

## **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 (abstracta): Esta clase contiene los atributos generales de cualquier animal, como el nombre científico, hábitat y esperanza de vida. Además, define métodos abstractos como calcularCostoMensual(), calcularAlimentoDiario(), y modificarInformacion(), que deberán ser implementados por las subclases.
- Mamifero: Hereda de Animal y añade atributos propios de los mamíferos (tipo de pelaje, crías, dieta, etc.). Esta clase proporciona algunas funcionalidades compartidas entre sus subclases, pero no tiene métodos abstractos.
- Felino: Hereda de Mamifero y contiene atributos específicos de los felinos (longitud de la cola, color del pelaje, velocidad máxima). Además, implementa los métodos abstractos de Animal relacionados con el cálculo de costos, alimentación y modificación de información.
- Primate: Hereda de Mamifero y contiene atributos específicos de los primates (estructura social, inteligencia, tamaño del cerebro). Implementa los métodos abstractos de Animal con lógica específica para los primates.
- Gestion: Maneja la lista de animales del zoológico (ahora polimórfica, es decir, lista de objetos de tipo Animal). Implementa métodos para gestionar los animales, calcular costos, modificar la información y verificar condiciones de recintos.
- DriverProgram: Es el controlador principal que interactúa con el usuario/director del zoológico. Maneja las entradas y salidas, permitiendo realizar operaciones sobre los animales, utilizando el polimorfismo en la interacción con los objetos.

### **Atributos de las Clases:**

Animal (abstracta):

- nombreCientifico: String: Almacena el nombre científico de los animales.
- habitat: String: Almacena la descripción del hábitat del animal.
- esperanza: double: Almacena la esperanza de vida de los animales.

Mamifero:

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

Felino:

- cola: double: Longitud de la cola en centímetros.
- color: String: Color del pelaje.
- velocidad: double: Velocidad máxima del felino.

Primate:

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

Gestion:

- listaAnimales: List<Animal>: Lista polimórfica que contiene objetos de tipo Animal (puede contener Felino, Primate, u otras subclases de Animal).

### **Métodos de las Clases:**

Animal (abstracta):

- calcularCostoMensual(): Método abstracto para calcular el costo mensual del animal. Implementado por las subclases Felino y Primate.

- `calcularAlimentoDiario()`: Método abstracto para calcular la cantidad de alimento diario que necesita el animal. Implementado por las subclases.
- `modificarInformacion(String atributo, String nuevoValor)`: Método abstracto que permitirá modificar la información de un animal, utilizando una cadena de texto para especificar el atributo a modificar y el nuevo valor.

Felino y Primate:

Implementarán los métodos abstractos de la clase `Animal`:

- `calcularCostoMensual()`: Implementa el cálculo del costo mensual específico para felinos o primates.
- `calcularAlimentoDiario()`: Implementa el cálculo de alimento diario específico.
- `modificarInformacion(String atributo, String nuevoValor)`: Modifica los atributos específicos de felinos o primates según el nombre del atributo y el nuevo valor.

Ejemplo de cómo funcionaría:

Si el atributo a modificar es "cola", en la clase `Felino` se actualiza el valor de la longitud de la cola con `nuevoValor`.

En la clase `Primate`, si se quiere modificar el atributo "inteligencia", se actualizará el valor de inteligencia.

Gestion:

- `agregarAnimal(Animal animal)`: Añade un nuevo animal a la lista polimórfica.
- `eliminarAnimal(Animal animal)`: Elimina un animal de la lista polimórfica.
- `mostrarAnimales()`: Muestra la lista de animales, aprovechando el polimorfismo para acceder a los métodos específicos de cada subclase.
- `calcularAlimentoDiario(Animal animal)`: Calcula el alimento diario mediante el uso de polimorfismo.
- `calcularCostoMensual(Animal animal)`: Calcula el costo mensual utilizando los métodos sobrescritos en las subclases.
- `modificarAnimal(Animal animal, String atributo, String nuevoValor)`: Permite modificar la información de cualquier animal de la lista, llamando al método `modificarInformacion()` correspondiente.
- `esRecintoAdecuado(Animal animal)`: Verifica si el recinto es adecuado para un animal específico.

- consultarIntercambios(): Muestra el historial de intercambios de animales.

DriverProgram:

- main(String[] args): Punto de inicio del programa que interactúa con el usuario para realizar las operaciones del zoológico, gestionando la entrada y salida de datos.

### **Diagrama de Clases en UML**

