

Tecnología y paradigmas de programación. Laboratorio 8. Grupos 2 y 3.

En este laboratorio se va a utilizar el modelo de datos proporcionado en el fuente que acompaña a este documento, modificación de un [ejemplo del MSDN](#). Se tienen cuatro clases: ProductWithOutVat, Category, Vat y Cart, que representan un producto, la categoría a la que pertenece, el IVA que le corresponde y una cesta con productos. Se proporcionan tres listas con datos: productsWithVat, categories y currentVat, con las claves necesarias para relacionarlas. El constructor por defecto del modelo rellena la cesta con algunos productos, se puede cambiar al gusto para hacer pruebas.

1. Listado de nombre de productos con precio menor que uno dado.
2. Listado de nombre de productos, precios y valor del IVA aplicable.
3. Listado de nombre de productos y precios con IVA aplicados.
4. Listado de nombre de productos, nombre de categoría y nombre de tipo de IVA.
5. El mismo listado que en el punto anterior, ordenado por nombre de categoría.
6. El mismo listado agrupado por tipo de IVA. Para cada tipo de IVA mostrar los productos a los que se le aplica y su categoría (la del producto). En este caso usar dos foreach (anidados) en el método que realiza la consulta, en lugar de usar el ForEach extensor.
7. Repetir el ejercicio agrupando por categoría de producto y para cada una de ellas mostrar cuantos productos hay (usar .Count()), el nombre de los productos que pertenecen a esa categoría y el tipo de IVA que se les aplica. En este caso no es evidente usar ForEach, se pueden usar dos foreach (anidados) en el método que realiza la consulta, en su lugar.
8. Precio medio de los productos, IVA incluido.
9. Precio medio de los productos con precio menor o igual a 100 euros.
10. Valor del precio del producto con el precio más alto. La redacción es rara, representa una forma de calcularlo, se busca el registro con el precio más alto y se devuelve el valor del precio o bien se devuelve el registro completo y se muestra el valor de su precio. Esta forma evita inicializar el valor de acumulador. También es posible trabajar directamente con el precio en este caso, los precios son siempre mayores que cero.
11. Nombre y precio sin IVA del producto con el precio más bajo con el IVA aplicado.
12. Listado de productos de una categoría determinada (cada producto debe de ser un objeto de esa clase, no es de un tipo anónimo), especificada por el nombre de la categoría.
13. El diccionario cart representa una compra. Por simplicidad consideramos que se compra un cierto número de unidades de algunos productos, representado este número por el valor (int) indexado por cada clave (el producto). Crear una consulta sobre ese diccionario que envía a la consola el listado de: nombre de producto, precio incluyendo el IVA y número de unidades para cada producto.
14. Calcular el importe de la compra, recordar incluir el IVA.
NOTA: cada elemento del diccionario es un par Key/Value. Key es de tipo ProductWithOutVat (por lo tanto se puede usar para acceder al precio y su IVA). Value es int y almacenará el número de unidades del producto.
15. Mostrar el nombre del producto de la cesta del que se han comprado más unidades en esta compra.
16. Mostrar el nombre del producto de la cesta en el que se ha gastado más dinero en esta compra.