

1.- En un proyecto de PHP vamos a implementar la siguiente jerarquía de clases para gestionar un aeropuerto.

Se implementa una interfaz **Volador** con un método acelerar que tiene como parámetro de entrada la velocidad.

Se implementa una clase abstracta **ElementoVolador** que implementa la interfaz Volador. Esta clase tendrá los siguientes atributos privados:

- nombre: Nombre del aparato
- numAlas: Numero de alas
- numMotores: Numero de motores disponibles
- altitud: Metros a los que se encuentra el aparato
- velocidad: Velocidad en la que se encuentra el aparato.

Los métodos que tiene son:

- Constructor público que recibe como argumentos los valores para inicializar los atributos excepto la altitud y la velocidad, que siempre se inicializan a cero.
- Los getter y setter que se crean necesarios.
- Método volando: que devuelve true o false si altitud es mayor que 0.
- Método acelerar: (Se sobreescribe de la interfaz Volador): recibe una velocidad por argumento y actualiza el valor del atributo velocidad.
- Método abstracto volar, que recibe un atributo de altitud.
- Método abstracto mostrarInformación, que muestra toda la información del objeto.

Se implementa la clase **Avion**, que hereda de ElementoVolador y tiene como atributos privados:

- companiaAerea: Compañía a la que pertenece
- fechaAlta: Fecha de alta en la compañía
- altitudMaxima: Altitud máxima que alcanza

Además, tiene los siguientes métodos:

- Constructor público que recibe todos los argumentos necesarios para inicializar sus atributos.
- Los getter y los setter públicos que sean necesarios.
- Método volar: que recibe la altitud que va a tomar. Hay que comprobar que la altitud recibida no sea menor que 0 o superior a la máxima del avión. Si se cumple

esta condición, hay que comprobar que la velocidad del avión es de 150 o más. Si es así, se va incrementando la altitud de 100 metros en 100 metros hasta alcanzar el objetivo. Este método debe sacar por pantalla cada una de las operaciones que realiza, o mensajes de error informando de porque no se ha realizado la acción.

- Método mostrarInformación : nos muestra todos los valores de sus atributos.

Se implementa la clase **Helicoptero**, que hereda de ElementoVolador y tiene como atributos privados:

- propietario: nombre del propietario del helicóptero
- nRotor: número de rotores

Además, tiene los siguientes métodos:

- Constructor público que recibe los argumentos necesarios para inicializar sus atributos
- Los getter y lo setter públicos que sean necesarios.
- Método volar: Recibe un entero con la altitud que va a tomar. La altitud no podrá ser superior a 100 por cada uno de los rotores que tiene. Si cumple la condición, se elevará de 20 en 20 metros.
- Método mostrarInformación: saca los valores de sus atributos.

Se implementa la clase **Aeropuerto**, que tiene como atributos privados un array de ElementosVoladores. Se programan los siguientes métodos públicos:

Constructor por defecto: crea el array.

Método insertar: Recibe un elementoVolador y lo da de alta en el array.

Método buscar: Busca por nombre del aparato. Si lo encuentra muestra sus datos por pantalla. En caso contrario, muestra un mensaje de no encontrado.

Método listarCompania: buscar por nombre de la compañía de un avión. Y si la encuentra muestra el listado de todos los aviones de esa compañía. En caso contrario informa de que la compañía no se encuentra

Método rotores: que muestra un listado de los helicópteros con este número de rotores que se pasa como argumento al método.

Método despegar: que recibe el nombre del elemento que va a despegar, la altitud que va a tomar y la velocidad que va a coger. Se hace una búsqueda en el array hasta que encontremos el elemento cuyo nombre sea igual al del argumento, se acelera este elemento a la velocidad indicada, se pone a volar el elemento a la altitud indicada y devuelve el objeto.

En el **programa principal**:

1. Crea un objeto de tipo Aeropuerto.
2. Crear tres aviones y tres helicópteros.
3. introduce los objetos creados en el aeropuerto.

4. Probar el método buscar. Primero pasando como argumento, un elemento volador que exista y segundo pasando como argumento un elemento volador que no exista
5. Probar el método listarCompania. Al igual que la prueba anterior primero pasando un parámetro con un valor que exista y luego con una compañía que no exista.
6. Probar el método rotores. Primero pasando un parámetro con el número de rotores igual a algunos de los helicópteros que se encuentran en el aeropuerto y después un número de rotores que no encuentre valor.
7. Realizar el despegue de un avión, mostrar si finalmente está volando y la información del avión.
8. Realiza el despegue de un helicóptero, mostrar si finalmente está volando y la información del él una vez realizado el despegue.