

U5.- POO

1. Clase COCHE

a) Definir una clase que represente a un coche. En la definición se debe incluir los siguientes atributos *private*

- el modelo
- el color
- si la pintura es metalizada o no
- la matrícula
- El tipo de coche que puede ser MINI, UTILITARIO, FAMILIAR o DEPORTIVO
- El año de fabricación
- la modalidad del seguro, que puede ser a terceros o a todo riesgo

b) Añadir a la clase Coche métodos para poder introducir y leer los datos de un coche desde otra clase

c) Añadir a la clase Coche del ejercicio 1 un método de nombre imprimeCoche que imprima el modelo y el color del coche.

2. Escribe un programa que disponga de una clase para representar asignaturas de una carrera:

- Una asignatura tiene un nombre, un código numérico y el curso en el cual se imparte
- Los valores iniciales han de proporcionarse en el constructor
- La clase ha de tener métodos para obtener los valores de los atributos
- El programa ha de construir un objeto con los siguientes valores: nombre "Matemáticas", código 1017, curso 1.
- A continuación, el programa ha de imprimir los valores del objeto por pantalla

3. Clase PUNTO

a) Definir una clase que represente un punto en un espacio bidimensional. La clase debe disponer de un constructor con las coordenadas del punto y métodos accesorios a las coordenadas

b) Escribe un programa que instancie cuatro puntos: el primero situado en el origen, el segundo situado en (5,3), el tercero en (2, -1) y el cuarto estará situado en el punto medio entre el segundo y el tercero.

c) Añade a la clase Punto un constructor sin parámetros, que permita construir puntos en el origen de coordenadas

d) Añade un método a la clase Punto que calcule la distancia a otro punto

e) Escribir un programa que cree un punto en (4,3) y visualice la distancia del punto al origen de coordenadas

- f) Escribe la clase Punto con dos métodos llamados distancia. Uno de ellos calcula la distancia a otro punto y el otro calcula la distancia al origen
4. Escribe una clase que represente un reloj que señale la hora, el minuto y el segundo.
- a) La clase dispondrá de dos constructores, uno sin parámetros que pone el reloj a 0:0:0 y otro al que se le pasa la hora, los minutos y los segundos

Se proporcionarán los siguientes métodos:

- b) Uno que da la hora, los minutos y los segundos, separados por el carácter “:”, en una cadena
- c) Otro que también da la hora pero en formato 24 horas (como el anterior) o en formato 12, en cuyo caso debe distinguir entre “am” (para las horas de 0 a 11) o “pm” (para las horas de 12 a 23), también en una cadena
- d) Un método para poner el reloj en hora. Se le pasa la hora y los minutos, poniendo los segundos a 0
- e) Un método para poner el reloj en hora al que, además, se le pasan los segundos
- f) Añada al ejercicio anterior un método para poner la hora especificando si es AM o es PM

U5.- POO static

1. Escribe un programa para representar el consumo de energía de una instalación eléctrica. Para ello se dispondrá de una clase que representa los aparatos conectados en la instalación, Cada aparato tiene un consumo eléctrico determinado. Al encender un aparato eléctrico, el consumo de energía se incrementa en la potencia de dicho aparato. Al apagarlo se decrementa el consumo. Inicialmente, los aparatos están todos apagados. Además, se desea consultar el consumo total de la instalación.

Hacer un programa que declare tres aparatos eléctricos, una bombilla de 100 watios, un radiador de 2000 watios y una plancha de 1200 watios. El programa imprimirá el consumo nada más crear los objetos. Posteriormente, se enciende la bombilla y la plancha, y el programa imprime el consumo. Luego se apaga la plancha y se enciende le radiador y se vuelve a imprimir el consumo.

2. Añade a la clase Punto una constante llamada ORIGEN que sea el origen de coordenadas. La constante debe ser accesible de forma estática. Escribe un programa, de nombre PruebaPunto, que determine la distancia de los puntos (3,4), (0,4) y (2,-1) al origen usando la constante ORIGEN.
3. Escribe una clase llamada Elemento, que disponga de un atributo con su nombre. La clase debe contener un método llamado númeroDeElementos que devuelve el número total de elementos que se han instanciado.
4. Escribe una clase de nombre Item, que acepta un nombre en su constructor. Cada ítem debe disponer de un número que servirá como identificador único de cada objeto. La clase ítem dispondrá de un método para obtener el identificador y otro para obtener el nombre. Haga un programa de prueba que genere tres ítems, “uno”, “dos” y “tres” y luego escriba los nombres e identificadores de cada ítem.
5. Se desea representar las bombillas que pueda haber en una casa. Cada bombilla tiene asociada un interruptor y sólo uno. Así mismo, existe un interruptor general de la casa. Un interruptor tiene dos estados, ON y OFF. Una bombilla luce si el interruptor general de la casa está ON y su interruptor asociado también. Escribe una clase de nombre Bombilla que permita modelar la información anterior. Para ello, la clase dispondrá de:
 - un método para cambiar el estado del interruptor de la bombilla.
 - Un método para cambiar el estado del interruptor general de la casa.
 - Un método que determina si una bombilla está luciendo o no.

Escribe un programa que utilice la clase Bombilla del apartado anterior. Para ello, se creará una Bombilla y se visualizará por pantalla si luce o no. Luego se pulsa el interruptor de la Bombilla y se vuelve a visualizar el estado de la misma. Por último, se pulsa el interruptor general y se visualiza el estado.