

Ejercicio 9

Realizar una aplicación que efectúe las siguientes operaciones:

1. Cree una colección que pueda contener objetos de las siguientes clases (la clave del Collection será codigoArticulo):

CLASE ARTÍCULO:	CLASE ARTICULO REBAJADO:
codigoArticulo (string)	Hereda de la clase ARTÍCULO
nombreArticulo(string)	y añadimos la propiedad
categoriaArticulo(string): Informática, Imagen, Telefonía, Sonido	rebajaArticulo(short)
precioArticulo(double)	
existenciasArticulo(integer)	

2. Para cada clase crear un constructor sin argumentos y otro donde le pasemos por argumento todas las propiedades. (Llamamos al constructor de la clase base con MyBase.New()). Crear las funciones necesarias para acceder a las propiedades.

3. Crear una función de actualización de existencias. Al pasarle la cantidad de pedido compruebe si hay suficientes existencias, en caso afirmativo nos mostrará el importe del pedido y actualizará las existencias restándole el valor del pedido y en caso negativo nos informa de que no hay suficientes existencias. En ambos casos nos informará de las existencias que quedan.

- 4.- Crear función imprimirArticulo a la que llamaremos a la hora de hacer los listados. La clase articuloRebajado sobrecargara (Overrides) la función de la clase base (Overridable) y añadirá el campo rebajaArticulo. (Llamamos a una función de la clase base con MyBase.nombreFuncion()).

5. Al hacer un **Alta** mediante un radioButton se indicará si se trata de la clase Artículo o ArtículoRebajado. Introducir los datos mediante cajas de texto, menos en la categoría la cual se seleccionará de las opciones mostradas en un comboBox. Haremos depuración de datos antes de pasarlos a la colección.

6. El menú **Guardar** añadirá el articulo a la colección si los datos son correctos, luego nos preguntará si deseamos crear otro artículo, abriéndose un MsgBox con botones Yes y No. En caso afirmativo se borrará el contenido de las cajas de texto, en caso negativo se dejarán de visualizar.

7. **Consulta/Baja** un artículo determinado. Se accederá al artículo, si existe se visualizarán sus datos y se visualizará botón Eliminar y Modificar. **Modificar** actualizará los datos (solo se puede modificar el precio y las existencias).

Eliminar elimina el artículo previa confirmación.

8. **Listar**. Se podrán mostrar los siguientes listados en otro formulario, dentro de una RichTextbox. En todos los casos se imprimirá cada linea del pedido llamando a la función **imprimirArticulo** (creada en punto 4):

- a) **Todos** los artículos .
- b) Artículos **en mínimos**, que serán aquellos articulos que tengan menos de 10 artículos en existencias.
- c) Artículos **por categoría**, se seleccionará la categoría de un ComboBox y se listarán todos los artículos de esa categoría.

Al mostrarse cualquiera de los listados se mostrará el botón **Guardar** para poder guardarlo como archivo de texto, facilitándose la tarea mediante la caja de diálogo común Guardar. También podremos **Abrir** (incluido en menú listado) cualquiera de los listados existentes usando también caja de diálogo Abrir.

- 9.- En menú **Pedido** se mostrará **Pedir artículo** y **Generar Albarán** con las siguientes funciones:

9.a- En menú **Pedir artículo** se realizará el pedido de un articulo indicando su código y la cantidad pedida. Una vez comprobado que existe el artículo se ejecutará la función de actualización de existencias y se añadirá la linea de pedido al albarán.

9.b- En menú **Generar Albarán** se mostrará por pantalla el Albarán, con n.º de albarán, que se irá incrementando de forma automática, y con todos los artículos pedidos. Se mostrará el botón Generar y al hacer click se guardará como archivo de texto ese albarán con el nombre del n.º de albarán. También podremos abrir para leer ese archivo, en ambos caso se utilizaran las cajas de diálogo.

- 10.- El botón Cancelar cancelará la operación que se esté realizando. Acerca de mostrará un mensaje con el autor del ejercicio y la versión.

