Ejercicio 1 clases

Crea la clase hora Hora

◦ Hora: entero

◦ Minutos: entero

◦ Segundos: entero

• Constructor con 3 parámetros: hora, minutos y segundos

• Constructor con dos parámetros: hora y minutos

• Constructor con un parámetro: hora

• Constructor de copias

Getters y setters que correspondan y comprueben que los segundos y los ministros estén entre 0 y 59 en caso que no quedarse con el resto

Método toString() con formato hh:mm:ss

Método público setLaHora(int hora, int minutos, int segundos) que cambia el valor de los

atributos que corresponden a la hora, minutos y segundos

La comparación está basada en la hora. Si las horas son iguales, la comparación está basada en los minutos. Y si los minutos son iguales la comparación está basada en los segundos.

compararCon (Hora h)

Devuelve negativo si este objeto Hora es menor que el objeto Hora h pasado cómo

parámetro, de lo contrario devuelven positivo si este objeto Hora fecha es mayor que el

objeto Hora h pasado cómo parámetro, de lo contrario devuelve cero si ambos objetos

(este y h) son iguales.

Ejercicio 2 herencia

Producto() : Se necesita esta clase para que sea plantilla de las demás, por lo tanto contendrá variables y métodos que serán

comunes para el resto de clases que la hereden.

Esta clase contiene los atributos:

nomComercial: texto que identifica la marca del producto.

contenido\_envase: (formato) el cash solo trabaja con 2L, 1L, 0,5L y 0,33L.

embalaje: los envases que utiliza el cash pueden contener desde 1 a 48 envases como máx.

precioSinIva: valor decimal del producto antes de aplicarle el IVA correspondiente.

Todos los productos tendrá el método calcularprecioVenta() que equivale a precioSinIva+IVA aunque el porcentaje

de iva depende del producto.

Bebida ():Clase que hereda de Producto(). No instancia objetos. Sus Atributos son:

caducidad : variable de formato cadena que guarda la fecha de caducidad del embalaje. Su formato será

DD/MM/AAAA; donde los días serán >0 y <32, los meses >0 y < 13.

ANNO : variable fija que contendrá el valor del año en curso, en este caso 2021. Esta variable será con la que se

comparará el AAAA introducido en la variable “caducidad”, y que tendrá que ser siempre mayor a ANNO y no

superar en 5 a este para evitar errores mayores.

Cerveza ():Clase que hereda de Bebidas y de la interfaz Generable

Esta clase contiene las atributos:

secuenciaIDCerveza=120 : Variable inicializada con el último código que se le puso al último objeto instanciado

de la clase Cerveza(). Esta variable es un número entero que se irá incrementando durante la ejecución del

proyecto por cada objeto diferente que se instancie de Cerveza() y en este proyecto tiene valor inicial de 120.

(0,10puntos).

IDCerveza : Cadena que identifica a cada objeto instanciado y que consta de la concatenación del atributo

'secuenciaIDCerveza' más la palabra 'cerveza'; el primer producto de la clase cerveza será 121cerveza, el segundo

122cerveza, el tercero 123cerveza, ... hasta que deje de ejecutarse el proyecto.

Alcohol : Variable booleana, donde 'true' indica que tiene alcohol y false que no.

IvaC: Variable fija 21, que es el 21% de iva que lleva la cerveza independientemente de si tiene alcohol o no.

tasa : Valor entero, que refleja el porcentaje de impuestos que lleva aplicado el precio. Este es 0% en aquellas

cervezas que no tienen alcohol y del 26% en las que sí lo tienen.

Refresco (): Clase que hereda de Bebidas y de la interfaz Generable

Esta clase contiene las atributos:

secuenciaIDRefresco=500 : Variable inicializada con el último código que se le puso al último objeto instanciado

de la clase Refresco(). Esta variable es un número entero que se irá incrementando durante la ejecución del

proyecto por cada objeto diferente que se instancie de Refresco() y en este proyecto tiene valor inicial de 500.

IDRefresco : Cadena que identifica a cada objeto instanciado y que consta de la concatenación del atributo

'secuenciaIDRefresco' más la palabra 'refresco'; el primer producto de la clase refresco será 501refresco, el

segundo 502refresco, el tercero 503refresco, ... hasta que deje de ejecutarse el proyecto.

azucar : Variable booleana, donde 'true' indica que tiene azucares añadidos y false que no.

IvaR: Variable entera que tendrá valor, 10 para aquellos refrescos sin azucar añadido y el 21 para los que contengan

azucares o edulcorantes.

Interface :

Generable(): esta interface tan solo tiene un único método : genararIDProducto(), cuya función será, en cada clase donde se

implemente, generar el identificador del producto que se instancia