

Ejercicio 5. Polimorfismo vía herencia - Análisis, Diseño & Programa

Requisitos Funcionales

- Registro de Animales:
- Permitir al usuario agregar un nuevo felino o primate y almacenar sus detalles.
- Cálculo de Costos:
- Calcular el costo de mantenimiento mensual de cada animal, incluyendo el costo de alimentación y recinto.
- Verificación de Recintos Adecuados:
- Comprobar si el zoológico tiene las condiciones adecuadas para alojar a un animal específico según el tamaño del recinto.
- Historial de Intercambios:
- Registrar y mostrar el historial de intercambios realizados con otros zoológicos.
- Decisión Basada en Condiciones:
- Verificar si se pueden aceptar nuevos animales según el presupuesto y el costo de construcción del recinto.
- Modificación de Información de Animales:
- Permitir modificar la información almacenada de los animales, tales como nombre científico, hábitat, esperanza de vida, entre otros atributos específicos, sin utilizar Object.

Identificación de Clases, Atributos y Métodos

Clases:

Animal (clase abstracta)

Mamifero (subclase de Animal)

Felino (subclase de Mamifero)

Primate (subclase de Mamifero)

Gestion

DriverProgram

Jerarquía de Clases:

Animal será una clase abstracta, ya que no se crearán instancias de esta clase directamente. Esta clase contendrá atributos generales y métodos abstractos.

Mamifero continuará siendo una subclase de Animal, y seguirá almacenando atributos específicos de los mamíferos.

Felino y Primate seguirán siendo subclases de Mamifero, pero ahora implementarán los métodos abstractos definidos en Animal.

Propósito de las Clases:

Animal

Propósito: Clase abstracta que representa la base para todos los animales en el zoológico. Define las propiedades y comportamientos comunes a todos los animales.

Atributos:

- protected String nombrec: Nombre científico del animal.
- protected String habitat: Descripción del hábitat del animal.
- protected double esperanza: Esperanza de vida en años.

Métodos:

- public abstract double calcularComida(): Método abstracto para calcular la cantidad de comida diaria.
- public String getNombrec(): Retorna el nombre científico del animal.
- public void setNombrec(String nombrec): Asigna el nombre científico del animal.
- public String getHabitat(): Retorna la descripción del hábitat.
- public void setHabitat(String habitat): Asigna el hábitat del animal.
- public double getEsperanza(): Retorna la esperanza de vida.
- public void setEsperanza(double esperanza): Asigna la esperanza de vida.
- public String toString(): Retorna una descripción completa del animal.

- **Mamifero**

Propósito: Clase abstracta que hereda de Animal y define las propiedades y comportamientos específicos de los mamíferos.

Atributos:

- protected int pelaje: Tipo de pelaje (1: corto, 2: largo, 3: grueso, 4: ausente).
- protected int crias: Número promedio de crías por camada.
- protected double peso: Peso del mamífero en kilogramos.
- protected double gestacion: Tiempo de gestación en meses.
- protected boolean extincion: Indica si el mamífero está en peligro de extinción.
- protected int dieta: Tipo de dieta (1: carnívoro, 2: omnívoro).
- protected double costodieta: Costo de la dieta por gramo.
- protected String especie: Nombre de la especie del mamífero.
- protected int recinto: Tamaño del recinto (1: pequeño, 2: mediano, 3: grande).
- protected int tamano: Tamaño del mamífero (1: pequeño, 2: mediano, 3: grande).

Métodos:

- public abstract double costoComida(): Método abstracto para calcular el costo de la comida mensual.
- public abstract double costoTotal(): Método abstracto para calcular el costo total mensual.
- Métodos get y set para todos los atributos.
- public int mantenerRecinto(): Calcula el costo de mantener el recinto.
- public String toString(): Retorna una descripción completa del mamífero.

• Felino

Propósito: Clase que hereda de Mamifero y representa a los felinos en el zoológico.

Atributos:

- private double cola: Longitud de la cola en centímetros.
- private String color: Color del pelaje.
- private double velocidad: Velocidad máxima en metros por segundo.

Métodos:

- public double calcularComida(): Calcula la cantidad de comida diaria para un felino.

- `public double costoComida():` Calcula el costo mensual de la comida para un felino.
- `public double costoTotal():` Calcula el costo total mensual para un felino.
- Métodos `get` y `set` para `cola`, `color` y `velocidad`.
- `public String toString():` Retorna una descripción completa del felino.

- **Primate**

Propósito: Clase que hereda de `Mamifero` y representa a los primates en el zoológico.

Atributos:

- `private int estructura:` Estructura social del primate (1: solitario, 2: grupal, 3: familiar, 4: jerárquico).
- `private int inteligencia:` Nivel de inteligencia del primate (1 a 100).
- `private double cerebro:` Peso del cerebro en gramos.

Métodos:

- `public double calcularComida():` Calcula la cantidad de comida diaria para un primate.
- `public double costoComida():` Calcula el costo mensual de la comida para un primate.
- `public double costoTotal():` Calcula el costo total mensual para un primate.
- Métodos `get` y `set` para `estructura`, `inteligencia` y `cerebro`.
- `public String toString():` Retorna una descripción completa del primate.

- **Gestion**

Propósito: Clase encargada de gestionar los animales en el zoológico, incluyendo agregar, modificar, eliminar y encontrar animales.

Atributos:

- `private ArrayList<Animal> listaAnimales:` Lista que contiene los animales del zoológico.

Métodos:

- `public String crearFelino(...):` Crea un nuevo objeto `Felino` y lo agrega a la lista.
- `public String crearPrimate(...):` Crea un nuevo objeto `Primate` y lo agrega a la lista.

- `public double costoFelino(...)`: Calcula el costo de mantenimiento anual de un Felino.
- `public double costoPrimate(...)`: Calcula el costo de mantenimiento anual de un Primate.
- `public String agregarFelino(...)`: Agrega un Felino a la lista y devuelve un mensaje de confirmación.
- `public String agregarPrimate(...)`: Agrega un Primate a la lista y devuelve un mensaje de confirmación.
- `public String encontrarAnimal(String nombrecie)`: Busca un animal por su nombre científico y devuelve su información.
- `public String modificarFelino(...)`: Modifica la información de un Felino.
- `public String modificarPrimate(...)`: Modifica la información de un Primate.
- `public String eliminarAnimal(String nombreeelim)`: Elimina un animal de la lista.
- `public String requisitoEspacioF(...)`: Verifica si hay espacio adecuado en otro zoológico para un Felino.
- `public String requisitoEspacioP(...)`: Verifica si hay espacio adecuado en otro zoológico para un Primate.
- `public String enviarAnimal(String animalm)`: Envía un animal a otro zoológico y lo elimina de la lista.
- `DriverProgram`

Propósito: Clase que actúa como punto de entrada del programa, interactuando con el usuario a través de un menú.

Métodos:

- `public static void main(String[] args)`: Método principal que ejecuta el programa y maneja la interacción del usuario.

Diagrama de Clases en UML

