

**FRUTN - MDP - Técnico Superior en Programación**  
**Programación III - Laboratorio III**

---

### Ejercicios varios

- 1) Haz una clase llamada Persona que siga las siguientes condiciones:
  - Sus atributos son: nombre, edad, DNI, sexo (H hombre, M mujer), peso y altura. No queremos que se accedan directamente a ellos. Piensa que modificador de acceso es el más adecuado, también su tipo. Si quieres añadir algún atributo puedes hacerlo.
  - Por defecto, todos los atributos menos el DNI serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.). Sexo sera hombre por defecto, usa una constante para ello.
  - Se implantaran varios constructores:
    - Un constructor por defecto.
    - Un constructor con el nombre, edad y sexo, el resto por defecto.
    - Un constructor con todos los atributos como parámetro.
  - Los métodos que se implementarán son:
    - calcularIMC(): calcula si la persona está en su peso ideal (peso en  $\text{kg}/(\text{altura}^2 \text{ en m})$ ), si esta fórmula devuelve un valor menor que 20, la función devuelve un -1, si devuelve un número entre 20 y 25 (incluidos), significa que está por debajo de su peso ideal la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1. Te recomiendo que uses constantes para devolver estos valores.
      - esMayorDeEdad(): indica si es mayor de edad, devuelve un booleano.
      - comprobarSexo(char sexo): comprueba que el sexo introducido es correcto. Si no es correcto, será H. No será visible al exterior.
      - toString(): devuelve toda la información del objeto.
      - generaDNI(): genera un número aleatorio de 8 cifras. Este método será invocado cuando se construya el objeto. Puedes dividir el método para que te sea más fácil. No será visible al exterior.
      - Métodos set de cada parámetro, excepto de DNI.

Ahora, crea una clase ejecutable (main) que haga lo siguiente:

- Pide por teclado el nombre, la edad, sexo, peso y altura.
- Crea 3 objetos de la clase anterior, el primer objeto obtendrá las anteriores variables pedidas por teclado, el segundo objeto obtendrá todos los anteriores menos el peso y la altura y el último por defecto, para este último utiliza los métodos set para darle a los atributos un valor.
- Para cada objeto, deberá comprobar si está en su peso ideal, tiene sobrepeso o por debajo de su peso ideal con un mensaje.
- Indicar para cada objeto si es mayor de edad.
- Por último, mostrar la información de cada objeto.

3) Haz una clase llamada Password que siga las siguientes condiciones:

Que tenga los atributos longitud y contraseña . Por defecto, la longitud sera de 8.

Los constructores serán los siguiente:

Un constructor por defecto.

Un constructor con la longitud que nosotros le pasemos. Generara una contraseña aleatoria con esa longitud.

Los métodos que implementa serán:

- **esFuerte()**: devuelve un booleano si es fuerte o no, para que sea fuerte debe tener más de 2 mayúsculas, más de 1 minúscula y más de 5 números.
- **generarPassword()**: genera la contraseña del objeto con la longitud que tenga.
- Método **get** para contraseña y longitud.
- Método **set** para longitud.

Ahora, crea una clase clase ejecutable:

- Crea un array de Passwords con el tamaño que tu le indiques por teclado.
- Crea un bucle que cree un objeto para cada posición del array.
- Indica también por teclado la longitud de los Passwords (antes de bucle).
- Crea otro array de booleanos donde se almacene si el password del array de Password es o no fuerte (usa el bucle anterior).
- Al final, muestra la contraseña y si es o no fuerte (usa el bucle anterior).

4) Crearemos una supeclase llamada Electrodomestico con las siguientes características:

- Sus atributos son precio base, color, consumo energético (letras entre A y F) y peso. Indica que se podrán heredar.
- Por defecto, el color sera blanco, el consumo energético sera F, el precioBase es de 100 € y el peso de 5 kg. Usa constantes para ello.
- Los colores disponibles son blanco, negro, rojo, azul y gris. No importa si el nombre esta en mayúsculas o en minúsculas.
- Los constructores que se implementaran serán
  - Un constructor por defecto.
  - Un constructor con el precio y peso. El resto por defecto.
  - Un constructor con todos los atributos.

Los métodos que implementara serán:

- Métodos get de todos los atributos.

- comprobarConsumoEnergetico(char letra): comprueba que la letra es correcta, sino es correcta usara la letra por defecto. Se invocara al crear el objeto y no sera visible.
- comprobarColor(String color): comprueba que el color es correcto, sino lo es usa el color por defecto. Se invocara al crear el objeto y no sera visible.
- precioFinal(): según el consumo energético, aumentara su precio, y según su tamaño, también.

Esta es la lista de precios:

LETRA	PRECIO
A	100 €
B	80 €
C	60 €
D	50 €
E	30 €
F	10 €

  

TAMAÑO	PRECIO
Entre 0 y 19 kg	10 €
Entre 20 y 49 kg	50 €
Entre 50 y 79 kg	80 €
Mayor que 80 kg	100 €

Crearemos una subclase llamada **Lavadora** con las siguientes características:

- Su atributo es carga, ademas de los atributos heredados.
- Por defecto, la carga es de 5 kg. Usa una constante para ello.
- Los constructores que se implementaran serán:
  - Un constructor por defecto.
  - Un constructor con el precio y peso. El resto por defecto.
  - Un constructor con la carga y el resto de atributos heredados. Recuerda que debes llamar al constructor de la clase padre.

Los métodos que se implementara serán:

- Método get de carga.
- precioFinal():, si tiene una carga mayor de 30 kg, aumentara el precio 50 €, sino es así no se incrementara el precio. Llama al método padre y añade el código necesario. Recuerda que las condiciones que hemos visto en la clase Electrodomestico también deben afectar al precio.

Crearemos una subclase llamada Television con las siguientes características:

- Sus atributos son resolución (en pulgadas) y sintonizador TDT (booleano), ademas de los atributos heredados.
- Por defecto, la resolución sera de 20 pulgadas y el sintonizador sera false.

- Los constructores que se implementaran serán:
  - Un constructor por defecto.
  - Un constructor con el precio y peso. El resto por defecto.
  - Un constructor con la resolución, sintonizador TDT y el resto de atributos heredados.

Los métodos que se implementara serán:

- Método get de resolución y sintonizador TDT.
- precioFinal(): si tiene una resolución mayor de 40 pulgadas, se incrementara el precio un 30% y si tiene un sintonizador TDT incorporado, aumentara 50 €.

Ahora crea una clase ejecutable que realice lo siguiente:

- Crea un array de Electrodomesticos de 10 posiciones.
- Asigna a cada posición un objeto de las clases anteriores con los valores que desees.
- Ahora, recorre este array y ejecuta el método precioFinal().Deberás mostrar el precio de cada clase, es decir, el precio de todas las televisiones por un lado, el de las lavadoras por otro y la suma de los Electrodomesticos (puedes crear objetos Electrodomestico, pero recuerda que Television y Lavadora también son electrodomésticos). Recuerda el uso operador instanceof.

Por ejemplo, si tenemos un Electrodomestico con un precio final de 300, una lavadora de 200 y una televisión de 500, el resultado final sera de 1000 (300+200+500) para electrodomésticos, 200 para lavadora y 500 para televisión.

5) Crearemos una clase llamada Serie con las siguientes características:

- Sus atributos son titulo, numero de temporadas, entregado, genero y creador.
- Por defecto, el numero de temporadas es de 3 temporadas y entregado false. El resto de atributos serán valores por defecto según el tipo del atributo.
- Los constructores que se implementaran serán:
  - Un constructor por defecto.
  - Un constructor con el titulo y creador. El resto por defecto.
  - Un constructor con todos los atributos, excepto de entregado.

Los métodos que se implementara serán:

- Métodos get de todos los atributos, excepto de entregado.
- Métodos set de todos los atributos, excepto de entregado.
- Sobrescribe los métodos toString.

Crearemos una clase Videojuego con las siguientes características:

- Sus atributos son titulo, horas estimadas, entregado, genero y compañía.
- Por defecto, las horas estimadas serán de 10 horas y entregado false. El resto de atributos serán valores por defecto según el tipo del atributo.

- Los constructores que se implementaran serán:
  - Un constructor por defecto.
  - Un constructor con el titulo y horas estimadas. El resto por defecto.
  - Un constructor con todos los atributos, excepto de entregado.

Los métodos que se implementara serán:

- Métodos get de todos los atributos, excepto de entregado.
- Métodos set de todos los atributos, excepto de entregado.
- Sobrescribe los métodos toString.

Tanto los videojuegos como las series tienen el siguiente comportamiento. Elegir el mejor diseño de clases para solventar el comportamiento comun

- entregar(): cambia el atributo prestado a true.
- devolver(): cambia el atributo prestado a false.
- isEntregado(): devuelve el estado del atributo prestado.
- Método compareTo (Object a), compara las horas estimadas en los videojuegos y en las series el numero de temporadas. Como parámetro que tenga un objeto, no es necesario que implementes la interfaz Comparable. Recuerda el uso de los casting de objetos.

Implementa los anteriores métodos en las clases Videojuego y Serie. Ahora crea una aplicación ejecutable y realiza lo siguiente:

- Crea dos arrays, uno de Series y otro de Videojuegos, de 5 posiciones cada uno.
- Crea un objeto en cada posición del array, con los valores que desees, puedes usar distintos constructores.
- Entrega algunos Videojuegos y Series con el método entregar().
- Cuenta cuantos Series y Videojuegos hay entregados. Al contarlos, devuélvelos.
- Por último, indica el Videojuego tiene más horas estimadas y la serie con mas temporadas. Muestralos en pantalla con toda su información (usa el método toString()).

5) Se desea programar la facturación de una empresa de ventas de productos, que tenga 2 tipos de clientes: clientes minoristas y clientes mayoristas. Los clientes mayoristas difieren de los minoristas en que reciben un porcentaje de descuento variable (no es el mismo para todos los clientes) sobre el total de su compra. Además la factura puede tener un número variable de renglones, cada uno de los cuales tendrá la cantidad de artículos, el código o descripción del artículo, el precio unitario y el precio total del renglón.

Debe realizar el programa utilizando una clase que herede de otra donde lo considere conveniente.

Almacene los clientes en un arreglo de hasta 100 clientes **sea cual sea su tipo** , o en una lista dinámica.

La salida del programa debe ser la impresión de la factura por pantalla (Consola), que muestre:

- número de factura
- fecha
- nombre y domicilio del cliente
- CUIT del cliente
- todos los renglones de la factura, y en cada renglón: cantidad, ítem, precio unitario y precio total.
- el total sin descuento
- el total con el descuento aplicado, si correspondiere.

Implemente los renglones de la factura como una lista, y escriba los métodos para agregar un nuevo renglón y para sumar el total de todos los renglones.

Funciones requeridas:

- Ingresar e imprimir nueva factura
- Ingresar nuevo cliente
- Todos los constructores y métodos necesarios para que el programa funcione.

6) En un sistema de administración de un negocio de alquiler de vehículos se desea llevar control del estado de los vehículos disponibles y su historial de alquileres.

Existen 2 tipos de vehículos: automóviles y utilitarios.

Para cada vehículo desea llevarse información de patente, marca, modelo, año, kilometraje, precio de alquiler diario y estado (disponible, alquilado, en reparación).

Para los vehículos utilitarios se quiere llevar información sobre el tipo de carga transportada.

Al alquilar el vehículo se asigna la fecha y el nombre del cliente que lo alquiló. Cuando el vehículo es devuelto se controla la cantidad de kilómetros recorridos y se envía el mismo a una revisión técnica.

Si la revisión técnica es favorable, se vuelve a colocar el vehículo disponible para alquiler.

En caso contrario, se lo envía a reparación.

Implemente un sistema que permita:

- Dar de alta y modificar vehículos
- Alquilar y devolver vehículos
- Establecer el resultado de la revisión técnica de un vehículo
- Listar el estado de la flota de vehículos
- Mostrar el historial de alquileres de un vehículo seleccionado de la flota

Los estados posibles para los vehículos son: disponible, en verificación técnica, en reparación, y alquilado.

Realice correctamente la documentación del código.

7) En un sistema de administración de un club se lleva la información contable de sus socios. Existen 3 tipos de socios: los normales, que pagan una cuota mensual, los vitalicios, que no pagan cuota, y los ausentes, que pagan una cuota anual de mantenimiento, y un acceso a un valor reducido al club cuando se encuentran presentes.

Implemente un sistema para cargar los datos de los socios, cargar el pago de las cuotas, y listar la información del estado de deuda de los socios del club.

Utilice herencia en la definición de los socios.

Defina todos los atributos y métodos necesarios para cada clase.

Defina un menú de acceso al sistema que contenga como mínimo opciones para cargar socios, pagar cuotas, y listar información.

Realice correctamente la documentación del código.