Ejercicios de POO – Bloque 02

Ejercicio 1:

Crear una clase Libro que contenga los siguientes atributos:

– ISBN

– Titulo

– Autor

– Número de páginas

Crear sus respectivos métodos get y set correspondientes para cada atributo. Crear el método mostrar_información() para mostrar la información relativa al libro con el siguiente formato:

«El libro con ISBN creado por el autor tiene páginas»

En el fichero main, crear 2 objetos Libro (los valores que se quieran) y mostrarlos por pantalla.

Por último, indicar cuál de los 2 tiene más páginas.

Ejercicio 2:

Crea una clase llamada Cuenta que tendrá los siguientes atributos: titular y cantidad (puede tener decimales).

El titular será obligatorio y la cantidad es opcional. Crea dos constructores que cumpla lo anterior.

Crea sus métodos get, set y un método para mostrar por pantalla toda la información de la cuenta. Además tendrá dos métodos especiales:

ingresar(double cantidad): se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.

retirar(double cantidad): se retira una cantidad a la cuenta, si restando la cantidad actual a la que nos pasan es negativa, la cantidad de la cuenta pasa a ser 0.

Ejercicio 3:

Vamos a realizar una clase llamada Raices, donde representaremos los valores de una ecuación de 2º grado.

Tendremos los 3 coeficientes como atributos, llamémosles a, b y c.

Hay que insertar estos 3 valores para construir el objeto.

Las operaciones que se podrán hacer son las siguientes:

obtenerRaices(): imprime las 2 posibles soluciones

obtenerRaiz(): imprime única raíz, que será cuando solo tenga una solución posible.

getDiscriminante(): devuelve el valor del discriminante (double), el discriminante tiene la siguiente formula, (b^2)-4*a*c

tieneRaices(): devuelve un booleano indicando si tiene dos soluciones, para que esto ocurra, el discriminante debe ser mayor o igual que 0.

tieneRaiz(): devuelve un booleano indicando si tiene una única solución, para que esto ocurra, el discriminante debe ser igual que 0.

calcular(): mostrara por consola las posibles soluciones que tiene nuestra ecuación, en caso de no existir solución, mostrarlo también.

Formula ecuación 2º grado: (-b±√((b^2)-(4*a*c)))/(2*a)

Ejercicio 4:

Haz una clase llamada Persona que siga las siguientes condiciones:

Sus atributos son: nombre, edad, DNI, sexo (H hombre, M mujer), peso y altura. Piensa cual es el tipo más adecuado. 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 será mujer por defecto, usa una constante para ello.

Los métodos que se implementaran son:

calcularIMC(): calculara si la persona esta en su peso ideal (peso en kg/(altura^2 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. Devolverá un booleano

mostrar_información(): devuelve toda la información del objeto.

generaDNI(): genera un número aleatorio de 8 cifras, genera a partir de este su número su letra correspondiente. Este método será invocado cuando se construya el objeto. Puedes dividir el método para que te sea más fácil.

Métodos get y set de cada parámetro, excepto de DNI.

Ahora, crea una clase ejecutable que haga lo siguiente:

Pide por teclado el nombre, la edad, sexo, peso y altura.

Crea 3 objetos de la clase anterior, obteniendo los datos por teclado.

Para cada objeto, deberá comprobar si está en su peso ideal, tiene sobrepeso o por debajo de su peso ideal y mostrar un mensaje.

Indicar para cada objeto si es mayor de edad.

Por último, mostrar la información de cada objeto.

Ejercicio 5:

Haz una clase llamada Password que siga las siguientes condiciones:

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

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 por defecto.

Ahora, crea un archivo main:

Crea una lista de Passwords con el tamaño que tú le indiques por teclado.

Crea un bucle que cree un objeto para cada posición de la lista.

Indica también por teclado la longitud de los Passwords (antes de bucle).

Crea otra lista de booleanos donde se almacene si el password de la lista de Password es o no fuerte (usa el bucle anterior).

Al final, muestra la contraseña y si es o no fuerte. Usa este simple formato:

contraseña1 valor_booleano1

contraseña2 valor_bololeano2

…

Ejercicio 6:

Crearemos una clase padre llamada Electrodomestico con las siguientes características:

Sus atributos son precio base, color, consumo energético (letras entre A y F) y peso.

Por defecto, el color será 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 está en mayúsculas o en minúsculas.

Los métodos que implementara serán:

comprobarConsumoEnergetico(char letra): comprueba que la letra es correcta, sino es correcta usara la letra por defecto.

comprobarColor(String color): comprueba que el color es correcto, sino lo es usa el color por defecto.

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 clase hija llamada Lavadora con las siguientes características:

Su atributo es carga, además de los atributos heredados.

Por defecto, la carga es de 5 kg. Usa una constante para ello.

Los métodos que se implementara serán:

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 clase hija llamada televisión con las siguientes características:

Sus atributos son resolución, tamaño (en pulgadas) y sintonizador TDT (booleano), además de los atributos heredados.

Por defecto, el tamaño será de 20 pulgadas y el sintonizador será false.

Los métodos que se implementara serán:

precioFinal(): si tiene una resolución mayor de 40 pulgadas, se incrementara el precio un 30% y si tiene un sintonizador TDT incorporado, aumentara 50 €. Recuerda que las condiciones que hemos visto en la clase Electrodomestico también deben afectar al precio.

Ahora crea un main que realice lo siguiente:

Crea una lista 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

Por ejemplo, si tenemos un electrodoméstico 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.

Ejercicio 7:

Crearemos una clase llamada Serie con las siguientes características:

Sus atributos son título, número de temporadas, entregado, género y creador.

Por defecto, el número de temporadas es de 3 temporadas y entregado false.

Los métodos que se implementara serán:

Crea un método mostrar información que mostrará toda la información de la Serie

Crearemos una clase Videojuego con las siguientes características:

Sus atributos son título, horas estimadas, entregado, género 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 métodos que se implementara serán:

Crea un método para mostrar toda la información del objeto.

Crea una clase padre a ambas, en dicha clase deberás implementar unos métodos, piensa si además debes realizar algún cambio más:

entregar(): cambia el atributo prestado a true.

devolver(): cambia el atributo prestado a false.

isEntregado(): devuelve el estado del atributo prestado.

Ahora crea un main y realiza lo siguiente:

Crea dos Lista, uno de Series y otro de Videojuegos, de 5 posiciones cada uno.

Crea un objeto en cada posición de la lista, con los valores que desees.

Entrega algunos Videojuegos y Series con el método entregar().

Cuenta cuantos Series y Videojuegos hay entregados. Al contarlos, muéstralos por pantalla.

Por último, indica el Videojuego tiene más horas estimadas y la serie con más temporadas. Muéstralos en pantalla con toda su información.

Ejercicio 8:

Nos piden realizar una agenda telefónica de contactos.

Un contacto está definido por un nombre y un teléfono (No es necesario de validar). Un contacto es igual a otro cuando sus nombres son iguales.

Una agenda de contactos está formada por un grupo de contactos

Los métodos de la agenda serán los siguientes:

añadirContacto(Contacto c): Añade un contacto a la agenda. No se pueden meter contactos que existan, es decir, no podemos duplicar nombres, aunque tengan distinto teléfono.

existeContacto(Conctacto c): indica si el contacto pasado existe o no.

listarContactos(): Lista toda la agenda

buscaContacto(String nombre): busca un contacto por su nombre y muestra su teléfono.

eliminarContacto(Contacto c): elimina el contacto de la agenda, indica si se ha eliminado o no por pantalla

Crea un menú con opciones por consola para probar todas estas funcionalidades.