Trabajo Práctico N°2 (C# Basics)

Fecha de entrega: 14/09/2023
enviar código (archivos .cs) a: marcos.rivero@unsta.edu.ar

1. Crear una clase llamada Cuenta, esta debe poseer los siguientes atributos: titular y cantidad. Crear dos constructores: uno que reciba dos parámetros y el otro que también reciba dos pero que uno de ellos sea opcional.
    a. Esta clase tendrá dos métodos especiales:
        - Depósito(decimal cantidad): Se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada. Si es positiva, sumar al valor actual.
        - Retiro(decimal cantidad): Si el valor pasado como parámetro es positivo, se restará de la cantidad actual, si es negativo no hacer nada, si el valor recibido es mayor a la cantidad actual, la cantidad actual de la cuenta pasa a ser 0.

2. Hacer una clase llamada Persona que siga las siguientes condiciones:
    a. Sus atributos son: nombre, edad, DNI, sexo (H hombre, M mujer, N no binario), peso y altura. Si quieres añadir algún atributo puedes hacerlo.
    b. Se implementaran varios constructores:
        i. Un constructor por defecto.
        ii. Un constructor con el nombre, edad y sexo.
        iii. Un constructor con todos los atributos como parámetro.
    c. Los métodos que se implementarán son:
        i. calcularIMC(): calculará si la persona está 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, significa que está por debajo de su peso ideal), si devuelve un número entre 20 y 25 (incluidos), significa que está en su peso ideal por lo que la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1.
        ii. esMayorDeEdad(): indica si es mayor de edad (mayor de 18), devuelve un booleano.
        iii. validarSexo(char sexo): comprueba que el sexo introducido es correcto. Si no es correcto, será H. No será visible al exterior.
        iv. toString(): devuelve toda la información del objeto.
    d. Ahora, crea Unit Tests para comprobar que todos los métodos hagan lo que tengan que hacer.

3. Crearemos una clase abstracta llamada Electrodomestico con las siguientes características:
    a. Sus atributos son precio base, color (blanco, negro, rojo, azul, verde), consumo energético (letras entre A y F) y peso.
    b. Los métodos que definirá esta clase serán:
        i. validarConsumoEnergetico(char letra): comprueba que la letra es correcta (A - F).
        ii. comprobarColor(String color): comprueba que el color es correcto.

iii. precioFinal(): según el consumo energético, aumentará 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

c. Crearemos una subclase llamada Lavadora con las siguientes características:
   i. Su atributo es carga, además de los atributos heredados.
   ii. Los constructores que se implementarán serán:
      1. Un constructor por defecto.
      2. Un constructor con el precio y peso.
      3. Un constructor con la carga y el resto de atributos heredados.
   iii. Los métodos que se implementara serán:
      1. precioFinal():, si tiene una carga mayor de 30 kg, aumentará el precio 50 €, sino es así no se incrementará el precio. Llama al método padre y añade el código necesario. Recuerda que las condiciones que hemos visto en la clase Electrodoméstico también deben afectar al precio.
d. Crearemos una subclase llamada Televisión con las siguientes características:

i. Sus atributos son resolución (en pulgadas), además de los atributos heredados.
ii. Los constructores que se implementarán serán:
    1. o Un constructor por defecto.
    2. o Un constructor con el precio y peso. La resolución será opcional.
    3. Un constructor con la resolución y el resto de atributos heredados.
iii. Los métodos que se implementara serán:
    1. precioFinal(): si tiene una resolución mayor de 40 pulgadas, se incrementará el precio un 30%. Recuerda que las condiciones que hemos visto en la clase Electrodomestico también deben afectar al precio.

e. Ahora crear unit tests que validen los métodos anteriores de todas las clases.