

**Московский государственный технический университет им. Н.Э.
Баумана**

Кафедра
“Системы обработки информации и управления”
(ИУ – 5)

**Отчет по лабораторной работе № 7
по дисциплине “Базовые компоненты Интернет-технологий”**

Выполнила:
студентка гр. ИУ5 - 31Б
Аушева Лиза
28 сентября 2018 г.

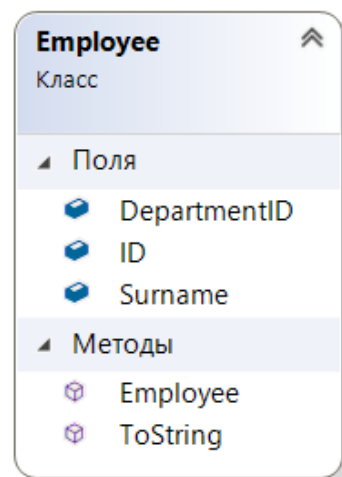
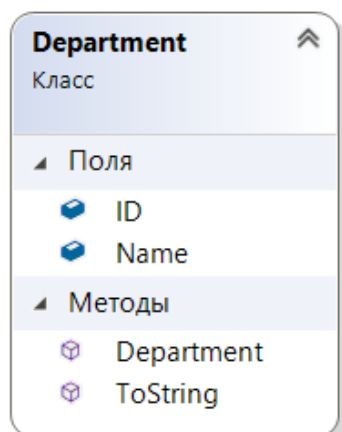
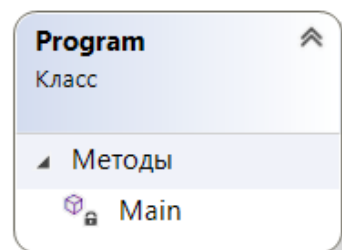
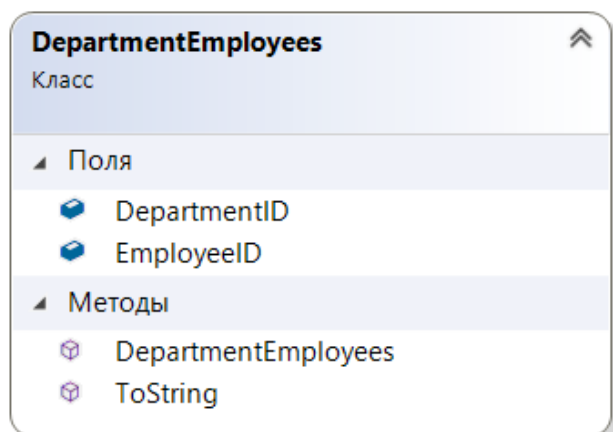
Москва – 2018 г.

Лабораторная работа №7

Разработать программу, реализующую работу с LINQ to Objects. В качестве примера используйте проект «SimpleLINQ» из примера «Введение в LINQ».

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Создайте класс «Сотрудник», содержащий поля:
 - ID записи о сотруднике;
 - Фамилия сотрудника;
 - ID записи об отделе.
3. Создайте класс «Отдел», содержащий поля:
 - ID записи об отделе;
 - Наименование отдела.
4. Предполагая, что «Отдел» и «Сотрудник» связаны соотношением один-ко-многим разработайте следующие запросы:
 - Выведите список всех сотрудников и отделов, отсортированный по отделам.
 - Выведите список всех сотрудников, у которых фамилия начинается с буквы «А».
 - Выведите список всех отделов и количество сотрудников в каждом отделе.
 - Выведите список отделов, в которых у всех сотрудников фамилия начинается с буквы «А».
 - Выведите список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы «А».
5. Создайте класс «Сотрудники отдела», содержащий поля:
 - ID записи о сотруднике;
 - ID записи об отделе.
6. Предполагая, что «Отдел» и «Сотрудник» связаны соотношением много-ко-многим с использованием класса «Сотрудники отдела» разработайте следующие запросы:
 - Выведите список всех отделов и список сотрудников в каждом отделе.
 - Выведите список всех отделов и количество сотрудников в каждом отделе.

Диаграмма классов



Текст программы

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.ComponentModel.Design;

namespace Lab7
{
    class Program
    {
        static void Main(string[] args)
        {
            var departments = new List<Department>
            {
                new Department(1, "Бухгалтерия"),
                new Department(2, "Маркетинговый отдел"),
                new Department(3, "Финансовый отдел"),
                new Department(4, "Отдел управления поставками"),
                new Department(5, "Отдел технической поддержки"),
                new Department(6, "Отдел разработки программного обеспечения"),
                new Department(7, "Отдел тестирования")
            };

            var employees = new List<Employee>
            {
                new Employee(1, "Кочетков", 6),
                new Employee(2, "Белкина", 6),
                new Employee(3, "Дехтеров", 7),
                new Employee(4, "Крюков", 5),
                new Employee(5, "Хрюков", 1),
                new Employee(6, "Канюков", 3),
                new Employee(7, "Бадышев", 1),
                new Employee(8, "Антонова", 1),
                new Employee(9, "Симонов", 4),
                new Employee(10, "Сифонов", 3),
                new Employee(11, "Котов", 7),
                new Employee(12, "Датов", 5),
                new Employee(13, "Аверин", 2),
                new Employee(14, "Агатов", 2)
            };

            Console.WriteLine("Выведите список всех сотрудников и отделов,
отсортированный по отделам");

            var list1 =
                from employee in employees
                join department in departments on employee.DepartmentID equals
department.ID
                orderby department.Name
                select new
                {
                    employee.Surname,
                    DepartmentName = department.Name
                };

            foreach (var item in list1)
            {
                Console.WriteLine(item);
            }
        }
    }
}
```

```

Console.WriteLine();
Console.WriteLine("Выведите список всех сотрудников, у которых фамилия
начинается с буквы «А»");

var list2 =
    from employee in employees
    where employee.Surname[0] == 'A'
    select employee;

foreach (var item in list2)
{
    Console.WriteLine(item);
}

Console.WriteLine();
Console.WriteLine("Выведите список всех отделов и количество сотрудников в
каждом отделе");

var list3 =
    from department in departments
    join employee in employees on department.ID equals employee.DepartmentID
into employeesOnDepartment
    select new
    {
        DepartmentID = department.ID,
        DepartmentName = department.Name,
        CountOfEmployees = employeesOnDepartment.Count()
    };

foreach (var item in list3)
{
    Console.WriteLine(item);
}

Console.WriteLine();
Console.WriteLine("Выведите список отделов, в которых у всех сотрудников
фамилия начинается с буквы «А»");

var list4 =
    from department in departments
    join employee in employees on department.ID equals employee.DepartmentID
into employeesOnDepartment
    where employeesOnDepartment.All(employee => employee.Surname[0] == 'A')
    select department;

foreach (var item in list4)
{
    Console.WriteLine(item);
}

Console.WriteLine();
Console.WriteLine("Выведите список отделов, " +
    "в которых хотя бы у одного сотрудника фамилия начинается с
буквы «А»");

var list5 =
    from department in departments
    join employee in employees on department.ID equals employee.DepartmentID
into employeesOnDepartment
    where employeesOnDepartment.Any(employee => employee.Surname[0] == 'A')
    select department;

```

```

foreach (var item in list5)
{
    Console.WriteLine(item);
}

Console.WriteLine();

var departmentEmployees = new List<DepartmentEmployees>
{
    new DepartmentEmployees(5, 3),
    new DepartmentEmployees(5, 1),
    new DepartmentEmployees(5, 3),
    new DepartmentEmployees(7, 5),
    new DepartmentEmployees(3, 2),
    new DepartmentEmployees(6, 7),
    new DepartmentEmployees(6, 7),
    new DepartmentEmployees(6, 7),
    new DepartmentEmployees(7, 5),
    new DepartmentEmployees(7, 2),
    new DepartmentEmployees(5, 1),
    new DepartmentEmployees(4, 6),
    new DepartmentEmployees(4, 5),
    new DepartmentEmployees(1, 4),
    new DepartmentEmployees(2, 3),
    new DepartmentEmployees(3, 3),
    new DepartmentEmployees(11, 2),
    new DepartmentEmployees(14, 5),
    new DepartmentEmployees(12, 4),
    new DepartmentEmployees(4, 7),
    new DepartmentEmployees(3, 5),
    new DepartmentEmployees(9, 2),
    new DepartmentEmployees(9, 4),
    new DepartmentEmployees(10, 6),
    new DepartmentEmployees(12, 4),
    new DepartmentEmployees(7, 2),
    new DepartmentEmployees(1, 1),
    new DepartmentEmployees(2, 1),
    new DepartmentEmployees(12, 5)
};

Console.WriteLine("Выведите список всех отделов и список сотрудников в каждом
отделе");

var list6 =
    from departmentEmployee in departmentEmployees
    group departmentEmployees by departmentEmployee.DepartmentID into deps
    select new
    {
        Department = departments.Single(dep => dep.ID == deps.Key),
        Employees = string.Join(", ", employees.FindAll(employee =>
employee.DepartmentID == deps.Key))
    };

foreach (var item in list6)
{
    Console.WriteLine(item);
}

Console.WriteLine();
Console.WriteLine("Выведите список всех отделов и количество сотрудников в
каждом отделе");

var list7 =
    from departmentEmployee in departmentEmployees
    group departmentEmployees by departmentEmployee.DepartmentID into deps

```

```

        select new
        {
            Department = departments.Single(dep => dep.ID == deps.Key),
            EmployeesCount = employees.FindAll(employee => employee.DepartmentID
== deps.Key).Count
        };

        foreach (var item in list7)
        {
            Console.WriteLine(item);
        }
    }
}

```

Employee.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab7
{
    public class Employee
    {
        public int ID;
        public string Surname;
        public int DepartmentID;

        public Employee(int id, string surname, int departmentId)
        {
            ID = id;
            Surname = surname;
            DepartmentID = departmentId;
        }

        public override string ToString()
        {
            return string.Format("{ Employee ID: {0}, Surname: {1}, DepartmentID: {2}
}}", ID, Surname, DepartmentID);
        }
    }
}

```

Department.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab7
{
    public class Department
    {
        public int ID;
        public string Name;

        public Department(int id, string name)
        {
            ID = id;

```

```

        Name = name;
    }

    public override string ToString()
    {
        return string.Format("{0} Department ID: {1}, Name: {2} }", ID, Name);
    }
}
}

```

DepartmentEmployees.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab7
{
    public class DepartmentEmployees
    {
        public int EmployeeID;
        public int DepartmentID;

        public DepartmentEmployees(int employeeId, int departmentId)
        {
            EmployeeID = employeeId;
            DepartmentID = departmentId;
        }

        public override string ToString()
        {
            return string.Format("{0} DepartmentEmployees EmployeeID: {1}, DepartmentID: {2} }", EmployeeID, DepartmentID);
        }
    }
}

```


Экранные формы с примерами выполнения программы:

```
cmd.exe
Выведите список всех сотрудников и отделов, отсортированный по отделам
{ Surname = Хрюков, DepartmentName = Бухгалтерия }
{ Surname = Бадмаев, DepartmentName = Бухгалтерия }
{ Surname = Антонова, DepartmentName = Бухгалтерия }
{ Surname = Аверин, DepartmentName = Маркетинговый отдел }
{ Surname = Аратов, DepartmentName = Маркетинговый отдел }
{ Surname = Кочетков, DepartmentName = Отдел разработки программного обеспечения }
{ Surname = Белкина, DepartmentName = Отдел разработки программного обеспечения }
{ Surname = Дехтеров, DepartmentName = Отдел тестирования }
{ Surname = Котов, DepartmentName = Отдел тестирования }
{ Surname = Крюков, DepartmentName = Отдел технической поддержки }
{ Surname = Датов, DepartmentName = Отдел технической поддержки }
{ Surname = Симонов, DepartmentName = Отдел управления поставками }
{ Surname = Канюков, DepartmentName = Финансовый отдел }
{ Surname = Сифонов, DepartmentName = Финансовый отдел }

Выведите список всех сотрудников, у которых фамилия начинается с буквы <A>
{ Employee ID: 8, Surname: Антонова, DepartmentID: 1 }
{ Employee ID: 13, Surname: Аверин, DepartmentID: 2 }
{ Employee ID: 14, Surname: Аратов, DepartmentID: 2 }

Выведите список всех отделов и количество сотрудников в каждом отделе
{ DepartmentID = 1, DepartmentName = Бухгалтерия, CountOfEmployees = 3 }
{ DepartmentID = 2, DepartmentName = Маркетинговый отдел, CountOfEmployees = 2 }
{ DepartmentID = 3, DepartmentName = Финансовый отдел, CountOfEmployees = 2 }
{ DepartmentID = 4, DepartmentName = Отдел управления поставками, CountOfEmployees = 1 }
{ DepartmentID = 5, DepartmentName = Отдел технической поддержки, CountOfEmployees = 2 }
{ DepartmentID = 6, DepartmentName = Отдел разработки программного обеспечения, CountOfEmployees = 2 }
{ DepartmentID = 7, DepartmentName = Отдел тестирования, CountOfEmployees = 2 }

Выведите список отделов, в которых у всех сотрудников фамилия начинается с буквы <A>
{ Department ID: 2, Name: Маркетинговый отдел }

Выведите список отделов, в которых хотя бы у одного сотрудника фамилия начинается с буквы <A>
{ Department ID: 1, Name: Бухгалтерия }
{ Department ID: 2, Name: Маркетинговый отдел }

Выведите список всех отделов и список сотрудников в каждом отделе
{ Department = { Department ID: 3, Name: Финансовый отдел }, Employees = { Employee ID: 6, Surname: Канюков, DepartmentID: 3 }, { Employee ID: 10, Surname: Сифонов, DepartmentID: 3 } }
{ Department = { Department ID: 1, Name: Бухгалтерия }, Employees = { Employee ID: 5, Surname: Хрюков, DepartmentID: 1 }, { Employee ID: 7, Surname: Бадмаев, DepartmentID: 1 }, { Employee ID: 8, Surname: Антонова, DepartmentID: 1 } }
{ Department = { Department ID: 5, Name: Отдел технической поддержки }, Employees = { Employee ID: 4, Surname: Крюков, DepartmentID: 5 }, { Employee ID: 12, Surname: Датов, DepartmentID: 5 } }
{ Department = { Department ID: 2, Name: Маркетинговый отдел }, Employees = { Employee ID: 13, Surname: Аверин, DepartmentID: 2 }, { Employee ID: 14, Surname: Аратов, DepartmentID: 2 } }
{ Department = { Department ID: 7, Name: Отдел тестирования }, Employees = { Employee ID: 3, Surname: Дехтеров, DepartmentID: 7 }, { Employee ID: 11, Surname: Котов, DepartmentID: 7 } }
{ Department = { Department ID: 6, Name: Отдел разработки программного обеспечения }, Employees = { Employee ID: 1, Surname: Кочетков, DepartmentID: 6 }, { Employee ID: 2, Surname: Белкина, DepartmentID: 6 } }
{ Department = { Department ID: 4, Name: Отдел управления поставками }, Employees = { Employee ID: 9, Surname: Симонов, DepartmentID: 4 } }

Выведите список всех отделов и список сотрудников в каждом отделе
{ Department = { Department ID: 3, Name: Финансовый отдел }, Employees = { Employee ID: 6, Surname: Канюков, DepartmentID: 3 }, { Employee ID: 10, Surname: Сифонов, DepartmentID: 3 } }
{ Department = { Department ID: 1, Name: Бухгалтерия }, Employees = { Employee ID: 5, Surname: Хрюков, DepartmentID: 1 }, { Employee ID: 7, Surname: Бадмаев, DepartmentID: 1 }, { Employee ID: 8, Surname: Антонова, DepartmentID: 1 } }
{ Department = { Department ID: 5, Name: Отдел технической поддержки }, Employees = { Employee ID: 4, Surname: Крюков, DepartmentID: 5 }, { Employee ID: 12, Surname: Датов, DepartmentID: 5 } }
{ Department = { Department ID: 2, Name: Маркетинговый отдел }, Employees = { Employee ID: 13, Surname: Аверин, DepartmentID: 2 }, { Employee ID: 14, Surname: Аратов, DepartmentID: 2 } }
{ Department = { Department ID: 7, Name: Отдел тестирования }, Employees = { Employee ID: 3, Surname: Дехтеров, DepartmentID: 7 }, { Employee ID: 11, Surname: Котов, DepartmentID: 7 } }
{ Department = { Department ID: 6, Name: Отдел разработки программного обеспечения }, Employees = { Employee ID: 1, Surname: Кочетков, DepartmentID: 6 }, { Employee ID: 2, Surname: Белкина, DepartmentID: 6 } }
{ Department = { Department ID: 4, Name: Отдел управления поставками }, Employees = { Employee ID: 9, Surname: Симонов, DepartmentID: 4 } }

Выведите список всех отделов и количество сотрудников в каждом отделе
{ Department = { Department ID: 3, Name: Финансовый отдел }, EmployeesCount = 2 }
{ Department = { Department ID: 1, Name: Бухгалтерия }, EmployeesCount = 3 }
{ Department = { Department ID: 5, Name: Отдел технической поддержки }, EmployeesCount = 2 }
{ Department = { Department ID: 2, Name: Маркетинговый отдел }, EmployeesCount = 2 }
{ Department = { Department ID: 7, Name: Отдел тестирования }, EmployeesCount = 2 }
{ Department = { Department ID: 6, Name: Отдел разработки программного обеспечения }, EmployeesCount = 2 }
{ Department = { Department ID: 4, Name: Отдел управления поставками }, EmployeesCount = 1 }

Для продолжения нажмите любую клавишу . . .
```