

Programação Visual

Trabalho de Laboratório Semana 7

Objetivo	Procurar informação numa lista de produtos usando LINQ.
Regras	Criar uma aplicação de consola. Implementar o código necessário e testar no fim de cada nível. Use as convenções de codificação adotadas para a linguagem C#.
Nível 1	<ol style="list-style-type: none"> 1. Descarregue do moodle e abra a solução LABS07_Materiais. Na solução que abriu, no método Main é criada a lista de produtos que pretendemos analisar. Os produtos são objetos da classe Produto onde estão definidas propriedades para a identificação do produto (ProductId), o nome do produto (ProductName), a categoria do produto (Category) o preço unitário (UnitPrice) e o número de unidades em Stock (UnitsInStock). No método Main é feita uma listagem de todos os produtos. 2. Imprima no método Main a string <code>"\nNível 1*****"</code> 3. Acrescente à classe Produto um método ToString() que mostre a informação do produto. 4. Usando uma <i>query</i> LINQ liste todos os produtos da lista. 5. Usando uma <i>query</i> LINQ liste todos os produtos que não existem em Stock. 6. Liste agora todos os nomes (apenas os nomes) dos condimentos por ordem alfabética.
Nível 2	<ol style="list-style-type: none"> 1. Imprima no método Main a string <code>"\nNível 2*****"</code> 2. Crie uma <i>query</i> LINQ para obter a lista de bebidas que custam menos de 25€ ordenadas pelo preço. 3. Acrescente à lista os seguintes produtos: 78, Sagres, Beverages, 12.2, 25 e 79, Super Bock, Beverages, 10.1, 15 4. Faça novamente a listagem reutilizando a <i>query</i> anterior (não volte a criá-la). 5. Acrescente o produto: Carlsberg, Beverages, 18.1, 12 <p>Nota: O número deste produto deve ser determinado automaticamente. Ou seja, obtenha o maior número de produto da lista usando LINQ e incremente-o uma unidade.</p>
Nível 3	<ol style="list-style-type: none"> 1. Imprima no método Main a string <code>"\nNível 3*****"</code> 2. Mostre o produto mais barato, o mais caro e valor médio dos produtos usando LINQ. 3. Pretende-se saber quanto dinheiro se tem em produtos no stock. Neste sentido obtenha uma listagem com todos os produtos mostrando em cada linha o produto e o valor total do seu stock (obtido multiplicando o número de unidades em stock pelo custo unitário). Neste caso utilize LINQ com projeção de dados, os objetos criados deverão ter as propriedades Nome e ValorTotal.
Nível 4	<p>Com a sintaxe dos métodos de extensão e expressões lambda faça as seguintes listagens em LinQ:</p> <ol style="list-style-type: none"> 1. A lista de produtos ordenada pelo preço por ordem ascendente. 2. A lista de categorias únicas (sem repetição) (operador Distinct) 3. Verifique se existem produtos com um custo maior que 200 (operador Any). 4. O número médio de unidades em stock.
Nível 5	<p>Usando LINQ liste:</p> <ol style="list-style-type: none"> 1. Os primeiros 3 produtos que custam mais de 50 (operador Take) 2. O quarto e o quinto produto começado pela letra C (operadores Take e Skip) 3. A lista de produtos ordenada pela categoria e depois pelo nome do produto (operadores OrderBy e ThenBy). 4. Mostre cada categoria seguida da lista de produtos incluídos na mesma (GroupBy).
Nível 6	<ol style="list-style-type: none"> 1. Crie uma classe Produtos que herda de List<Produto>. 2. No método Main copie todos os produtos para um objecto da classe Produtos 3. Na classe Produtos crie métodos para todas as queries que definiu nos níveis 1 a 5. 4. Experimente-as no método Main
Nível 7	<ol style="list-style-type: none"> 5. Defina os métodos Equals e GetHashCode para a classe Produto 6. Crie uma classe Stock que seja um Dictionary<Produto, int> em que a chave é o Produto e o valor a quantidade de produtos existentes em stock 7. Defina um método Venda que aceite um Produto e uma quantidade e remova do stock essa quantidade do produto.