    - **Métodos:**
        - Implementa los métodos de PetService para agregar, eliminar y listar mascotas.
- **PetManager**
    - **Descripción:** Maneja la interacción con el usuario y utiliza PetService para gestionar las mascotas.
    - **Métodos:**
        - start(): Muestra el menú y maneja la entrada del usuario para realizar acciones con mascotas.
        - main(String[] args): Inicializa PetService y PetManager, y comienza la aplicación.

---

3. Funcionamiento del Programa

3.1 Método start() en PetManager:

- **Descripción:** Proporciona un menú interactivo para que el usuario pueda elegir entre agregar, eliminar, listar mascotas o salir del programa.
- **Flujo:**

- Muestra el menú principal.
- Lee la opción del usuario.
- Procesa la opción seleccionada (agregar, eliminar, listar, salir).
- Repite el proceso hasta que el usuario elija salir.

3.2 Método main() en PetManager:

- **Descripción:** Punto de entrada principal para ejecutar la aplicación.
- **Flujo:**
  - Inicializa PetService con una instancia de PetServiceImpl.
  - Crea una instancia de PetManager pasando el PetService.
  - Llama a start() para comenzar la interacción con el usuario.

---

4. Ejemplos de Uso

Agregar una Mascota:

- El usuario selecciona la opción "Añadir una mascota" del menú.
- Introduce el nombre, edad y tipo de la mascota.
- La mascota se añade a la lista gestionada por PetService.

Eliminar una Mascota:

- El usuario selecciona la opción "Eliminar una mascota" del menú.
- Introduce el nombre de la mascota que desea eliminar.
- La mascota se elimina de la lista gestionada por PetService.

Listar Mascotas:

- El usuario selecciona la opción "Listar todas las mascotas" del menú.
- La aplicación muestra la información de todas las mascotas en la lista, utilizando el método toString() de la clase Pet.

---

5. Consideraciones Adicionales

- **Validaciones:** Se recomienda agregar validaciones adicionales para asegurarse de que las entradas del usuario sean válidas (por ejemplo, que la edad sea un número positivo).
- **Extensibilidad:** El proyecto puede ampliarse para incluir características adicionales, como la persistencia de datos en archivos o bases de datos.

---

6. Conclusión

Este proyecto proporciona una base sólida para la gestión de mascotas utilizando conceptos de programación orientada a objetos y modularidad. La implementación demuestra el uso de inyección de dependencias y ofrece una estructura simple y eficaz para la gestión de objetos en una aplicación de consola.