

Primer Parcial Mayo 2024

Para evaluar este examen se tendrá en cuenta tanto el correcto funcionamiento de los ejercicios, como la eficiencia de los mismos, por esa razón un ejercicio podrá obtener la máxima nota siempre y cuando tenga en cuenta ambas premisas.

Los ejercicios deben estar resueltos en una página HTML para demostrar su correcto funcionamiento, pudiéndose utilizar la consola, además se debe utilizar una codificación estricta de JavaScript. **Si un ejercicio no está demostrado, el ejercicio se puntuará con un máximo de la mitad de la nota de dicho ejercicio.**

En cada ejercicio se deberá utilizar los métodos de los diferentes objetos del API de JavaScript para su resolución. En el caso de implementar funcionalidad que ya exista en el API, el ejercicio se verá penalizado por dicha circunstancia.

Por último, no se pueden presentar errores en las pruebas de los ejercicios. Errores de sintaxis o no captura de excepciones implicarán la reducción de la nota o la pérdida total de la puntuación.

Instrucciones de entrega

Crea una carpeta con tu nombre, ejemplo "LizanoMontalvoPablo", y dentro de ella crea una carpeta por cada ejercicio, ejemplo "ejercicio1", "ejercicio2", etc, las cuales deberán contener la resolución de cada ejercicio. Empaqueta la carpeta principal en un ZIP y súbelo a Delphos.

Si lo ves necesario, explica el ejercicio en un párrafo de la página HTML que hayas creado.

1. Ejercicio único

El siguiente diagrama muestra una jerarquía de clases que modela la estructura del **reino animal**, la cual deberás implementar mediante un sistema de clases JavaScript. La implementación de cada uno de los métodos tan solo tendrá que imprimir en la consola el nombre del método y la clase desde la que se está ejecutando. Recuerda que las clases **Animal**, **Canino** y **Felino** son **abstractas**. Por último, deberás implementar un método **toString()** que recoja el valor de cada atributo del objeto.

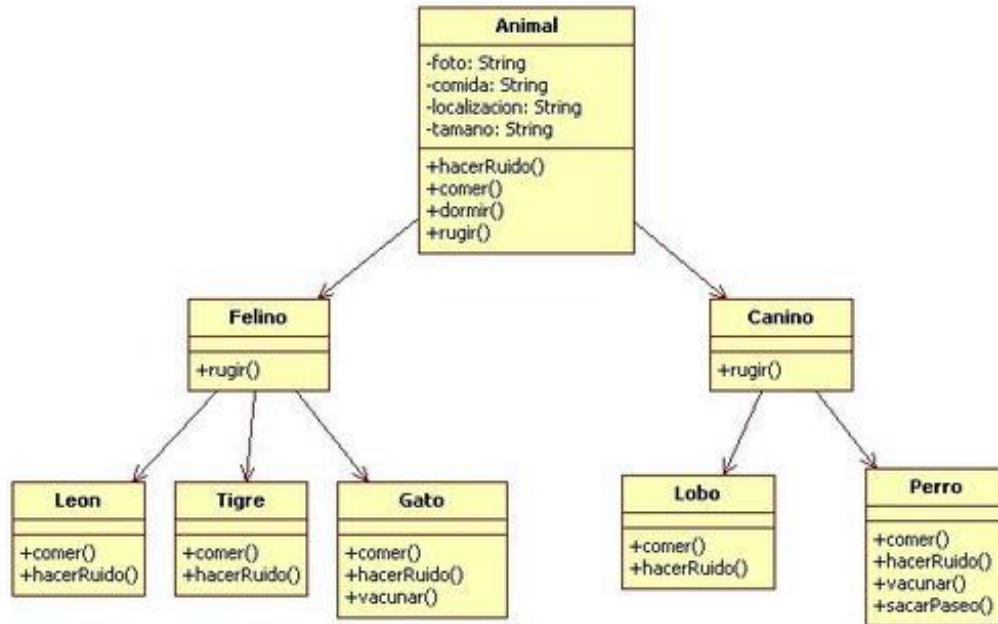


Imagen 1 Diagrama de clases

Vamos a implementar un objeto **Protectora** que permita acoger a todos los animales que vayamos encontrando. La clase deberá disponer de los siguientes métodos:

Método	Funcionalidad	Argumentos	Retorno	Excepciones
Getter animals	Devolver un iterador con todos los animales de la protectora	-	Iterador	-
addAnimal	Añade un nuevo animal a la protectora. Multiargumento y encadenarse	Animal	-	El objeto no es un Animal. El animal ya está registrado.
getAnimals	Devuelve un iterador con los animales de un clase determinada de objeto	Tipo de instancia	Iterador	El tipo de clase no existe.
removeAnimal	Elimina un animal de la protectora Multiargumento y encadenarse	Animal	-	El objeto Animal no existe en la protectora
find	Devuelve los animales que cumplen los criterios de la función de callback	Function	Iterador	

Tabla 1 Listado de métodos

Calificación

La siguiente tabla muestra cómo se calificará este examen:

Criterio de evaluación	Puntos
Clase protectora utilizando un Singleton	0,5 puntos
Jerarquía de clases	1 punto
Clases abstractas	0,5 puntos
Sobrecarga de métodos, incluido toString().	0,5 puntos
Getter animals. Ejemplo de que se recorren	1 punto
Método addAnimal()	1,5 puntos
Método getAnimals()	1,5 puntos
Método removeAnimal()	1 punto
find	1,5 punto
Selecciona todos los animales que sean de tamaño grande y que sean gatos o perros.	1 punto

Tabla 2 Rúbrica