

Folla 5.1. POO 1. Introducción á POO en PHP

1. Crea unha clase **contacto**, con propiedades privadas: nome, apelidos, e tfno. Define un construtor con 3 argumentos, que asigne os valores ás propiedades

```
public function __construct($nome, $apelidos, $tfno )
{
    ...
}
```

Define agora para cada propiedade un método para asignar valor e outro para ler dito valor: **getNome()**, **setNome(\$nome)**, **getApelidos()**, **setApelidos(\$apelido)**, **leTfno()**, **asignaTfno(\$tfno)**, **mostraInformacion()**. Garda:

- o nome e apelidos coa primeira letra en maiúscula
- comproba que o tfno (sen prefixo) é un número correcto antes de gardalo.

Crea 3 obxectos da clase contacto, asigna valores para todas as súas propiedades, e mostra a continuación os seus valores, primeiro cos métodos **get()**, e logo con **mostraInformacion()**.

2. Engade un método **__destruct()**, que mostre por pantalla que cada obxecto está sendo destruído.
3. Define agora unha clase **bombilla**, que teña como atributo *privado* a súa **potencia**. Define agora un construtor, que asigne por defecto a súa potencia a 10. Define agora 2 métodos **setPotencia(\$valor)** que asigne á variable potencia ese valor, e **getPotencia()**, que devolva a potencia que ten o obxecto desde que o chamamos.

Define tamén 2 métodos máis **umentaPotencia(\$val)** que aumente a potencia no valor val, e **baixaPotencia(\$val)** que baixe a potencia nese valor. Controla:

- Os valores teñen que estar entre 2 e 35W

Crea un obxecto e comproba que todo funciona: asigna 60, sube 50, baixa 10, asigna 10 e baixa 20, revisando cada vez o valor da potencia.

4. Modifica a clase bombilla creada no exercicio anterior de xeito que teña unha variable estática chamada **numBombillas**, na que se leve a conta das bombillas que se van creando. Fai os seguintes pasos:
 - Crea unha bombilla **bombilla1**, aumenta 20 e mostra a súa potencia
 - Mostra cantas bombillas hai accedendo a **numBombillas**
 - Crea outra bombilla **bombilla2**, aumenta 30 e mostra a súa potencia
 - Elimina **bombilla1**, e mostra cantas bombillas hai.
 - Reduce a potencia de **bombilla2** en 10 e mostra a súa potencia
5. Modifica o exercicio anterior definindo un **destrutor** de bombilla que faga un echo do número de bombillas que existen no programa. Debería mostrar 0 ao rematar a execución do script
6. No exemplo completo da sección **Herdanza** dos apuntes fai os seguintes cambios:
 - x O construtor da clase **empregado** asignará un salario de entrada e tamén un nome, que se pasarán como argumentos. O salario de entrada non poderá superar os 2000 euros.
 - x Crea un método **getSalario()** que nos devolva o salario do obxecto que o chame.

- x Modifica o construtor de Operario para que reciba o turno como argumento (ademais de nome e salario), comprobando tamén que só pode ser “diurno” ou “nocturno”.
- x Fai os cambios que precises na parte de fóra da clase para comprobar que todo funcionalidade

7. Crea unha clase **participante**, con propiedades **nome** e **idade**. Crea os seus construtores e métodos de acceso. Crea agora 2 subclases: **xogador** e **arbitro**. Dos xogadores gardamos o seu **posto** no campo, e do árbitro os **anos** que leva arbitrando. Define construtores e métodos das subclases (lembra que hai que chamar ao construtor da clase nai), establecer limitacións nos datos de entrada.

Crea 2 obxectos de cada subclase dando valores ás súas propiedades e comproba que funcionan os métodos para cambiar **nome**, **idade**, **posto** e **anosArbitraxe**. Mostra todas as propiedades dos 2 obxectos.

8. Crea unha clase **Planeta** que teña como propiedades **nome**, **tamaño** en "soles", **accesible** (valores booleanos *true* ou *false*), e **distancia** á Terra. Garda tamén unha propiedade **estática numPlanetasAccesibles**, que se irá modificando ao introducir planetas no programa. Define os construtores e os métodos para modificar eses valores, así como un método **explotou()**, que cando é chamado cambia a propiedade **accesible** a falso.

Define agora 2 subclases de planeta chamadas **PlanetaHabitable** e **PlanetaHostil**. Os planetasHabitables terán unha propiedade **numeroPersoas** que collerían no planeta, así como **temperaturaMinima** e **temperaturaMáxima**. Dos **PlanetaHostil** gardaremos se hai *vida* ou non, e outro booleano que indica se hai *fontesDeEnerxia*.

Define construtores para as subclases (lembra que hai que chamar aos construtor da clase nai antes de nada) e os métodos para modificar as propiedades novas.

Define tamén un método **listaTodo()** que liste todas as súas propiedades.

Comproba que todo funciona:

- x Crea un obxecto de **PlanetaHabitable**, de nome Fión, 24 soles e 1.5 anos luz de distancia, collendo 1000000 de persoas, entre 0°C e 10°C.
- x Modifica o valor da distancia (mediuse mellor) , e dálle outro nome que che guste máis, así como cambia a **temperaturaMáxima** e **temperaturaMinima**.
- x Crea un obxecto **PlantetaHostil**, de nome Kaleva, 50 soles e 0.4 anos luz, *true* e *true*.
- x Mostra os datos de ambos.
- x Introduce un planeta de cada con valores que queiras, e lista todos os valores de todos os obxectos creados (chamando a **listaTodo()**), comprobando tamén o valor da variable estática.
- x Fai explotar o último planeta creado, e comproba cantos planetas están accesibles.