using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CursoCSharpBackup.Exercicios

{

/\* Exercício prático sobre LINQ

Crie uma consulta LINQ para encontrar todos os nomes das pessoas que têm o nome começando com a letra 'A' e que têm mais de 25 anos. \*/

/\*

public class Pessoa

{

public string Nome { get; set; }

public int Idade { get; set; }

}

public class Program

{

public static void Main()

{

var pessoas = new List<Pessoa>

{

new Pessoa { Nome = "Ana", Idade = 30 },

new Pessoa { Nome = "Bruno", Idade = 22 },

new Pessoa { Nome = "Alice", Idade = 28 },

new Pessoa { Nome = "Carlos", Idade = 35 }

};

var nomesComA = pessoas

.Where(p => p.Nome.StartsWith("A") && p.Idade > 25)

.Select(p => p.Nome)

.ToList();

// Exibindo os resultados

foreach (var nome in nomesComA)

{

Console.WriteLine(nome);

}

}

}

\*/

/\* Adicione uma lista de produtos com propriedades Nome e Preço. Faça uma consulta LINQ para encontrar os produtos cujo preço é superior à média de todos os produtos.

public class Produto

{

public string Nome { get; set; }

public decimal Preco { get; set; }

}

public class Program

{

public static void Main()

{

var produtos = new List<Produto>

{

new Produto { Nome = "Notebook", Preco = 1200.00M },

new Produto { Nome = "Mouse", Preco = 50.00M },

new Produto { Nome = "Teclado", Preco = 100.00M },

new Produto { Nome = "Monitor", Preco = 2000.00M },

new Produto { Nome = "Cadeira", Preco = 80.00M }

};

var precoMedio = produtos.Average(p => p.Preco);

var produtosComPrecoSuperiorAoMedio = produtos

.Where(p => p.Preco > precoMedio)

.ToList();

// Exibindo os resultados

Console.WriteLine("Produtos com preço superior à média:");

foreach (var produto in produtosComPrecoSuperiorAoMedio)

{

Console.WriteLine($"{produto.Nome}, Preço: {produto.Preco}");

}

}

}

\*/

/\* Utilizando a lista de pessoas, crie uma consulta LINQ para listar o nome das pessoas em ordem alfabética reversa.

public class Pessoa

{

public string Nome { get; set; }

public int Idade { get; set; }

}

public class Program

{

public static void Main()

{

var pessoas = new List<Pessoa>

{

new Pessoa { Nome = "Davi", Idade = 21 },

new Pessoa { Nome = "Arthur", Idade = 29 },

new Pessoa { Nome = "Thiago", Idade = 21 },

new Pessoa { Nome = "Marcao", Idade = 19 }

};

var nomesOrdenados = pessoas

.Select(p => p.Nome)

.OrderByDescending(n => n)

.ToList();

Console.WriteLine("Nomes em ordem alfabética reversa:");

foreach (var nome in nomesOrdenados)

{

Console.WriteLine(nome);

}

}

}

\*/

/\* Crie uma consulta LINQ para encontrar a pessoa mais jovem de cada cidade.

public class Pessoa

{

public string Nome { get; set; }

public int Idade { get; set; }

public string Cidade { get; set; }

}

public class Program

{

public static void Main()

{

var pessoasPorCidade = new List<Pessoa>

{

new Pessoa { Nome = "Ana", Idade = 30, Cidade = "São Paulo" },

new Pessoa { Nome = "Bruno", Idade = 22, Cidade = "São Paulo" },

new Pessoa { Nome = "Carlos", Idade = 35, Cidade = "Rio de Janeiro" },

new Pessoa { Nome = "Diana", Idade = 25, Cidade = "Rio de Janeiro" }

};

var maisJovensPorCidade = pessoasPorCidade

.GroupBy(p => p.Cidade)

.Select(g => g.OrderBy(p => p.Idade).First())

.ToList();

// Exibindo os resultados

Console.WriteLine("Pessoa mais jovem de cada cidade:");

foreach (var pessoa in maisJovensPorCidade)

{

Console.WriteLine($"{pessoa.Nome}, Idade: {pessoa.Idade}, Cidade: {pessoa.Cidade}");

}

}

}

\*/

/\* Adicione uma lista de livros com propriedades Título, Autor e AnoPublicacao. Faça uma consulta LINQ para encontrar os livros publicados após o ano de 2000 e ordene-os por ano de publicação em ordem crescente.

public class Livro

{

public string Titulo { get; set; }

public string Autor { get; set; }

public int AnoPublicacao { get; set; }

}

public class Program

{

public static void Main()

{

var livros = new List<Livro>

{

new Livro { Titulo = "Livro A", Autor = "Autor A", AnoPublicacao = 2001 },

new Livro { Titulo = "Livro B", Autor = "Autor B", AnoPublicacao = 1999 },

new Livro { Titulo = "Livro C", Autor = "Autor C", AnoPublicacao = 2005 }

};

var livrosRecentes = livros

.Where(l => l.AnoPublicacao > 2000)

.OrderBy(l => l.AnoPublicacao)

.ToList();

// Exibindo os resultados

Console.WriteLine("Livros publicados após 2000:");

foreach (var livro in livrosRecentes)

{

Console.WriteLine($"{livro.Titulo}, Ano: {livro.AnoPublicacao}");

}

}

}

\*/

/\* Crie uma nova lista de pessoas com pelo menos 5 pessoas e adicione uma propriedade Cidade à classe Pessoa2. Realize uma consulta LINQ para listar todas as pessoas de uma cidade específica.

public class Pessoa

{

public string Nome { get; set; }

public int Idade { get; set; }

public string Cidade { get; set; }

}

public class Program

{

public static void Main()

{

var pessoas = new List<Pessoa>

{

new Pessoa { Nome = "Ana", Idade = 30, Cidade = "São Paulo" },

new Pessoa { Nome = "Bruno", Idade = 22, Cidade = "São Paulo" },

new Pessoa { Nome = "Carlos", Idade = 35, Cidade = "Rio de Janeiro" },

new Pessoa { Nome = "Diana", Idade = 25, Cidade = "Rio de Janeiro" }

};

var cidadeEspecifica = "São Paulo";

var pessoasNaCidade = pessoas

.Where(p => p.Cidade == cidadeEspecifica)

.ToList();

// Exibindo os resultados

Console.WriteLine($"Pessoas em {cidadeEspecifica}:");

foreach (var pessoa in pessoasNaCidade)

{

Console.WriteLine($"{pessoa.Nome}, Idade: {pessoa.Idade}");

}

}

}

\*/

/\* Utilizando a lista de pessoas criada anteriormente, faça uma consulta LINQ para encontrar a pessoa mais velha de cada cidade.

public class Pessoa

{

public string Nome { get; set; }

public int Idade { get; set; }

public string Cidade { get; set; }

}

public class Program

{

public static void Main()

{

var pessoas = new List<Pessoa>

{

new Pessoa { Nome = "Ana", Idade = 30, Cidade = "São Paulo" },

new Pessoa { Nome = "Bruno", Idade = 22, Cidade = "São Paulo" },

new Pessoa { Nome = "Carlos", Idade = 35, Cidade = "Rio de Janeiro" },

new Pessoa { Nome = "Diana", Idade = 25, Cidade = "Rio de Janeiro" }

};

var maisVelhosPorCidade = pessoas

.GroupBy(p => p.Cidade)

.Select(g => g.OrderByDescending(p => p.Idade).First())

.ToList();

// Exibindo os resultados

Console.WriteLine("Pessoa mais velha de cada cidade:");

foreach (var pessoa in maisVelhosPorCidade)

{

Console.WriteLine($"{pessoa.Nome}, Idade: {pessoa.Idade}, Cidade: {pessoa.Cidade}");

}

}

}

\*/

/\* Adicione uma nova lista de pessoas com informações de salários e crie uma consulta LINQ para calcular o salário médio das pessoas.

public class Pessoa

{

public string Nome { get; set; }

public decimal Salario { get; set; }

}

public class Program

{

public static void Main()

{

var pessoas = new List<Pessoa>

{

new Pessoa { Nome = "Ana", Salario = 3000 },

new Pessoa { Nome = "Bruno", Salario = 2500 },

new Pessoa { Nome = "Carlos", Salario = 4000 }

};

var salarioMedio = pessoas.Average(p => p.Salario);

// Exibindo o resultado

Console.WriteLine($"Salário médio: {salarioMedio}");

}

}

\*/

/\* Crie uma consulta LINQ para listar os nomes das pessoas que têm idade entre 20 e 30 anos e ordene os resultados por nome.

public class Pessoa

{

public string Nome { get; set; }

public int Idade { get; set; }

}

public class Program

{

public static void Main()

{

var pessoas = new List<Pessoa>

{

new Pessoa { Nome = "Ana", Idade = 30 },

new Pessoa { Nome = "Bruno", Idade = 22 },

new Pessoa { Nome = "Carlos", Idade = 35 },

new Pessoa { Nome = "Diana", Idade = 25 }

};

var nomesEntre20e30 = pessoas

.Where(p => p.Idade >= 20 && p.Idade <= 30)

.OrderBy(n => n.Nome)

.Select(n => n.Nome)

.ToList();

// Exibindo os resultados

Console.WriteLine("Nomes das pessoas com idade entre 20 e 30 anos:");

foreach (var nome in nomesEntre20e30)

{

Console.WriteLine(nome);

}

}

}

\*/

/\* Crie uma consulta LINQ para contar quantas pessoas têm mais de 30 anos e imprimir o resultado.

public class Pessoa

{

public string Nome { get; set; }

public int Idade { get; set; }

}

public class Program

{

public static void Main()

{

var pessoas = new List<Pessoa>

{

new Pessoa { Nome = "Ana", Idade = 30 },

new Pessoa { Nome = "Bruno", Idade = 22 },

new Pessoa { Nome = "Carlos", Idade = 35 },

new Pessoa { Nome = "Diana", Idade = 40 }

};

var contagemMaisDe30 = pessoas.Count(p => p.Idade > 30);

// Exibindo o resultado

Console.WriteLine($"Número de pessoas com mais de 30 anos: {contagemMaisDe30}");

}

}

\*/

}