

Ejercicio clase Consumo

Implementa una clase *consumo*, la cual forma parte de la centralita electrónica de un coche y tiene las siguientes características:

Atributos (Propiedades, Datos):

- kms. Kilómetros recorridos por el coche.
- litros. Litros de combustible consumido.
- vmed. Velocidad media.
- pgas. Precio de la gasolina.

Constructores:

Crea un constructor por defecto con los valores que quieras, y crea otro constructor que reciba los 4 parámetros de entrada (propiedades de la clase) que le pase el usuario por pantalla.

Métodos:

- get y set para cada uno de las propiedades de la clase.
- tiempo. Indicará el tiempo empleado en realizar el viaje.
- consumoMedio. Consumo medio del vehículo (en litros cada 100 kilómetros).
- consumoEuros. Consumo medio del vehículo (en euros cada 100 kilómetros).

Programa principal:

- Crear un primer objeto de la clase consumo con el constructor por defecto.
- Crear un segundo objeto de la clase consumo con el constructor que recibe parámetros de entrada (habrá que pedir esos datos previamente por pantalla).
- Con el primer objeto, mostrar todos sus atributos por pantalla.
- Con el primer objeto, llamar al método consumoMedio y sacar el resultado por pantalla.
- Con el segundo objeto, llamar al método Tiempo y sacar el resultado por pantalla.
- Con el segundo objeto, llamar al método consumoMedio y sacar el resultado por pantalla.
- Con el segundo objeto, llamar al método consumoEuros y sacar el resultado por pantalla.

Ejercicio clase MiNumero

Realizar un programa que dado un número por pantalla, calcule el doble, triple, y cuádruple de ese número proporcionado, y muestre el resultado por pantalla.

Para ello vamos a crear una clase que contenga un atributo, un constructor, y los métodos que calculen las 3 cosas (doble, triple y cuádruple).

El funcionamiento será el siguiente, pediremos el número al usuario por pantalla, y mostraremos un menú preguntando por la acción que quiere hacer:

- 1) doble
- 2) triple
- 3) Cuádruple

En base a la opción elegida, nos devolverá el resultado.

El menú tendrá que poder volver si lo deseamos.

*/*de momento obviad la siguiente línea, lo recuperaremos más adelante:*

Hay que gestionar las excepciones que puedan suceder en el código.

Ejercicio clase Peso

Crea un programa principal que pida por pantalla un peso en kg, y pregunte si se quiere transformar en libras, lingotes, onzas, peniques, gramos o quintales. Luego devolverá el resultado de la conversión elegida.

Para ello, crea la clase **Peso**, la cual tendrá las siguientes características:

Un atributo "kilos" donde se almacene el peso de un objeto en kilogramos.

Deberá de tener los siguientes métodos:

- *setKilos*. Guarda el peso en kg.
- *getKilos*. Devuelve el peso en kg.
- *getPeso*. Devuelve el peso en la medida que se pase como parámetro (libras, lingotes, onzas, peniques, kilos, gramos o quintales).

Para la realización del ejercicio toma como referencia los siguientes datos:

- 1 Libra = 453 gramos.
- 1 Lingote = 14,59 kg.
- 1 Onza = 28,35 gramos.
- 1 Penique = 1,55 gramos.
- 1 Quintal = 43,3 kg.

*/*de momento obviad la siguiente línea, lo recuperaremos más adelante:*

Hay que gestionar las excepciones que puedan suceder en el código.

Ejercicio Restaurante

Un restaurante mejicano cuya especialidad son las papas con chocos, nos pide diseñar un programa que gestione su almacén, y que para ello haga todo esto:

- Añadir papas: Guardar kg de papas en almacén (añadiéndolos a los que ya tengamos en el almacén)

- Mostrar almacén papas: Muestra por pantalla los kilos de papas que hay en el almacén.

- Añadir chocos: Guardar kg de chocos en almacén (añadiéndolos a los que ya tengamos en el almacén)

- Mostrar almacén chocos: Muestra por pantalla los kilos de chocos que hay en el almacén.

- Merma de chocos: Restamos los kg de chocos que se han estropeado y se han tirado a la basura, del total que tenemos en el almacén.

- Merma de papas: Restamos los kg de papas que se han estropeado y se han tirado a la basura, del total que tenemos en el almacén.

- Devuelve el número de clientes que puede atender el restaurante: Para calcularlo, utilizaremos el número de papas y el número de chocos que tenemos actualmente en el almacén sabiendo que cada persona consume $\frac{1}{4}$ kilo de papas y $\frac{1}{4}$ kilo de chocos (con cada kilo de papas y chocos se atiende a 4 clientes).

Será necesario crear una clase almacén, y el programa principal con un menú que gestionará todas las peticiones anteriores.

Ejemplo Examen T2 PROG

Ejemplo Examen T2 PROG

Hay que desarrollar una aplicación para gestionar una tienda de ropa. En ésta tienda se venden Camisas y Pantalones.

Esa aplicación deberá tener **un menú principal**, que nos presentará por pantalla las siguientes opciones:

1) Añadir Camisas al almacén. (Al pulsar ésta opción, el dueño de la tienda podrá introducir por pantalla una cantidad de Camisas, que nos han entrado en la tienda, y acumular la cantidad a la que ya tuviéramos)

2) Añadir Pantalones al almacén. (Igual que el anterior pero con Pantalones)

3) Vender Camisa. (Al pulsar ésta opción, restaremos una camisa a las camisas que tenemos)

4) Vender Pantalón. (Igual que el anterior pero con Pantalones)

5) Comprobar stock Camisas. (Recuperamos el número de Camisas que tenemos en la tienda, y vemos si tenemos que hacer pedido de la siguiente manera:

- Si el número de Camisas es menor a 5, sacaremos por pantalla "Es necesario hacer un pedido de Camisas"

- Si el número de Camisas está entre 5 y 10, sacaremos por pantalla "Próximamente será necesario hacer un pedido de Camisas"

- Si el número de Camisas es mayor a 10, sacaremos por pantalla "De momento no es necesario hacer pedido de Camisas"

6)Comprobar stock Pantalones.(Igual que el anterior pero con Pantalones)

Al finalizar cada opción del menú, se debe preguntar si se quiere volver al menú de nuevo, para elegir otra opción y seguir trabajando con la aplicación.

Para ello, tendréis que utilizar **un programa principal, y una clase Tienda.**

Nota: los apartados 1), 2), 5), y 6) valen 2ptos. cada uno, los apartados 3) y 4) valen 1pto. entre los dos, y el pto. restante es para la calidad del programa, orden...