

A partir del siguiente artículo, resuelve los siguientes ejercicios y crea en función de lo que creas una Interfaz o una Clase (excepto si se especifica lo contrario). Una vez hechos crea un único documento con todas las soluciones y entrégalo en esta tarea. Recordad siempre de poner el tipo de los parámetros, atributos y retorno de funciones.

<https://www.javatpoint.com/typescript-class-vs-interface>

1. Cread una Interfaz o Clase que se llame **Producto** y que contenga: *Nombre, Precio y Oferta (opcional)*. En el **programa principal**, tendréis que crear los siguientes productos dentro de un array, crear una función que muestre todos los productos en oferta y otra que los muestre todos ordenados de mayor a menor precio:
 1. Camisa de 60€.
 2. Pantalones de 45€, además estará en oferta.
 3. Polo de 35€.
 4. Slip de 15€, además estará de oferta.
2. A partir de la documentación de este [enlace](#). Cread una Interfaz o Clase que se llame **Persona** y que contenga: *Nombre, Edad, DNI y Domicilio*. Tiene que tener **constructor** con parámetros en el que se inicialicen por defecto todos los parámetros y cada parámetro tiene que tener getter + setter. Además tiene que tener una función que se llame **MostrarInfo**, que devuelva toda la información de la persona en una cadena de texto. En el **programa principal**, tendréis que crear 2 usuarios de prueba y mostrar su información por consola.
3. Cread una Clase que se llame **Cafetera** y que contenga los : *Capacidad Maxima (la cantidad máxima de café que puede contener la cafetera en ml)* y *Cantidad Actual (la cantidad actual de café que hay en la cafetera en ml)*. Tiene que tener al menos las siguientes funciones:

Constructor por defecto que establezca la capacidad máxima en 1000 y la actual en cero (cafetera vacía).

Los getters y setters que correspondan.

Función **Llenar Cafetera**: Hace que la cantidad actual sea igual a la capacidad.

Función **Servir Taza**: Simula la acción de servir una taza con la capacidad que se le pase por parámetro. Si la cantidad actual de café "no alcanza" para llenar la taza, se sirve lo que haya en la cafetera.

Función **Vaciar Cafetera**: Pone la cantidad de café actual a cero.

Función **Agregar Café**: Añade a la cafetera la cantidad de café indicada por parámetro.

4. Se plantea desarrollar un programa que permita la gestión de una empresa agroalimentaria que trabaja con tres tipos de productos: Frescos, Refrigerados y Congelados. Todos los productos llevan esta información en común: Nombre del producto y Número de lote. A su vez, cada tipo de producto lleva alguna información específica:
- Los productos frescos deben llevar la fecha de envasado y el país de origen.
 - Los productos refrigerados deben llevar el código del organismo de supervisión alimentaria.
 - Los productos congelados deben llevar la temperatura de congelación recomendada.

Implementa las clases Producto, ProductoFresco, ProductoRefrigerado y ProductoCongelado. Tenéis que usar una relación de herencia entre las clases que pertoca. Cada clase debe disponer de constructor y permitir establecer (set) y recuperar (get) el valor de sus atributos y tener un método que permita mostrar la información del objeto.

A continuación os dejo una página que os puede servir de ayuda para mejorar la calidad de vuestro código y reducir el número de líneas ([link](#)).