Ejercicio 3 -Arreglos de Dinámicos C – Alejandro Sagui

Análisis

1. Requisitos que tiene que tener mi sistema:
   1. Registrar dueños de mascotas y poner sus datos
   2. Registrar la mascota de cada dueño y poner su edad y su raza
   3. Registrar sus consultas médicas
   4. Manejar cualquier cantidad de dueños o mascotas con arraylist
   5. Ver reportes como ver que especia es la más atendida la medicina que más uso
2. Clases que tendrá mi programa:
   1. Clase dueño: tendrá información del registro del dueño de las mascotas
   2. Clase mascota: tendrá información del registro de la mascota
   3. Consulta: es la visita al veterinario con información de fecha, tratamiento
   4. Main: acá pondré toda la interacción con el usuario:
   5. Veterinaria: acá se administra el registro de dueños, mascotas y consultas.
3. Propósito de los atributos de cada clase
   1. Dueño
      1. Nombre: nombre del dueño
      2. Teléfono: contacto para las citas
      3. Email: otro método de contacto
      4. Mascotas: será arraylistj<Mascota> con las consultas de las mascotas
   2. Mascota
      1. nombre: identificar al animal.
      2. especie: clasificar (perro, gato, ave, otro animal).
      3. raza: precisar más dentro de la especie.
      4. edad: conocer la etapa de vida del animal.
      5. dueño: dueño responsable.
      6. consultas: lista dinámica (ArrayList<Consulta>) con las consultas de la mascota.
   3. Consulta
      1. fecha: momento en que ocurrió la atención.
      2. motivo: razón de la consulta (enfermedad, control, vacuna)
      3. tratamiento: acciones o medicamentos indicados.
      4. mascota: referencia a la mascota atendida.
   4. Veterinaria
      1. dueños: lista dinámica (ArrayList<Dueño>) con todos los clientes.
      2. mascotas: lista dinámica (ArrayList<Mascota>) con todos los animales registrados.
      3. consultas: lista dinámica (ArrayList<Consulta>) con todas las atenciones.
4. Propósito de los métodos principales
   1. Dueño
      1. agregarMascota(Mascota m): añade una nueva mascota al dueño.
      2. getMascotas(): devuelve la lista de mascotas.
   2. Mascota
      1. agregarConsulta(Consulta c): añade una nueva consulta.
      2. getConsultas(): devuelve todas las consultas.
   3. Consulta
      1. Métodos getters para acceder a sus datos (fecha, motivo, tratamiento).
   4. Veterinaria
      1. registrarDueño(Dueño d): guarda un nuevo dueño.
      2. registrarMascota(Mascota m): guarda una mascota en el programa.
      3. registrarConsulta(Consulta c): guarda una consulta en el sistema.
      4. getEspecieMasAtendida(): calcula qué animal aparece más veces en las consultas.
      5. getMascotaMasConsultada(): devuelve la mascota con mayor número de atenciones.
      6. getPromedioConsultasPorMes(): calcula el promedio mensual.
   5. Principall
      1. main(String[] args): muestra menús, solicita datos y manda llamar los métodos de Veterinaria.

Diseño

Tabla

El contenido generado por IA puede ser incorrecto.