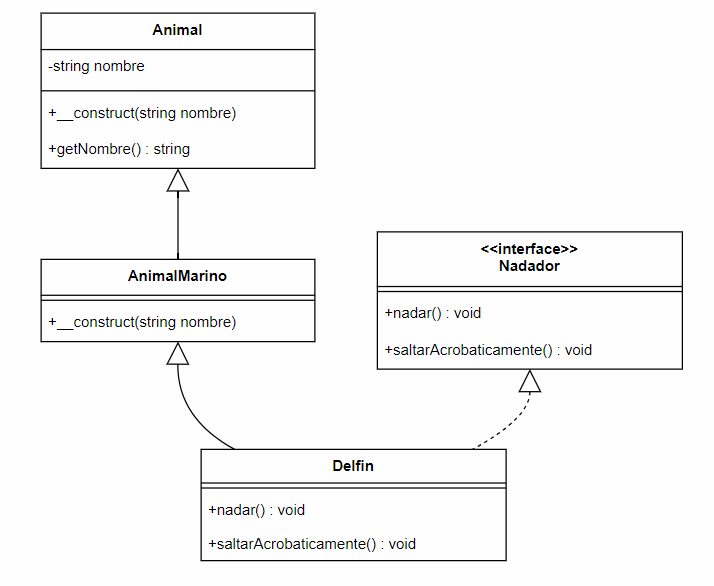
Lista ejercicios prácticos que puedes realizar para aplicar y reforzar los conocimientos adquiridos sobre programación orientada a objetos, interfaces, y clases en PHP:

1. **Herencia Múltiple Simulada con Interfaces:**
   * Crea una clase AnimalMarino y una interfaz Nadador. Implementa una clase Delfin que extienda Animal y que también implemente la interfaz Nadador, agregando métodos específicos para nadar y hacer saltos acrobáticos.
2. **Jerarquía de Clases con Métodos Abstractos:**
   * Diseña una jerarquía de clases con una clase abstracta Vehiculo que tenga métodos abstractos arrancar() y detener(). Luego, implementa clases concretas como Coche, Motocicleta, y Bicicleta, cada una con su propia versión de los métodos.
3. **Gestión de Animales con Polimorfismo:**
   * Crea un programa que pueda manejar diferentes tipos de animales (mamíferos, aves, reptiles) en una misma colección (por ejemplo, un array). Utiliza polimorfismo para llamar a métodos comunes como hacerSonido() sin saber de antemano qué tipo de animal es.
4. **Interfaz de Mascota y Clases que la Implementan:**
   * Define una interfaz Mascota con métodos como jugar(), alimentar(), y mostrarAfecto(). Implementa esta interfaz en clases como Gato, Perro, y Pez. Luego, crea un programa que gestione una colección de mascotas y las cuide utilizando los métodos definidos.
5. **Clases y Métodos Estáticos:**
   * Implementa una clase Zoologico que contenga una lista de animales (mamíferos y aves) y que tenga métodos estáticos para agregar animales a la lista y mostrarlos. Practica el uso de métodos y propiedades estáticas en PHP.
6. **Extensión de Clases con Nuevos Atributos:**
   * Extiende la clase Mamifero para crear una clase Felino, agregando un atributo especie (por ejemplo, león, tigre, gato doméstico) y un método para describir la especie. Luego, crea instancias de diferentes felinos y llama a sus métodos.
7. **Manejo de Excepciones en Métodos:**
   * Implementa una clase Veterinario que tenga un método revisarSalud() que arroje una excepción si el animal está enfermo. Crea un programa que maneje estas excepciones y proporcione un mensaje al usuario.
8. **Serialización y Deserialización de Objetos:**
   * Crea un programa que serialice un objeto Mamifero o Ave a un formato JSON o XML, y luego lo deserialice, mostrando que los datos se mantienen intactos durante el proceso.
9. **Uso de \_\_toString() y Métodos Mágicos:**
   * Agrega el método mágico \_\_toString() a las clases Mamifero y Ave para que devuelvan una cadena con una descripción del animal. Luego, imprime los objetos para ver cómo se invoca automáticamente el método.
10. **Gestión de un Mini Zoológico:**
    * Desarrolla un pequeño programa que simule un zoológico, donde puedas agregar diferentes animales, mostrar la lista de animales disponibles, y realizar acciones como alimentar o hacer que los animales hagan trucos. Utiliza interfaces, clases abstractas, y herencia para organizar el código.



Ejercicio 1



<?php

// Definición de la clase base Animal

class Animal {

    protected $nombre;

    // Constructor para inicializar el nombre del animal

    public function \_\_construct($nombre) {

        $this->nombre = $nombre;

    }

    // Método público para obtener el nombre del animal

    public function getNombre() {

        return $this->nombre;

    }

}

// Definición de la clase AnimalMarino que extiende Animal

class AnimalMarino extends Animal {

    // Constructor para inicializar el nombre del animal marino

    public function \_\_construct($nombre) {

        parent::\_\_construct($nombre);

    }

}

// Definición de la interfaz Nadador

interface Nadador {

    public function nadar();

    public function saltarAcrobaticamente();

}

// Implementación de la clase Delfin que extiende AnimalMarino e implementa Nadador

class Delfin extends AnimalMarino implements Nadador {

    // Implementación del método nadar de la interfaz Nadador

    public function nadar() {

        echo $this->nombre . " está nadando rápido y elegantemente.\n";

    }

    // Implementación del método saltarAcrobaticamente de la interfaz Nadador

    public function saltarAcrobaticamente() {

        echo $this->nombre . " realiza un salto acrobático impresionante fuera del agua.\n";

    }

}

// Ejemplo de uso

$delfin = new Delfin("Flipper");

echo $delfin->getNombre() . " es un delfín.\n";

$delfin->nadar();

$delfin->saltarAcrobaticamente();

?>