

Programação Visual

Trabalho de Laboratório nº 2

Objetivo	Programação avançada em C# utilizando a linguagem de interrogação LINQ
Programa	Analisar uma lista de Aeroportos
Regras	<p>Criar uma aplicação de consola.</p> <p>Implementar o código necessário e testar ao fim de cada nível.</p> <p>Use as convenções de codificação adotadas para a linguagem C#.</p> <p>Na classe do programa não coloque atributos nem crie nenhum método para além do Main.</p> <p>Não é necessário obter dados do utilizador. Forneça os dados ao nível do código.</p>

Nível 1

- Abra a solução **AirportAnalysis.sln** que acompanha este enunciado. Na solução que abriu, dentro do método **Main** é obtida a lista de aeroportos que se pretendem analisar. Os aeroportos são objetos da classe **Airport** onde estão definidas propriedades implícitas para: a identificação do aeroporto (**AirportId**), o código do aeroporto (**ICAO**), o país do aeroporto (**Country**), a cidade do aeroporto (**City**), a indicação de se o aeroporto pretence ao espaço Schengen (**IsSchengen**) e a média de voos diários do aeroporto (**DailyFlightAverage**). No método **Main** é feita uma listagem de todos os aeroportos.
- Usando uma query LINQ liste todos os aeroportos localizados no espaço Schengen por ordem alfabética do seu código (ICAO).
- Liste agora todos os aeroportos cujo código começa pela letra 'L'.
- Liste todos os aeroportos localizados no espaço Schengen cujo nome começa pela letra 'L'.

Nível 2

- Crie uma query LINQ para obter a lista de aeroportos que estão no United Kingdom e têm mais de 5000 voos diários.
- Dentro do método **Main**, acrescente à lista de aeroportos os seguintes registos:
 1. 46, EGCC, United Kingdom, MANCHESTER, true, 20000
 2. 47, EGJJ, United Kingdom, JERSEY, true, 10000
- Faça novamente a listagem reutilizando a query anterior e verifique que a lista aparece atualizada.
- Usando LINQ, obtenha o número de voos diários (**DailyFlightAverage**) mais alto dos aeroportos existentes na lista e escreva o seu ICAO e o número de voos diários no ecrã.

Programação Visual

Trabalho de Laboratório nº 2

Nível 3

- Pretende-se saber quantos voos diários, em média, ocorrem em cada país. Neste sentido obtenha uma listagem com todos os países mostrando em cada linha o nome do país e a quantidade média de voos diários nesse país (pela soma do número médio de voos diários de cada aeroporto desse país). Neste caso utilize LINQ com projeção de dados, os objetos criados deverão ter as propriedades **Country** e **DailyFlights**.

Nota: Para a soma utilize o operador **Sum()** do LINQ.

- Usando novamente os operadores de agregação do LINQ, mostre a seguir o aeroporto com maior média diária de voos e com menor média diária de voos.

Nível 4

- Usando LINQ com a sintaxe dos métodos de extensão e expressões lambda (quando for necessário) faça as seguintes apresentações:
 - A lista de aeroportos ordenada pela cidade (operador **OrderBy**)
 - A lista de países sem repetição (operador **Distinct**)
 - Verificar se existem aeroportos com um número médio diário de voos inferior a 5000 (operador **Any**).
 - A média do número de voos diários de todos os aeroportos.

Nível 5

- Usando LINQ com a sintaxe dos métodos de extensão:
 - Os primeiros 5 aeroportos fora do espaço Schengen com mais de 100000 voos diários ordenados pelo número de voos diários (operador **Take**)
 - Os 5 primeiros aeroportos que aparecem a seguir aos do ponto anterior (operadores **Take** e **Skip**).
 - A lista dos aeroportos Alemães (Country = Germany) ordenada pela cidade e, caso existam mais do que um aeroporto para a mesma cidade, pelo ICAO (operadores **OrderBy** e **ThenBy**).

Desafio

- Mostre cada cidade Alemã presente na lista fornecida seguida da lista de aeroportos localizados nessa cidade e omitindo o nome da cidade e do país na listagem (utilize o operador **GroupBy**).

Notas

Para os identificadores siga as convenções adotadas pelo C#, nomeadamente:

- A notação camelCase para o nome das variáveis locais e identificadores privados.
- A notação PascalCase para os nomes públicos dos métodos, classes e interfaces.
- Não utilize o símbolo '_' nos identificadores nem abreviaturas