

Assignment 7

Linq Demo:

Program.cs

C# LinqDemo_Console_App

LINQHandsOn

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4
5 namespace LINQHandsOn
6 {
7     // 8 references | Project S, Less than 5 minutes ago | 1 author, 1 change
8     public class Employee
9     {
10         // 5 references | Project S, Less than 5 minutes ago | 1 author, 1 change
11         public int Id { get; set; }
12         // 16 references | Project S, Less than 5 minutes ago | 1 author, 1 change
13         public string Name { get; set; }
14         // 7 references | Project S, Less than 5 minutes ago | 1 author, 1 change
15         public string Department { get; set; }
16         // 11 references | Project S, Less than 5 minutes ago | 1 author, 1 change
17         public decimal Salary { get; set; }
18         // 8 references | Project S, Less than 5 minutes ago | 1 author, 1 change
19         public int DepartmentId { get; set; }
20     }
21
22     // 5 references | Project S, Less than 5 minutes ago | 1 author, 1 change
23     public class Department
24     {
25         // 5 references | Project S, Less than 5 minutes ago | 1 author, 1 change
26         public int DepartmentId { get; set; }
27         // 5 references
28         public string DepartmentName { get; set; }
29     }
30
31     // 0 references
32     internal class Program
33     {
34         // 0 references
35         static void Main(string[] args)
36         {
37             Employee employee = new Employee();
38             List<Employee> employees = new List<Employee>
39             {
40                 new Employee { Id = 1, Name = "John", Department = "IT", Salary = 60000, DepartmentId = 1 },
41                 new Employee { Id = 2, Name = "Sarah", Department = "HR", Salary = 50000, DepartmentId = 2 },
42                 new Employee { Id = 3, Name = "Steve", Department = "IT", Salary = 70000, DepartmentId = 1 },
43                 new Employee { Id = 4, Name = "Anna", Department = "Finance", Salary = 80000, DepartmentId = 3 },
44                 new Employee { Id = 5, Name = "Mike", Department = "Finance", Salary = 65000, DepartmentId = 3 },
45                 new Employee { Id = 6, Name = "John", Department = "IT", Salary = 60000, DepartmentId = 1 }, // duplicate name
46             };
47
48             List<Department> departments = new List<Department>
49             {
50                 new Department { DepartmentId = 1, DepartmentName = "IT" },
51                 new Department { DepartmentId = 2, DepartmentName = "HR" },
52                 new Department { DepartmentId = 3, DepartmentName = "Finance" }
53             };
54
55             var highSalary = employees.Where(e => e.Salary > 60000);
56             Console.WriteLine("1. Employees with Salary > 60000:");
57             foreach (var e in highSalary)
58             {
59                 Console.WriteLine($"{e.Name} - {e.Salary}");
60             }
61             Console.WriteLine();
62         }
63     }
64 }
```

Program.cs

C# LinqDemo_Console_App

LINQHandsOn.Program

```
49 Console.WriteLine();
50
51
52 var nameAndSalary = employees.Select(e => new { e.Name, e.Salary });
53 Console.WriteLine("2. Employee Names and Salaries:");
54 foreach (var e in nameAndSalary)
55     Console.WriteLine($"{e.Name} - {e.Salary}");
56 Console.WriteLine();
57
58 var sortByName = employees.OrderBy(e => e.Name);
59 Console.WriteLine("3A. Employees Sorted by Name (Ascending):");
60 foreach (var e in sortByName)
61     Console.WriteLine($"{e.Name}");
62 Console.WriteLine();
63
64 var sortBySalaryDesc = employees.OrderByDescending(e => e.Salary);
65 Console.WriteLine("3B. Employees Sorted by Salary (Descending):");
66 foreach (var e in sortBySalaryDesc)
67     Console.WriteLine($"{e.Name} - {e.Salary}");
68 Console.WriteLine();
69
70
71 var groupedByDept = employees.GroupBy(e => e.Department);
72 Console.WriteLine("4. Grouped by Department:");
73 foreach (var group in groupedByDept)
74 {
75     Console.WriteLine($"{group.Key} Department:");
76     foreach (var e in group)
77         Console.WriteLine($"- {e.Name}");
78     Console.WriteLine();
79 }
80
```

```

79     }
80
81
82     var innerJoin = employees.Join(departments,
83         emp => emp.DepartmentId,
84         dept => dept.DepartmentId,
85         (emp, dept) => new { emp.Name, dept.DepartmentName });
86
87     Console.WriteLine("5A. Joined Data (Employee + Department):");
88     foreach (var e in innerJoin)
89         Console.WriteLine($"{e.Name} - {e.DepartmentName}");
90     Console.WriteLine();
91
92
93     var groupJoin = departments.GroupJoin(employees,
94         dept => dept.DepartmentId,
95         emp => emp.DepartmentId,
96         (dept, emps) => new { dept.DepartmentName, Employees = emps });
97
98     Console.WriteLine("5B. Departments and their Employees:");
99     foreach (var d in groupJoin)
100     {
101         Console.WriteLine($"{d.DepartmentName} Department:");
102         foreach (var e in d.Employees)
103             Console.WriteLine($"- {e.Name}");
104         Console.WriteLine();
105     }
106
107
108     var distinctNames = employees.Select(e => e.Name).Distinct();
109     Console.WriteLine("6. Distinct Employee Names:");
110     Console.WriteLine(string.Join(", ", distinctNames));
111     Console.WriteLine();
112
113     int pageSize = 2;
114     int pageNumber = 2;
115     var paginated = employees.Skip((pageNumber - 1) * pageSize).Take(pageSize);
116
117     Console.WriteLine($"7. Pagination (Page {pageNumber}):");
118     int count = (pageNumber - 1) * pageSize + 1;
119     foreach (var e in paginated)
120     {
121         Console.WriteLine($"Employee {count++}: {e.Name}");
122     }
123
124     Console.ReadLine();
125 }
126 }
127 }
128

```

Github Link:

https://github.com/praveen-dotnet-chn/LinqDemo_Console_App