

Modelo Primer Parcial

Laboratorio II C# - 2016

Se desea realizar el control de comprar y ventas de un comercio, para eso deberemos crear las siguientes clases en un proyecto de tipo Class Library (Entidades):

Articulo
Class

Fields

`_codigo : int`

`_nombre : string`

`_precioCosto : float`

`_precioVenta : float`

`_stock : int`

Properties

`NombreYCodigo { get; } : string`

`PrecioCosto { set; } : float`

`PrecioVenta { get; } : float`

`Stock { set; } : int`

Methods

`Articulo(int codigo, string nombre, float precioCosto, int cantidad)`

`HayStock(int cantidad) : bool`

`operator !=(Articulo articuloUno, Articulo articuloDos) : bool`

`operator -(Articulo articuloUno, int cantidad) : int`

`operator +(Articulo articuloUno, Articulo articuloDos) : int`

`operator ==(Articulo articuloUno, Articulo articuloDos) : bool`

Venta
Class

Fields

`_articuloVendido : Articulo`

`_cantidad : int`

Methods

`RetornarGanacia() : float`

`Venta(Articulo articuloVendido, int Cantidad)`

Comercio
Class

Fields

`_dueño : string`

`_misArticulos : List<Articulo>`

`_misVentas : List<Venta>`

Methods

`Comercio(string dueño)`

`ComprarArticulo(Articulo articuloComprado) : void`

`MostrarArticulos(Comercio ComercioAMostrar) : void`

`MostrarGanacia(Comercio ComercioParaResumen) : void`

`VenderArticulo(Articulo articuloSolicitado, int cantidad) : void`

1) Propiedades:

NombreYCodigo: sólo retorna un String que tienen concatenado el nombre y el código del Artículo.

PrecioCosto: Solo permite asignar un nuevo precio de Costo al artículo y asigna el precio de venta sumándole un 30% al costo.

PrecioVenta: asociada al atributo `_precioVenta`.

Stock: sólo permite asignar un nuevo Stock.

2) Constructores: realizar los constructores para cada clase (**Artículo, Venta, Comercio**) con los parámetros que se detallan en la imagen.

3) Método **RetornarGanancia:** retorna el dinero que obtuvimos por la venta de la cantidad del Artículo por el precio del mismo.

4) Método **HayStock:** retorna un booleano indicando si el Stock de ese Artículo es mayor o igual a la cantidad que recibe por parámetro.

5) **Operador == (Igual):** retornara un booleano informando si el nombre y el código de los artículos coinciden al mismo tiempo.

6) Método estático **MostrarArticulos:** Recorre la lista de artículos de mi comercio y muestra el nombre y el código de cada artículo.

7) Método estático **MostrarGanacia:** muestra la cantidad total de ganancia que se obtuvo en las ventas realizadas en el comercio.

8) **operador + (Mas):** retorna un número entero que es el resultado de la suma del stock de los Artículos que recibe por parámetro.

Ej. Utilización:

```
ArticuloDeLaLista.Stock = ArticuloDeLaLista + articuloComprado;
```

9) **operador - (menos):** retorna un número entero que es el resultado de la resta del stock del Artículo menos la cantidad recibida por parámetro.

Ej. Utilización:

```
ArticuloDeLaLista.Stock = ArticuloDeLaLista - cantidad;
```

10) Método **ComprarArticulo:** si el artículo ya existe en el comercio, solo limitaremos la acción a incrementar el stock del artículo existente, de lo contrario agregaremos este producto que recibimos por parámetro a la lista de artículos, (para comparar los artículos utilizaremos el operador "==" "y para incrementar el stock utilizaremos el operador "+" los dos sobrecargados anteriormente).

11) Método **VenderArticulo:** en este método se recibirá un artículo y una cantidad, si el artículo existe y su Stock es mayor o igual a la cantidad solicitada, esta cantidad será descontada del stock del artículo de nuestra lista, y se deberá registrar la venta en la lista de ventas de mi comercio. (Para comparar los artículos utilizaremos el operador "==" "y para disminuir el stock utilizaremos el operador "-" los dos sobrecargados anteriormente).

12) Las clases deben estar en distintos NameSpaces. Se valorará el buen uso de las convenciones al nombrar/invocar a atributos, métodos, etc.

Este es el Main (el alumno lo debe copiar y no se podrá modificar):

```
class Program
{
    static void Main(string[] args)
    {
        Comercio ElBolicheDeToni = new Comercio("TONI");

        Articulo articuloUno = new Articulo(100, "Sprite", (float)2.5, 10);
        Articulo articuloDos = new Articulo(101, "Fanta", (float)2.5, 10);
        Articulo articuloTres = new Articulo(100, "Sprite", (float)2.5, 25);
        Articulo articuloCuatro = new Articulo(103, "Quattro", (float)2.5, 10);
        Articulo articuloCinco = new Articulo(103, "Pepsi", (float)2.5, 10);

        ElBolicheDeToni.ComprarArticulo(articuloUno);
        ElBolicheDeToni.ComprarArticulo(articuloDos);
        ElBolicheDeToni.ComprarArticulo(articuloTres);
        ElBolicheDeToni.ComprarArticulo(articuloCuatro);
        Console.WriteLine("El Boliche De Tony");
        Console.WriteLine("Vender:");
        ElBolicheDeToni.VenderArticulo(articuloUno, 1);
        ElBolicheDeToni.VenderArticulo(articuloDos, 1);
        ElBolicheDeToni.VenderArticulo(articuloTres, 1);
        ElBolicheDeToni.VenderArticulo(articuloDos, 1);
        ElBolicheDeToni.VenderArticulo(articuloTres, 20);
        ElBolicheDeToni.VenderArticulo(articuloTres, 50);
        ElBolicheDeToni.VenderArticulo(articuloCuatro, 1);
        ElBolicheDeToni.VenderArticulo(articuloCinco, 1);
        Console.WriteLine();

        Comercio.MostrarArticulos(ElBolicheDeToni);
        Console.WriteLine();
        Comercio.MostrarGanacia(ElBolicheDeToni);
        Console.ReadLine();
    }
}
```

Resultado de consola:

```
El Boliche De Tony
Vender:
El siguiente Producto no tiene stock para la venta!!!
100--SPRITE
El siguiente Producto no Existe en nuestro Comercio !!!
103--PEPSI

Los productos existentes son
100--SPRITE
101--FANTA
103--QUATTRO

la ganancia obtenida es de : 81,25
-
```