

- 1) Desarrolla una clase Tetera con atributos `_capacidadMaxima` (la cantidad máxima de te que puede contener la tetera) y `_cantidadActual` (la cantidad actual de te que hay en la tetera).

Implementa, al menos, los siguientes métodos:

- Constructor predeterminado: establece la capacidad máxima en 1000 (c.c.) y la actual en cero (tetera vacía).
- Constructor con la capacidad máxima de la tetera; inicializa la cantidad actual de te igual a la capacidad máxima.
- Constructor con la capacidad máxima y la cantidad actual. Si la cantidad actual es mayor que la capacidad máxima de la te, la ajustará al máximo.
- Observadores y Modificadores
- `llenarTe()`: hace que la cantidad actual sea igual a la capacidad.
- `servirTaza(int)`: simula la acción de servir una taza con la capacidad indicada. Si la cantidad actual de te “no alcanza” para llenar la taza, se sirve lo que quede.
- `vaciarTe()`: pone la cantidad de te actual en cero.
- `agregarCafe(int)`: añade a la tetera la cantidad de te indicada

- 2) Crea una clase NIF que se usará para gestionar DNIs con su correspondiente letra. Los atributos serán el número de DNI (entero largo) y la letra que le corresponde. La clase dispondrá de los siguientes métodos:

- Constructor predeterminado que inicialice el nº de DNI a 0 y la letra a espacio en blanco (será un NIF no válido).
- Constructor que reciba el DNI y establezca la letra que le corresponde.
- Modificadores y Observadores.
- `leer()`: que pida el número de DNI (ajustando automáticamente la letra).
- Método que nos permita mostrar el NIF (ocho dígitos, un guión y la letra en mayúscula; por ejemplo: 00395469-F) La letra se calculará con un método auxiliar (privado) de la siguiente forma: se obtiene el resto de la división entera del número de DNI entre 23 y se usa la siguiente tabla para obtener la letra que corresponde:

0 - T	1 - R	2 - W	3 - A	4 - G	5 - M	6 - Y
7 - F	8 - P	9 - D	10 - X	11 - B	12 - N	13 - J
14 - Z	15 - S	16 - Q	17 - V	18 - H	19 - L	20 - C
21 - K	22 - E					

3) Crea la clase:

Motor: con métodos para arrancar el motor y apagarlo.

4) Crea la clase:

Rueda: con métodos para inflar la rueda y desinflarla.

5) Crea la clase:

Ventana: con métodos para abrirla y cerrarla.

6) Crea la clase:

Puerta: con una ventana y métodos para abrir la puerta y cerrar la puerta.

7) Crea la clase

Furgoneta: con un motor, cuatro ruedas y 5 puertas con los métodos que te necesites.

8) Desarrolla una clase Serie con los siguientes atributos:

título: una variable String que almacena el título de la serie.

autor: una variable String que almacena el autor de la serie.

actor: una variable int que almacena el número de actores.

Con los siguientes métodos:

Serie(String, String): constructor que recibe como parámetros el título y el autor de la serie (por este orden).

Serie(): constructor predeterminado que inicializa el título y el autor a cadenas vacías.

Observadores y Modificadores.

9) Desarrolla un programa en Java que calcule el pedido que realiza un cliente en un restaurante.

Los pedidos de un restaurante están formados por:

Un primer plato.  
Un segundo plato.  
Una bebida.  
Un postre.

Cada item anterior tiene un nombre y un valor. Se necesita realizar métodos sobrecargados para calcular el valor del pedido dependiendo si el cliente solicita:

Un primer plato y una bebida.  
Un primer plato, un segundo plato y una bebida.  
Un primer plato, un segundo plato, una bebida y un postre.

Implementar un método main que utilice los tres métodos sobrecargados en tres diferentes pedidos.

- 10) Realiza un programa en Java que permita realizar las siguientes conversiones de unidades de longitud:

Metros a centímetros.  
Metros a milímetros.  
Metros a pulgadas.  
Metros a pies.  
Metros a yardas.

Utiliza métodos.

- 11) Desarrollar una clase denominada Resta, la cual tiene varios métodos restar sobrecargados:

Un método restar que obtiene la resta de dos valores enteros pasados como parámetros.  
Un método restar que obtiene la resta de tres valores enteros pasados como parámetros.  
Un método restar que obtiene la resta de dos valores double pasados como parámetros.  
Un método restar que obtiene la resta de tres valores double pasados como parámetros.

- 12) Desarrollar un programa en Java que permita modelar un artículo deportivo. Los artículos deportivos contienen los siguientes metadatos: nombre del artículo,

autor, palabras claves, nombre de la publicación, año y resumen. Se deben definir tres constructores sobrecargados:

El primero inicializa un artículo deportivo con solo su título y autor.

El segundo constructor, un artículo deportivo con su nombre, autor, palabras claves, nombre de la publicación y año. Debe invocar al primer constructor.

El tercer constructor, un artículo deportivo con su nombre, autor, palabras claves, nombre de la publicación, año y resumen. Debe invocar al segundo constructor. Se requiere un método que imprima los atributos de un artículo en pantalla. Realizar un método main que utilice el tercer constructor para instanciar un artículo deportivo e imprima los valores de sus atributos en pantalla.

- 13) Realizar un programa en Java que permita modelar un atleta, el cual tiene los siguientes atributos:

Identificador del atleta de tipo int.

Nombre del atleta de tipo String.

Tiempo en realizar una prueba de 100 metros por relevos, de tipo double.

Además, se tienen los siguientes atributos estáticos:

Un contador con la cantidad de atletas creados, que servirá como identificador de un atleta.

Un atributo estático denominado selección de tipo String con el valor "Francia".

Un atributo estático denominado tiempoEquipo que sume los valores de tiempo obtenidos por un equipo de atletas.

Es necesario crear un constructor para la clase con los parámetros: nombre y tiempo de cada atleta. En el constructor se debe inicializar el identificador del atleta con base en el contador estático. También se deben implementar los siguientes métodos:

Correr 100 metros: este método actualiza el tiempo total de carrera del equipo sumando el tiempo de carrera de cada atleta.

Un método estático que imprima el nombre de la selección de atletismo, el cual es un atributo estático.

Un método estático que imprima el tiempo total obtenido por el equipo de atletismo.

Un método main que debe crear cuatro atletas que conforman el equipo de relevos de 100 metros.

